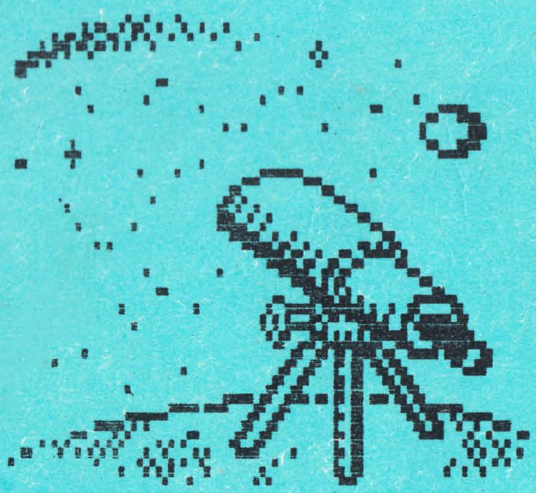
Tim-S Plus

șapte calculatoare într-unul singur

Pănescu Dumitru
Bărbulescu Corneliu
Moldovan Horațiu



Vol. II

Timișoara, 1990

Tim-S Plus

sapte calculatoare intr-unul singur

Panescu Dumitru

Societatea μHard

Fabrica de memorii electronice si
componente pentru tehnica de calcul

Editura "TM", Timisoara 1990

Tim-S Plus

sapte calculatoare intr-unul singur

Compatibilitate soft:

Calculatorul personal ZX-Spectrum 48K
Calculatorul personal ZX-Spectrum +
Calculatorul personal ZX-Spectrum 128K
Calculatorul personal ZX-Spectrum +2
Calculatorul personal ZX-Spectrum +3
Interface I
Sistemul de operare CP/M V2.2

Panescu Dumitru

Facultatea Electrotehnica
Institutul Politehnic Traian Vuia

Societatea µHard

Fabrica de memorii electronice si
Componente pentru tehnica de calcul

Muzica: Panescu Dumitru

Timisoara, 04.09.89

Catedra de calculatoare a Institutului Politehnic;
B-dul Vasile Pirvan, nr. 2, tel. 961.12330.371, Timisoara

Fabrica de memorii electronice si componente pentru tehnica de calcul; B-dul Gh.Lazar, nr. 9, tel. 961.30078, Timisoara

6 Sistemul de operare CP/M

Rutina 23 - Schimbare nume fisier
Rutina 24 - Citire vector de unitati-disc active
Rutina 25 - Citire numar disc selectat
Rutina 26 - Modificare "adresa DMA"
Rutina 27 - Citire adresa vector de alocare
Rutina 28 - Setare atribut R/O pentru o unitate de disc
Rutina 29 - Citire vector de unitati R/O
Rutina 30 - Modificare atribute fisier
Rutina 31 - Citire adresa "bloc de parametrii disc"
Rutina 32 - Citire/Modificare numar utilizator
Rutina 33 - Citire directa
Rutina 34 - Scriere directa
Rutina 35 - Determinare lungime fisier
Rutina 36 - Determinare numar inregistrare
Rutina 37 - Dezactivare discuri
Rutina 40 - Scriere directa cu umplere cu zero

6.7.6 Breviarul principalelor rutine CP/M

6.8 Programe utilitare de baza sub CP/M

6.8.1 Editor - ED

6.8.1.1 Prezentare generala

6.8.1.2 Comenzi ED

6.8.1.2.1 Comenzi de I/E

6.8.1.2.2 Comenzi de editare texte

6.8.1.2.2.1 Comenzi la nivel de caracter

6.8.1.2.2.2 Comenzi la nivel de linie

6.8.1.2.2.3 Comenzi asupra sirurilor de caractere

6.8.1.2.3 Comenzi de lucru cu biblioteci sursa

6.8.1.2.4 Comenzi generale

6.8.1.3 Mesaje de eroare ED

6.8.1.4 Caractere de control disponibile in ED

6.8.1.5 Comenzi disponibile in ED

6.8.2 Asamblor - ASM

6.8.2.1 Prezentare generala

6.8.2.2 Formatul fisierului sursa

6.8.2.2.1 Cimpul "numar linie"

6.8.2.2.2 Cimpul "eticheta"

6.8.2.2.3 Cimpul "cod operatie"

6.8.2.2.4 Cimpul "operand"

6.8.2.2.4.1 Operanzi

6.8.2.2.4.2 Operatori

6.8.2.2.5 Cimpul "comentariu"

6.8.2.3 Mesaje de eroare ASM

6.8.2.4 Directive acceptate de ASM

6.8.3 Depanator - ZSID

6.8.3.1 Prezentare generala

6.8.3.2 Comenzi ZSID

6.8.3.2.1 Comanda D (display)

6.8.3.2.2 Comanda F (fill)

6.8.3.2.3 Comanda M (move)

6.8.3.2.4 Comanda S (substitute)

6.8.3.2.5 Comanda X (examine)

6.8.3.2.6 Comanda H (hexa)

6.8.3.2.7 Comanda G (go)

6.8.3.2.8 Comanda T (trace)

6.8.3.2.9 Comanda U (untrace)

6.8.3.2.10 Comanda L (list)

6.9.5.9 DISCONNECT
6.9.5.10 DISPLAY
6.9.5.11 ERASE
6.9.5.12 EXIT
6.9.5.13 FINISH
6.9.5.14 GET
6.9.5.15 HANGUP
6.9.5.16 HOST
6.9.5.17 LOCAL
6.9.5.18 QUIT
6.9.5.19 PRINT
6.9.5.20 RECEIVE
6.9.5.21 REMOTE
- 6.9.5.21.1 BYE
- 6.9.5.21.2 COPY
- 6.9.5.21.3 DIRECT
- 6.9.21.5.4 ERASE
- 6.9.5.21.5 FINISH
- 6.9.5.21.6 GET
- 6.9.5.21.7 HELP
- 6.9.5.21.8 RENAME
- 6.9.5.21.9 TYPE

6.9.5.22 RENAME
6.9.5.23 RSX11M
6.9.5.24 SEND
6.9.5.25 SERVER
6.9.5.26 SET
- 6.9.5.26.1 ATTRIBUTES
- 6.9.5.26.2 BAUD
- 6.9.5.26.3 DEBUG
  - 6.9.5.26.3.1 ALL
  - 6.9.5.26.3.2 CONSOLE
  - 6.9.5.26.3.3 CONNECT
  - 6.9.5.26.3.4 FILE
  - 6.9.5.26.3.5 HELP
  - 6.9.5.26.3.6 NONE
  - 6.9.5.26.3.7 OFF
  - 6.9.5.26.3.8 ON
  - 6.9.5.26.3.9 PACKET
- 6.9.5.26.4 DELAY
- 6.9.5.26.5 DEFAULT
- 6.9.5.26.6 DUPLEX
- 6.9.5.26.7 END-OF-LINE
- 6.9.5.26.8 ESCAPE
- 6.9.5.26.9 FILETYPE
  - 6.9.5.26.9.1 ASCII
  - 6.9.5.26.9.2 BINARY
- 6.9.5.26.10 HOME
- 6.9.5.26.11 LINE
- 6.9.5.26.12 LOGFILE
- 6.9.5.26.13 PACKET-LENGHT
- 6.9.5.26.14 PARITY
- 6.9.5.26.15 PAUSE
- 6.9.5.26.16 PROMPT
- 6.9.5.26.17 RANDOM
- 6.9.5.26.18 RECORD-FORMAT
- 6.9.5.26.19 RETRY
- 6.9.5.26.20 SPEED
- 6.9.5.26.21 TIMEOUT

6.9.5.27 SHOW
- 6.9.5.27.1 ALL

6 Sistemul de operare CP/M

6.1 Generalitati

CP/M este un sistem de operare pentru microcalculatoarele care utilizeaza o unitate centrala compatibila la nivel de cod obiect cu microprocesorul Intel 8080 si care cuprind in configuratie o unitate de memorie externa cu acces aleator de tip disc magnetic. El ofera cadrul general necesar pentru constructia, stocarea si editarea programelor, dispunind totodata de facilitati pentru asamblarea si depanarea acestora. O caracteristica a sistemului CP/M este faptul ca el poate fi usor adaptat la orice configuratie de microcalculator cu memorie interna de cel putin 16 KB si pina la 16 unitati de discuri.

CP/M asigura accesul rapid la programe prin intermediul unui sistem de gestiune a fisierelor. Acest sistem gestioneaza o structura de fisiere, identificate prin nume, permite alocarea dinamica a spatiului de pe disc pentru fisiere si asigura accesul secvential si direct la fisiere.

Prin utilizarea acestui sistem de gestiune a fisierelor se pot stoca, in format sursa sau direct executabil (cod-masina) un mare numar de programe distincte.

Sistemul CP/M standard contine, de asemenea, un editor de texte (contextual), un asamblor (compatibil Intel), instrumente (subsisteme) pentru depanarea programelor. Optional, sistemul CP/M mai include un macroasamblor (compatibil Intel), un depanator simbolic de programe, precum si o gama larga de limbaje de nivel inalt. Exploatarea tuturor componentelor CP/M prin intermediul "procesorului de comenzi-consola" - CCP (Console Command Processor), care asigura un regim de lucru conversational, face ca sistemul CP/M sa posede in ansamblu facilitati egale sau chiar mai mari decit cele existente la sistemele de calcul mari.

Din punct de vedere logic, sistemul CP/M este alcatuit din urmatoarele componente:

- **BIOS** (Basic I/O System)
  sistemul de I/E de baza
  dependent de hardware

- **BDOS** (Basic Disk Operating System)
  sistemul de exploatare
  a discurilor

- **CCP** (Console Command Processor)
  procesorul de comenzi-consola

- **TPA** (Transient Program Area)
  zona pentru programe tranzitorii

Componenta BIOS asigura operatiile de baza elementare necesare pentru accesul la unitatile de discuri, precum si pentru interfata cu perifericele standard (consola (TTY) display (CRT) cititor/perforator de banda de hirtie si periferice definite de utilizator). Componenta BIOS poate fi modificata pentru orice configuratie hardware particulara.

Componenta BDOS asigura gestiunea discurilor, controlind una sau mai multe unitati de discuri, ce contin fisiere "director" independente. Componenta BDOS implementeaza strategia de alocare

a discului, care asigura constructia total dinamica a fisierelor, minimizind in acelasi timp miscarea capului de citire pe disc, in timpul accesului la acesta.

Orice fisier CP/M poate contine oricite inregistrari, in limita spatiului unui volum disc. Un volum disc poate contine un numar maxim de fisiere dependent de modul in care sistemul a fost generat. In versiunea actuala, un volum disc poate contine maximum 128 fisiere distincte.

Componenta BDOS are puncte de intrare, ce pot fi folosite de catre programele-utilizator in rutine ce implementeaza operatii primitive, cum ar fi:

SEARCH - cautarea unui fisier pe disc dupa numele acestuia;
OPEN - deschiderea unui fisier pentru operatii ulterioare;
CLOSE - inchiderea unui fisier;
RENAME - schimbarea numelui unui fisier;
READ - citirea unei inregistrari dintr-un fisier dat;
WRITE - scrierea unei inregistrari pe disc;
SELECT - selectarea unei anumite unitati de disc pentru operatii ulterioare.

Componenta CCP asigura interfata simbolica intre utilizator si sistemul CP/M. CCP preia informatiile furnizate de utilizator prin consola si executa urmatoarele comenzi:

-listarea continutului fisierului "director" asociat utilizatorului curent;
-listarea continutului unor fisiere;
-stergerea unor fisiere;
-schimbarea numelui unor fisiere;
-salvarea continutului unei zone de memorie;
-controlul executiei unor programe tranzitorii (ca de exemplu: asambloare, editoare de texte, compilatoare, programe-utilizator).

Lista comenzilor CCP disponibile este prezentata in capitolele urmatoare.

Ultima componenta a CP/M este o zona numita "zona pentru programe tranzitorii", TPA (Transient Program Area). In aceasta zona se depun toate programele care se incarca de pe disc sub controlul CCP. In timpul editarii unui program de exemplu, zona TPA contine codul-obiect al editorului de texte CP/M, precum si zonele de date cu care acesta lucreaza. In mod similar, programele create sub CP/M pot fi testate prin incarcarea lor in zona TPA, unde urmeaza a se face testarea, prin lansarea lor in executie (in general sub controlul unui program specializat in depanare, cum ar fi ZSID).

Oricare din componentele CP/M (sau toate) pot fi "reacoperite" de catre un program in executie. Astfel, dupa ce un program utilizator a fost incarcat in zona TPA, zonele care contin componentele CCP, BDOS si BIOS pot fi folosite de program ca zona proprie de date. Ori de cite ori zona corespunzatoare componentei BIOS nu este "reacoperita", utilizatorul poate, prin program, sa apeleze "incarcatorul", programul de initializare a lucrului cu CP/M. Astfel, un program-utilizator, la sfirsitul executiei sale, nu trebuie sa execute, in acest caz, decit un salt la "incarcator", prin aceasta asigurindu-se automat posibilitatea reincarcarii complete, de pe disc, a sistemului CP/M.

Subliniem inca o data faptul ca sistemul de operare CP/M este alcatuit din module distincte, inclusiv din partea de BIOS care defineste mediul hardware (configuratia curenta) in care

se executa CP/M. Astfel, sistemul CP/M standard poate fi usor modificat pentru orice mediu hardware nestandard prin schimbarea driverelor (rutinelor) pentru periferice in functie de configuratia particulara.

6.2 Functionare CP/M

6.2.1 Proceduri de exploatare pentru utilizarea CP/M

Utilizatorul lucreaza cu sistemul CP/M in primul rind prin intermediul componentei CCP, care preia comenzile introduse de la consola si le interpreteaza. In general, componenta CCP adreseaza la un moment dat o singura unitate de disc din cele care sint operationale (on-line) curent (sistemul CP/M standard recunoaste maximum 16 unitati de discuri). Unitatile de discuri se identifica in CP/M prin literele A, B, C,...P. Un disc este "instalat" daca CCP il adreseaza in momentul respectiv. Pentru a indica clar in fiecare moment care este discul instalat, CCP comunica utilizatorului (prin afisare la consola) numele discului, urmat de simbolul ">", indicind in acest mod faptul ca CCP asteapta o noua comanda.

Initializarea sistemului CP/M se realizeaza astfel:

-se introduce in unitatea 0 ("A") un disc CP/M;
-se actioneaza comutatorul LOAD de la panoul frontal.

In urma acestor comenzi va apare la consola mesajul:

xxk CP/M vers m.m

xx -este dimensiunea memoriei interne (in Kiloocteti) pe care o gestioneaza sistemul CP/M curent;
m.m -este numarul de versiune.

Sistemele CP/M pot fi usor reconfigurate pentru orice dimensiune de memorie de care dispune microcalculatorul gazda (vezi comanda tranzitorie MOVCPM).

Initializarea CP/M-ului are ca efect incarcarea de pe disc in memorie a componentelor CCP, BDOS si BIOS.

Dupa mesajul de inceput, sistemul CP/M instaleaza automat discul "A", afiseaza mesajul "A>" si asteapta o noua comanda.

Orice schimbare de disc intr-o unitate trebuie urmata de o operatie denumita "reincarcarea" CP/M-ului, care se realizeaza prin comanda CTRL/C. "Reincarcarea" CP/M-ului se poate face numai dupa ce sistemul a fost cel putin o data initializat. Aceasta operatie consta in "reincarcarea" de pe disc in memorie a componentelor CCP si BDOS. Operatia de "reincarcare" se realizeaza si prin:

-executia instructiunilor RST 0 sau JMP 0;
-o intrerupere pe nivelul 0.

In timpul lucrului cu sistemul CP/M, utilizatorul poate sa reasigneze discul instalat (care era initial discul "A") printr-o comanda de tipul:

nume-dispozitiv:(CR)

unde **nume-dispozitiv** este una din literele A, B, ... P si reprezinta numele noului disc care va fi instalat. Astfel, secventa

de comenzi prezentata in continuare poate fi utilizata dupa ce sistemul CP/M a fost initializat de pe discul "A":

```
     53k CP/M vers 2.2 (mesaj afisat automat la initializare)
A>DIR   (listeaza numele tuturor fisierelor de pe discul "A")

A: PROG     ASM : PROG     PRN : PROG     LIB : DUMP     ASM
A: PRO      ASM : TEST     ASM

A>B:      (reasignarea discului "B" ca disc instalat)
B>DIR *.ASM  (listarea numelor tuturor fisierelor cu extensie
             "ASM" de pe discul "B").

B: PROG     ASM : DUMP     ASM : PRO      ASM : TEST     ASM

B>A:      (reasignarea discului "A" ca disc instalat)

A>
```

6.2.2 Comenzi CP/M

Sub CP/M, un program direct executabil se numeste "comanda". Exista doua tipuri de comenzi CP/M:

- comenzi rezidente;
- comenzi tranzitorii.

Comenzile rezidente reprezinta programe ce sint incluse in componenta CCP, in timp ce comenzile tranzitorii reprezinta programe ce se incarca de pe disc (sub controlul CCP) in zona TPA si apoi se lanseaza in executie.
Comenzile rezidente sint:

USER - precizarea numarului utilizatorului curent;
ERA - sterge fisierele specificate;
DIR - afiseaza la consola numele tuturor fisierelor din "director";
REN - redenumeste un fisier dat;
TYPE - afiseaza la consola continutul unui fisier sursa ASCII;
SAVE - salveaza continutul memoriei intr-un fisier.

Comenzile tranzitorii pot fi:

- programe utilitare din sistemul CP/M;
- programe traducatoare;
- programe pentru editare de texte;
- programe pentru editare de legaturi;
- programe-utilizator de aplicatii.

Orice comanda se introduce de la consola, existind o serie de facilitati pentru editarea de linii (vezi sectiunea 2.4 "caractere de editare").

6.2.3 Identificarea fisierelor

Un specificator de fisier identifica un fisier sau un grup de fisiere de pe un disc CP/M. Un specificator poate fi individual sau multiplu. Un specificator-individual identifica

un singur fisier, in timp ce un specificator-multiplu poate fi satisfacut de mai multe fisiere diferite.
Forma generala a unui specificator de fisier este:

[dispozitiv:]nume-fisier[.extensie]

unde:

-dispozitiv este numele unitatii de discuri (literele A-P) pe care se gaseste fisierul (implicit acest nume este numele discului instalat).
-nume-fisier este numele fisierului alcatuit din maximum 8 caractere alfanumerice si speciale, cu exceptia urmatoarelor caractere: **<, >, ., ,, ;, :, =, ?, *, [,], ^, spatiu.**
-extensie identifica de obicei tipul fisierului si poate fi alcatuit din maximum 3 caractere alfanumerice si speciale, cu exceptia caracterelor interzise pentru specificarea numelui-fisierului.

Desi extensia unui fisier este optionala, exista o serie de extensii standard folosite de programele tranzitorii ale CP/M; de exemplu extensia "ASM" este folosita pentru identificarea unui fisier sursa in limbaj de asamblare, iar extensia "COM" indica un program absolut direct executabil.

Un specificator-multiplu are o forma similara unui specificator-individual, cu exceptia faptului ca pot fi folosite, pentru nume-fisier si extensie, caracterele "?" si "*".

Caracterul "?" inlocuieste practic, in pozitia respectiva, orice caracter dintr-un specificator de fisier. Astfel, specificatorul-multiplu:

X?Z.C?M

este satisfacut de urmatoarele specificatoare-individuale de fisier:

XYZ.COM si X3Z.COM

Caracterul "*" inlocuieste oricite si orice caractere din numele-fisierului si/sau extensie. Astfel, specificatorul-multiplu:

.

identifica toate fisierele cu orice nume-fisier si orice extensie de pe discul instalat si este echivalent cu:

????????.???

in timp ce

pppppppp.* si *.sss

sint echivalente cu:

pppppppp.??? si ????????.sss

In continuare se prezinta citeva exemple de specificatori de fisiere:

- specificatori-individuali de fisiere:

```
B:X       XYZ      . B:GAMA        GAMA.ASM
  X.Y   A:XYZ.COM     GAMA.1     A:P2.BAK
```

- specificatori-multipli de fisiere:

```
B:X.A?M         B:*.ASM
  *.PRN         B:R??A.*
```

NOTA: Daca intr-un specificator de fisier se folosesc literele minuscule ale alfabetului, acestea sint automat transformate in majuscule de catre componenta CCP.

6.2.4 Caractere de editare

CCP contine o serie de functii de editare de linii care sint activate prin urmatoarele caractere (denumite "caractere de editare linii").

RUBOUT - sterge din "buffer"-ul de intrare si reda in ecou ultimul caracter introdus de la consola.

CTRL/U - sterge integral linia introdusa de la consola.

CTRL/X - identic cu CTRL/U.

CTRL/R - tipareste la consola pe linia imediat urmatoare continutul curent al "buffer"-ului de intrare. Prin acest caracter se poate vizualiza continutul curent al unei linii in care s-au efectuat corectii prin RUBOUT (DEL).

CTRL/E - indica sfirsitul fizic al unei linii; cursorul se pozitioneaza pe inceputul liniei dar linia nu se transmite decit atunci cind se tasteaza (CR).

CTRL/Z - indica sfirsitul unui fisier introdus de la consola (se utilizeaza in comenzile PIP si ED).

CTRL/H - sterge din "buffer"-ul de intrare si de pe ecranul terminalului ultimul caracter introdus.

CTRL/J - este echivalent unui caracter (LF) si reprezinta sfirsitul unei linii.

CTRL/M - este echivalent unui caracter (CR) si reprezinta sfirsitul unei linii.

CTRL/P - permite ca tot ceea ce se introduce din acel moment de la consola, pina la un nou CTRL/P sa fie transmis si la perifericul tip **LST:** curent (vezi comanda STAT).

CTRL/S - opreste temporar un proces de afisare de informatii la consola. Procesul de afisare se reia atunci cind se introduce orice caracter de la consola (se recomanda introducerea unui alt caracter CTRL/S). Caracterul CTRL/S se foloseste pentru a putea urmarii o succesiune de imagini-ecran care se deruleaza foarte rapid pe consola.

Caracterele CTRL/x se introduc prin apasarea simultana pe tasta CTRL si pe tasta x.

Liniile de comanda preluate de CCP pot contine pina la 128 de caractere; ele nu sint interpretate de catre componenta CCP decit dupa ce s-a actionat tasta (CR).

6.3 Descrierea comenzilor CP/M rezidente

Orice comanda rezidenta se apeleaza prin:

nume-comanda[argumente](CR)

Un apel de tipul: "dispozitiv:nume-comanda[argumente](CR)" este eronat.

Se pot folosi pentru numele de comenzi si pentru argumente atit caracterele majuscule cit si cele minuscule, tinind cont de faptul ca CCP le transforma automat pe acestea din urma in majuscule.

6.3.1 Comanda USER

Forma generala este:

USER n(CR)

unde "n" este o valoare intreaga, cuprinsa intre 0 si 15. Comanda USER permite precizarea numarului utilizatorului curent, tinind cont de faptul ca sistemul CP/M gestioneaza fisiere apartinind mai multor utilizatori. Toate fisierele de pe un disc CP/M sint gestionate prin intermediul unui fisier "director" UNIC la nivelul discului, dar un utilizator are acces doar la fisierele corespunzatoare numarului sau.

La initializarea sistemului CP/M, utilizatorul curent este utilizatorul cu numarul 0. Prin comanda USER se poate, in orice moment, selecta un alt numar utilizator, prin aceasta asigurindu-se accesul la fisierele proprii acestuia. Numarul utilizatorului curent este pastrat pina la o noua comanda USER, sau pina la o initializare a sistemului CP/M (care va fixa ca utilizator curent utilizatorul 0). Precizarea numarului utilizatorului curent este esentiala pentru lucrul cu celelalte comenzi rezidente, avind in vedere faptul ca ele actioneaza la nivelul utilizatorului curent (deci numai asupra fisierelor proprii acestuia).

6.3.2 Comanda ERA

Forma generala:

ERA specificator-fisier(CR)

unde specificator-fisier poate fi un specificator-individual sau un specificator-multiplu.

Comanda ERA (ERAse) realizeaza stergerea unuia sau mai multor fisiere specificate, apartinind utilizatorului curent. Odata cu stergerea fisierelor din "directorul" discului, este eliberat si spatiul ocupat de acestea.

Exemple de utilizare a comenzii ERA:

ERA X.Y - fisierul cu numele X.Y de pe discul instalat este sters din "director" si spatiul ocupat de acesta este eliberat.

ERA X.* - toate fisierele cu nume X indiferent de extensie sint sterse de pe discul curent.

ERA B:*.PRN - sterge toate fisierele de pe discul din unitatea "B" care au extensia PRN.

ERA *.* - sterge toate fisierele de pe discul instalat, indiferent de nume-fisier si extensie. In acest caz componenta CCP afiseaza la consola mesajul "ALL (Y/N)?". In cazul in care se raspunde "Y" functia este executata, in caz contrar ea este abandonata.

6.3.3 Comanda DIR

Are doua forme:

(1) **DIR dispozitiv:(CR)**

(2) **DIR specificator-fisier(CR)**

unde specificator-fisier poate fi un specificator-individual sau un specificator-multiplu.

Comanda DIR (DIRectory) afiseaza la consola numele tuturor fisierelor de pe discul specificat - forma (1) - sau ale tuturor fisierelor care corespund specificatorului-fisier - forma (2). Comanda DIR se refera numai la fisierele corespunzatoare utilizatorului curent.

Exemple de utilizare a comenzii DIR:

DIR - afiseaza numele fisierelor de pe discul instalat.

DIR *.ASM - afiseaza numele tuturor fisierelor de pe discul instalat, care au extensia "ASM".

DIR B: - afiseaza numele tuturor fisierelor de pe discul din unitatea "B".

Comanda DIR afiseaza la consola mesajul "NO FILE" daca nu gaseste nici un fisier care sa corespunda specificatorului din comanda.

6.3.4 Comanda REN

Are doua forme:

REN specificator-individual(1)=specificator-individual(2)(CR)

REN nume-fisier-nou=nume-fisier-vechi(CR)

Comanda REN (REName) permite schimbarea numelui unui fisier, respectiv fisierul care corespunde specificatorului-individual(2) va primi un nume conform specificatorului-individual(1).

REN accepta urmatoarele forme:

a) REN nume-fisier(1)[.extensie(1)]=nume-fisier(2)[.extensie(2)]
n-f e n-f e

b) **REN dispozitiv:n-f(1)[.e(1)]=n-f(2)[.e(2)](CR)**

c) **REN n-f(1)[.e(1)]=dispozitiv:n-f(2)[.e(2)](CR)**

d) **REN dispozitiv:n-f(1)[.e(1)]=dispozitiv:n-f(2)[.e(2)](CR)**

Prezenta unui singur nume de dispozitiv intr-o comanda REN (inainte sau dupa semnul "=") implica automat ca operatia de redenumire se va aplica asupra unui fisier existent pe acel dispozitiv. In cazul formei (d) dispozitivele prezente in comanda trebuie sa fie identice.

Daca pe discul implicat in comanda REN exista deja un fisier cu nume identic cu noul nume (specificator-individual(1)), la consola va apare mesajul:

FILE EXISTS

si nu se va efectua nici un fel de modificare.

Daca fisierul de redenumit (specificator-individual(2)) nu exista pe dispozitivul specificat, apare la consola mesajul:

NOT FOUND

Exemple de utilizare a comenzii REN:

REN A:X.ASM=Y.ASM - numele fisierului Y.ASM devine X.ASM, pe discul "A".

REN B:ZAP.BAS=ZOT.BAS - numele fisierului ZOT.BAS devine ZAP.BAS, pe discul "B".

REN B:A.ASM=B:A.BAK - fisierul A.BAK este redenumit A.ASM, pe discul "B".

6.3.5 Comanda TYPE

Forma generala:

TYPE specificator-individual(CR)

Comanda TYPE afiseaza la consola continutul unui fisier sursa ASCII. Comanda TYPE tine cont de prezenta tabulatorilor (CTRL/I), presupunind ca acestia sint prezenti din 8 in 8 coloane.

Exemple de utilizare a comenzii TYPE:

TYPE X.Y - afiseaza la consola continutul fisierului cu numele X.Y, de pe discul "A".

TYPE X.PLM - afiseaza la consola continutul fisierului cu numele X.PLM, de pe discul "A".

6.3.6 Comanda SAVE

Forma generala:

SAVE n specificator-individual(CR)

Comanda SAVE salveaza un numar "n" de pagini de memorie pe disc (o pagina de memorie are 256 octeti). Salvarea se face incepind de la adresa #100 (adresa de inceput a zonei TPA) intr-un fisier al carui nume corespunde specificatorului-individual. Spre exemplu, daca programul utilizatorului ocupa o zona cuprinsa intre adresele #100 si #2FF, in comanda SAVE trebuie specificate 2 pagini de memorie. Daca zona salvata contine un program executabil (in cod-masina), fisierul in care s-a facut salvarea poate fi incarcat si apoi executat. In urma executiei comenzii SAVE, zona de memorie salvata pe disc ramine nemodificata.

Exemple de utilizare a comenzii SAVE:

SAVE 3 X.COM - salveaza zona de memorie cuprinsa intre adresele #100 si #3FF intr-un fisier cu numele X.COM.

SAVE 4 B:X.Y - salveaza zona de memorie cuprinsa intre adresele #100 si #4FF intr-un fisier cu numele X.Y, de pe discul "B".

6.4 Descrierea comenzilor CP/M tranzitorii

Comenzile tranzitorii sint programe care se incarca de pe disc in zona TPA si apoi se executa. Structura generala a unei comenzi tranzitorii este:

[dispozitiv:]nume-comanda[argumente](CR)

unde:

dispozitiv - numele unitatii de discuri de pe care se va incarca in zona TPA programul indicat prin nume-comanda.

nume-comanda - este numele unui fisier disc care are extensie standard COM (aceasta extensie nu se specifica in nume-comanda).

argumente - parametri cu continut si sintaxa dependente de tipul comenzii.

In continuare se vor prezenta o serie de comenzi tranzitorii ce reprezinta programe utilitare de baza din sistemul CP/M. Pentru simplificarea scrierii sintaxei proprii fiecarei comenzi tranzitorii, in cele ce urmeaza se va omite intentionat numele dispozitivului (discului) pe care se gaseste comanda tranzitorie, presupunindu-se ca aceasta este rezidenta pe discul curent instalat.

Trebuie subliniat faptul ca sistemul CP/M ofera si alte componente (comenzi tranzitorii), descrierea acestora facind obiectul unor manuale separate.

6.4.1 Comanda STAT

Forma generala:

STAT[argumente](CR)

Prezinta urmatoarele caracteristici:

- furnizeaza informatii statistice generale privind fisierele

stocate pe un disc si configuratia curenta a sistemului de I/E;
- permite modificarea atributelor unor fisiere/volume disc, precum si a configuratiei curente de I/E;
- dispune de autodocumentare.

Exista urmatoarele forme valide ale comenzii STAT:

(1) **STAT VAL:**(CR)

- afiseaza la consola sintaxa tuturor formelor VALide ale comenzii STAT (inclusiv toate asignarile de periferice acceptate).

Exemplu:

STAT VAL:

poate avea ca efect afisarea la consola a urmatoarelor mesaje:

```
Temp R/O Disk: d:=R/O
Set Indicator: d:filename.typ $R/O $R/W $SYS $DIR
Disk Status  : DSK: d:DSK
User Status  : USR:
Iobyte Assign:
CON: = TTY: CRT: BAT: UC1:
RDR: = TTY: PTR: UR1: UR2:
PUN: = TTY: PTR: UP1: UP2:
LST: = TTY: CRT: LPT: UL1:
```

(2) **STAT DEV:**(CR)

afiseaza la consola configuratia de I/E curenta sub forma:

"periferic-logic"="periferic-fizic"

unde:

"periferic-logic" defineste numele unui echipament logic, care poate fi:

CON: echipament tip consola (conversational);
RDR: echipament tip "reader" (cu functii de intrare de pe suport extern);
PUN: echipament tip "punch" (cu functii de iesire pe suport extern);
LST: echipament tip "list" (cu functii de listare).

"periferic-fizic" defineste numele echipamentului fizic, asociat echipamentului logic respectiv. Se pot folosi urmatoarele denumiri de echipamente fizice:

TTY: consola sistem;
CRT: consola tip display;
BAT: consola "batch" (functiile de intrare sint satisfacute de dispozitivul **RDR:** curent iar cele de iesire de dispozitivul **LST:** curent;
UC1: consola utilizator (periferic nestandard);
PTR: cititor de banda de hirtie;
UR1: periferic de intrare 1 nestandard;
UR2: periferic de intrare 2 nestandard;

LPT: imprimanta;
UL1: periferic de listare nestandard;
PTP: perforator banda de hirtie;
UP1: periferic de iesire nestandard;
UP2: periferic de iesire nestandard.

Exemplu:

STAT DEV:

poate avea ca efect afisarea mesajelor:

```
CON: is TTY:
RDR: is TTY:
PUN: is PTP:
LST: is CRT:
```

NOTA: Comanda **STAT DEV:** afiseaza la consola starea curenta a octetului IOBYTE (de la adresa 0003H). Configuratia binara asociata acestui octet este interpretata standard de catre sistemul CP/M conform schemei din ANEXA 1. Intrucit structura octetului IOBYTE este dependenta de versiunea de BIOS, informatiile furnizate de comanda STAT trebuie adaptate la structura curenta a octetului IOBYTE, corespunzatoare microcalculatorului gazda.

(3) **STAT USR:(CR)**

afiseaza la consola numarul utilizatorului curent (Active User) si numerele utilizatorilor care au fisiere pe discul instalat (Active Files).

Exemplu:

STAT USR:

poate avea ca efect afisarea mesajelor:

```
Active User : 0
Active Files: 0
```

(4) **STAT [dispozitiv:]DSK:(CR)**

afiseaza la consola caracteristicile tuturor volumelor disc CP/M prezente in toate unitatile de discuri operationale in momentul respectiv.

Exemplu:

STAT DSK:

poate avea ca efect afisarea mesajelor:

```
   A: Drive Caracteristics
5616: 128 Byte Record Capacity
 702: Kilobyte Drive  Capacity
 128: 32  Byte Directory Entries
 128: Checked  Directory Entries
 128: Records/ Extent
  16: Records/ Block
  36: Sectors/ Track
   4: Reserved Tracks
```

```
   B: Drive Caracteristics
5616: 128 Byte Record Capacity
 702: Kilobyte Drive  Capacity
 128: 32  Byte Directory Entries
 128: Checked  Directory Entries
 128: Records/ Extent
  16: Records/ Block
  36: Sectors/ Track
   4: Reserved Tracks
```

Informatiile listate au urmatoarea semnificatie:

- un volum disc are 160 de piste (dispuse cite doua - fata/spate - pe cite un cilindru) cu 36 de sectoare logice pe pista (1 sector logic = o inregistrare = 128 Bytes) deci 5760 sectoare (5760=160*36);
- exista 4 piste rezervate pentru sistemul CP/M (pistele 0, 1, 2 si 3, care tin de primii doi cilindri);
- unitatea de alocare pe disc ("blocul de alocare") are o capacitate de 16 inregistrari (16 * 128 B = 2048 B = 2 KB); B este prescurtarea lui Bytes;
- capacitatea utila a unui volum disc (utilizabila pentru fisierele de date si fisierul "director") este de 5616 inregistrari (de exemplu 698 KB = 351 blocuri de alocare = 5616 sectoare logice);
- dimensiunea unei "extensii logice" a unui fisier este de 128 inregistrari (128 * 128 B = 16 KB);
- orice intrare in "director" are o capacitate de 32 B;
- "directorul" unui volum disc are maximum 128 de intrari (de exemplu ocupa maximum 128 * 32 B = 32 inregistrari).

Pentru informatii suplimentare privind organizarea discului CP/M, vezi "Manualul de interfata CP/M".

Daca dispozitivul este precizat, se vor afisa doar informatii referitoare la discul din unitatea respectiva.

Exemplu:

STAT B:DSK:

```
   B: Drive Caracteristics
5616: 128 Byte Record Capacity
 702: Kilobyte Drive  Capacity
 128: 32  Byte Directory Entries
 128: Checked  Directory Entries
 128: Records/ Extent
  16: Records/ Block
  36: Sectors/ Track
   4: Reserved Tracks
```

(5) STAT specificator-fisier [$S](CR)

afiseaza la consola urmatoarele informatii privind fisierele care corespund specificatorului din comanda:

Size - numarul de ordine al ultimei inregistrari scrise in fisier (exprimat in zecimal); acest cimp apare listat numai daca in comanda STAT s-a specificat parametrul "$S".

Recs - numarul de inregistrari (de cite 128 B fiecare) ocu-

pate de fisier (exprimat in zecimal); acest cimp insumeaza numarul de inregistrari din cadrul fiecarei "extensii logice" a fisierului.

Bytes - numarul de Kiloocteti alocati fisierului; intrucit spatiul-disc minim ce poate fi alocat pentru un fisier este egal cu 1 bloc de alocare (1KB), acest cimp indica numarul de blocuri de alocare asociate fisierului.

Ext - numarul de "extensii logice" asociate fisierului; reprezinta numarul de "intrari" in "director" rezervate pentru fisierul respectiv.

Acc - atributele fisierului, care pot fi:

R/O - protejat la scriere;
R/W - neprotejat la scriere.

Daca fisierul are atributul SYS, numele fisierului este afisat intre paranteze.

Exemplu:

STAT *.*

poate avea ca efect afisarea mesajelor:

```
 Recs      Bytes     Ext  Acc
   58         8k       1  R/W A:AMS00.DOC
   41         6k       1  R/O (A:STAT.COM)
  128        16k       2  R/W A:WS.COM
  218        28k       2  R/O A:WSMSGS3.OVR
  266        34k       3  R/O A:WSOVLY13.OVR
Bytes Remaining On A: 606k
```

iar

STAT *.* $S

```
 Size      Recs      Bytes     Ext  Acc
   58         2         8k       1  R/W A:ASM00.DOC
   41        41         6k       1  R/O (A:STAT.COM)
  128       128        16k       2  R/W A:WS.COM
  218       218        28k       2  R/O A:WSMSGS3.OVR
  266       266        34k       3  R/O A:WSOVLY13.OVR
Bytes Remaining On A: 606k
```

Pentru fisierele create secvential, valorile din cimpurile "Size" si "Recs" sint identice si reprezinta lungimea reala a fisierului. Pentru fisierele create in acces direct cimpul "Recs" indica lungimea reala a fisierului (numarul de inregistrari ocupate de catre fisier), iar cimpul "Size" indica lungimea virtuala a fisierului (numarul ultimei inregistrari scrise in fisier).

(6) **STAT specificator-fisier $atribut(CR)**

permite modificarea (setare/resetare) atributelor fisierelor care corespund specificatorului-fisier din comanda.

Atributul poate fi:

R/O - fisier protejat la scriere (read only);

R/W - fisier care permite acces in scriere si citire (read/write); acesta este atributul pe care il au initial toate fisierele;

SYS - fisier invizibil (informatii privind acest fisier nu

se pot obtine prin comanda DIR, ci numai prin comanda STAT);

DIR - fisier vizibil (reversul atributului SYS). Fisierele sint create implicit cu atributul DIR.

Exemplu:

STAT WS.COM $R/O

are ca efect:

A:WS.COM set to R/O.

(7) **STAT dispozitiv:=R/O(CR)**

are ca efect declararea temporara a discului din unitatea specificata ca disc de tip R/O (read only). Comanda este efectiva pina la o initializare sau o reincarcare a sistemului CP/M. Un disc declarat R/O permite doar operatii de citire.

Exemplu:

STAT B:=R/O

(8) **STAT [dispozitiv:](CR)**

afiseaza la consola numarul de Kiloocteti disponibili pe discul din unitatea specificata. Daca dispozitivul nu este specificat, comanda analizeaza discurile prezente in toate unitatile operationale in acel moment, si afiseaza:

- numarul unitatii;
- atributul asociat volumului (R/O sau R/W);
- spatiul disponibil pe fiecare disc (in Kocteti).

Exemple:

STAT B:

Bytes Remaining On B: 346k

STAT

A: R/W, Space: 606k
B: R/W, Space: 346k

(9) **STAT pli=pfi[,pl2=pf2,...](CR)**

realizeaza asignarea unui periferic fizic unui periferic logic. Semnificatiile argumentelor sint urmatoarele:

-pli - reprezinta numele unui periferic logic (CON:, PUN:, RDR:, LST:);
-pfi - reprezinta numele unui periferic fizic (TTY:, LPT:, PTP:, etc.).

Comanda are ca efect modificarea configuratiei de I/E curente, respectiv ea permite sa se asigneze unui periferic logic un anumit periferic fizic.

Exemplu:

STAT LST:=TTY:

In utilizarea acestei comenzi trebuie tinut cont de diferentele existente intre "structura standard a octetului IOBYTE" utilizata de sistemul CP/M si "structura octetului IOBYTE specifica versiunii de BIOS existenta in microcalculatorul gazda".

6.4.2 Comanda LOAD

Forma generala:

LOAD specificator-individual(CR)

Se aplica numai asupra unui fisier rezident pe disc de tip "HEXA" (program cod-masina in format hexa) care are extensie standard "HEX". Citeste fisierul care corespunde specificatorului din comanda si produce, pe disc, un fisier imagine-memorie (program cod-obiect absolut, direct executabil). Intrucit trateaza numai fisiere care se identifica prin:

[dispozitiv:]nume-fisier.HEX

este suficient ca specificatorul individual sa aiba forma:

[dispozitiv:]nume-fisier.

Programul LOAD creeaza automat, pe disc, un fisier imagine-memorie identificabil prin:

[dispozitiv:]nume-fisier.COM

unde "dispozitiv" si "nume-fisier" sint identice cu specificatorul din comanda. Intrucit rezultatul actiunii programului LOAD este un program cod-masina direct executabil, acestui program i se asociaza extensia COM, el devenind astfel o comanda tranzitorie CP/M.

Fisierul rezultat in urma comenzii LOAD poate fi incarcat in memorie si executat prin comanda:

[dispozitiv:]nume-fisier(CR)

sub controlul componentei CCP (dupa aparitia textului "dispozitiv>").

In general, componenta CCP citeste numele X care urmeaza promptérului ">" si il compara cu numele comenzilor CP/M rezidente. Daca X nu corespunde unei comenzi rezidente, componenta CCP cauta in "directorul" discului specificat un fisier cu numele **X.COM**. Daca il gaseste, atunci incarca programul cod-masina prezent in acel fisier in zona TPA si il lanseaza in executie. Rezulta ca pentru a obtine un program cod-obiect absolut este suficient sa se aplice o singura data comanda LOAD asupra unui fisier hexa.

Dar ce este un fisier hexa ? Este un fisier care prezinta urmatoarele caracteristici:

- contine blocuri (inregistrari) in format cod-masina hexazecimal (rezultat, de exemplu, dintr-o executie a programului ASM);
- are ca adresa de incarcare in memorie adresa 100H (adresa

de inceput a zonei TPA);
- are adrese de incarcare la nivel de bloc ordonate crescator.

Formatul general al unui fisier hexa este:

:xxyyyyzzaaaa...aabb

unde semnificatia simbolurilor este:

:	- marcajul de inceput de bloc (inregistrare);
xx	- lungimea blocului (in octeti);
yyyy	- adresa de incarcare in memorie a blocului (daca blocul este de date) sau adresa de lansare automata in executie a programului (daca blocul este bloc EOF);
zz	- tipul blocului (00H=bloc de date; 01H=bloc EOF);
aa...a	- continutul blocului;
bb	- cifra de control la nivel de bloc.

Daca adresele de incarcare ale blocurilor nu sint strict succesive (exista zone de memorie neutilizate) atunci comanda LOAD va umple automat zonele neocupate (neutilizate) cu zero, fisierul X.COM devenind astfel contiguu.

Comanda LOAD trebuie utilizata numai pentru crearea de fisiere (comenzi tranzitorii) standard CP/M, care se incarca si se executa numai in zona TPA. Pentru programe care ocupa alte regiuni de memorie decit zona TPA se va folosi, pentru incarcarea lor, programul DDT sau ZSID.

Exemplu:

LOAD B:BETA

transforma fisierul BETA.HEX de pe discul "B" in program direct executabil cu numele de BETA.COM (tot pe discul "B").

6.4.3 Comanda SYSGEN

Prezinta doua forme:

(1) **SYSGEN(CR)**
(2) **SYSGEN specificator-individual(CR)**

La calculatorul Tim-S Plus sistemul de operare CP/M recunoaste volume disc (diskete) care:

- sint formatate (premarcate) IBM (cu numerotare secventiala a sectoarelor fizice [un sector fizic = 512 B = 4 sectoare logice], de la 1 la 9, in cadrul fiecarei piste);
- contin componentele de baza ale sistemului CP/M: BDOS, BIOS, CCP.

Comanda SYSGEN nu realizeaza si premarcarea suportului magnetic. Daca acest lucru este necesar, se foloseste o alta comanda tranzitorie specifica configuratiei hardware.

Comanda SYSGEN (forma (1)), incarca in memorie de pe un disc CP/M initializat deja, primele 4 piste (0, 1, 2 si 3) care contin componentele BDOS, BIOS si CCP si transfera aceasta imagine-memorie pe discul de initializat. Comanda SYSGEN (forma (2))

incarca in memorie fisierul indicat prin "specificator individual" (fisier ce poate contine o imagine memorie CP/M salvata pe disc cu ajutorul comenzii "SAVE kk CPMxx.COM(CR)") si transfera aceasta imagine memorie pe discul de initializat. Comanda SYSGEN se apeleaza prin:

(1) **SYSGEN(CR)**

dupa care urmeaza un dialog, cu utilizatorul, de tipul:

SYSGEN VERSION m.m

mesaj prin care comanda se prezinta (m.m este numarul de versiune). Urmeaza:

SOURCE DRIVE NAME(OR RETURN TO SKIP)

mesaj la care utilizatorul trebuie sa raspunda cu numele unei unitati de disc (una din literele A - B) care va contine un disc cu sistemul de operare CP/M (un disc deja initializat). Uzual se foloseste discul "A". Daca in memorie exista deja o copie a CP/M-ului, datorata unei comenzi MOVCPM, este suficienta tastarea caracterului (CR). Dupa tastarea numelui unitatii de disc "x", programul SYSGEN va afisa la consola mesajul:

SOURCE ON x THEN TYPE RETURN.

Prin acest mesaj se solicita introducerea in unitatea de disc "x" a unui disc initializat deja, ce contine sistemul de operare CP/M. La terminarea operatiei se va tasta la consola caracterul (CR). Ca urmare a acestui raspuns, sistemul CP/M va fi copiat in memorie, terminarea operatiei de copiere fiind marcata de mesajul:

FUNCTION COMPLETE

Urmeaza apoi un nou mesaj, prin care se solicita utilizatorului precizarea numelui unitatii de disc in care se va afla discul ce trebuie initializat:

DESTINATION DRIVE NAME (OR RETURN TO REBOOT)

In conditiile in care nu se mai doreste initializarea unui alt volum disc, se va tasta (CR), sistemul reinitializindu-se prin intermediul discului "A". Dupa introducerea numelui unitatii de disc "x", programul SYSGEN va afisa la consola mesajul:

DESTINATION ON "x" THEN TYPE RETURN

Dupa aparitia acestui mesaj, se asteapta introducerea de catre utilizator, in unitatea de disc "x", a volumului disc de initializat. La terminarea operatiei se va tasta la consola caracterul (CR). In continuare pe noul disc va fi copiat sistemul de operare CP/M din memorie, finalul operatiei fiind marcat de mesajul:

FUNCTION COMPLETE

Programul SYSGEN permite initializarea succesiva a mai multor discuri, mesajul DESTINATION DRIVE NAME (OR RETURN TO REBOOT) repetindu-se pina la tastarea caracterului (CR). Dupa o comanda SYSGEN aplicata asupra unui disc vid (doar formatat), pe disc se

vor gasi componentele BDOS, BIOS si CCP, singurele comenzi disponibile fiind comenzile CP/M rezidente. Daca un disc supus actiunii comenzii SYSGEN continea deja o serie de fisiere CP/M, prin comanda SYSGEN acestea nu vor fi afectate.
NOTA: Sistemul CP/M accepta in lucru cu unitatile "B" - "P" si discuri neinitializate cu comanda SYSGEN, ci doar formatate. Un disc neinitializat nu va putea fi niciodata utilizat in unitatea "A" (unitate rezervata pentru initializarea sistemului CP/M).

Exemple:

```
SYSGEN
SYSGEN VER 2.0
SOURCE DRIVE NAME (OR RETURN TO SKIP)A
SOURCE ON A, THEN TYPE RETURN
FUNCTION COMPLETE
DESTINATION DRIVE NAME (OR RETURN TO REBOOT)B
DESTINATION ON B, THEN TYPE RETURN
FUNCTION COMPLETE
DESTINATION DRIVE NAME (OR RETURN TO REBOOT)

SYSGEN CPM53.COM
SYSGENVER2200
DESTINATION DRIVE NAME (OR RETURN TO REBOOT)B
DESTINATION ON B, THEN TYPE RETURN
FUNCTION COMPLETE
DESTINATION DRIVE NAME (OR RETURN TO REBOOT)
```

6.4.4 Comanda DUMP

Forma generala:

DUMP specificator-individual(CR)

Programul DUMP afiseaza la consola, in format hexazecimal, continutul unui fisier corespunzator specificatorului individual. Continutul fisierului este afisat pe linii de 16 octeti, fiecare linie fiind precedata, la stinga, de adresa absoluta a blocului respectiv (exprimata tot in format hexazecimal). Afisarile prea lungi pot fi intrerupte tastind de la consola caracterul RUBOUT (DEL).

6.4.5 Comanda SUBMIT

Forma generala:

SUBMIT specificator-individual [par1 par2... parn](CR)

Comanda SUBMIT asigura lucrul in mod "batch" cu sistemul CP/M, permitind preluarea comenzilor CP/M dintr-un fisier existent pe disc si nu de la consola.
Fisierul corespunzator "specificatorului" din comanda, numit si "fisier de comenzi", este un fisier cu extensie obligatorie "SUB" rezident pe discul instalat. El este creat cu ajutorul editorului de texte (ED) si contine secvente de comenzi CP/M parametrizate sau nu. In fisierul de comenzi, parametrii sint "parametri formali", identificati prin simbolurile:

$1

```
$2
$3
.
.
.
$n
```

iar in comanda SUBMIT, parametrii par1,...parn sint "parametri actuali", care vor inlocui, in ordine, parametrii formali din fisierul de comenzi; (exemplu: par1 va inlocui pe $1, s.a.m.d.

Daca numarul de parametri formali nu este egal cu numarul de parametri reali, functia SUBMIT este intrerupta, iar la consola va apare un mesaj de eroare. Comanda SUBMIT creeaza pe discul instalat un fisier intermediar cu numele "$$$.SUB", fisier care contine comenzile CP/M din "fisierul de comenzi" cu parametrii actuali. Cind sistemul se reinitializeaza (la sfirsitul comenzii SUBMIT), fisierul intermediar "$$$.SUB" este citit de componenta CCP ca sursa de intrare, in locul consolei. Daca functia SUBMIT este efectuata pe un alt disc decit cel din unitatea "A", comenzile nu vor fi tratate decit atunci cind discul va fi pus in unitatea de disc "A" si sistemul va fi reinitializat.

Utilizatorul poate opri in orice moment executia comenzii SUBMIT sau a comenzilor din fisierul intermediar, tastind "RUBOUT" (DEL). In acest caz fisierul "$$$.SUB" va fi sters, iar comenzile ulterioare vor fi preluate de la consola. Tratarea comenzilor este de asemenea abandonata daca componenta CCP detecteaza o eroare in una din comenzi.

Pentru introducerea simbolurilor "$" intr-un fisier SUBMIT, utilizatorul poate tasta "$$" care se transforma intr-un "$" unic in interiorul fisierelor de comenzi. Mai mult, simbolul "^" poate preceda un caracter alfabetic "x", ceea ce creeaza un caracter unic CTRL/x in interiorul fisierului.

Ultima comanda dintr-un fisier de comenzi poate apela un alt fisier de comenzi, permitind astfel lucrul cu fisiere de comenzi inlantuite.

Exemplu: presupunind ca fisierul ASMBL.SUB exista pe disc si contine comenzile:

```
ASM $1
DIR $1.*
ERA *.BAK
PIP $2:=$1.PRN
ERA $1.PRN
```

si ca a fost introdusa de catre operator comanda:

```
SUBMIT ASMBL X PRN(CR).
```

Programul SUBMIT va citi fisierul ASMBL.SUB si va inlocui cu X peste tot unde apare $1 si cu PRN peste tot unde apare $2. Se va crea astfel fisierul intermediar $$$.SUB care va avea urmatorul continut:

```
ASM X
DIR X.*
ERA *.BAK
PIP PRN:=X.PRN
ERA X.PRN
```

iar comenzile din acest fisier intermediar vor fi executate sec-

vential de catre componenta CCP.
Nota: Comanda SUBMIT poate utiliza si un fisier de comenzi care se gaseste pe o alta unitate de disc diferita de unitatea "A" (cu precizarea numelui unitatii in "specificatorul-individual") dar comenzile din fisierul intermediar se vor executa numai daca se gasesc pe discul din unitatea "A".

6.4.6 Comanda XSUB

Permite ca o serie de programe care solicitau informatii (comenzi), introduse de la consola, sa primeasca aceste date dintr-un fisier pe disc (fisier de tip SUBMIT). Comanda se foloseste ca prima linie intr-un fisier SUBMIT si ea face ca toate functiile de "citire buffer consola" sa fie inlocuite cu citiri de inregistrari din fisierul SUBMIT. De exemplu, fisierul TEST.SUB poate contine urmatoarele linii:

```
XSUB
DDT
I $1.HEX
R
GO
SAVE 1 $2.COM
```

Un apel de tipul:

```
SUBMIT TEST X Y(CR)
```

conduce la executia secventiala a urmatoarelor operatii:

- incarcarea programului XSUB si lansarea lui in executie;
- incarcarea programului DDT si lansarea lui in executie;
- transmiterea comenzilor:
  ```
  IX.HEX
  R
  GO
  ```
  pentru programul DDT.
- preluarea controlului sistemului de catre componenta CCP (datorita comenzii GO);
- executia comenzii **SAVE 1 Y.COM.**

NOTA: Programul XSUB ramine activ pina la o "initializare" a sistemului CP/M, el putind fi folosit in alte fisiere SUBMIT fara a fi reapelat.

6.4.7 Comanda MOVCPM

Forma generala:

```
MOVCPM [par1][par2](CR)
```

Programul MOVCPM permite utilizatorului sa reconfigureze sistemul de operare CP/M pentru a lucra cu orice dimensiune particulara de memorie. Pot fi dati doi parametri optionali pentru a indica:

- dimensiunea memoriei pentru care se genereaza noul sistem;
- actiunea care se va efectua la sfirsitul comenzii MOVCPM.

Daca primul parametru este omis sau este un caracter "*", programul MOVCPM va reconfigura sistemul CP/M in functie de dimensiunea maxima a memoriei interne din sistemul gazda (numarul de Kiloocteti de memorie RAM contigua, incepind de la adresa #0000). Daca al doilea parametru este omis, la sfirsitul comenzii MOVCPM se va lansa in executie noul sistem CP/M generat, fara a fi insa inregistrat si pe disc. Daca se transmite - ca al doilea parametru - caracterul "*", sistemul CP/M generat va fi pastrat in memorie in vederea salvarii lui ulterioare pe disc, prin comanda SYSGEN sau SAVE.

Formele posibile pentru comanda MOVCPM sint:

(1) **MOVCPM(CR)**

reconfigureaza sistemul CP/M pentru o dimensiune de memorie egala cu dimensiunea maxima a memoriei din sistem si il lanseaza in executie (memoria este examinata incepind de la adresa #100). La sfirsitul relocarii, noul sistem este executat dar nu si inregistrat pe disc.

(2) **MOVCPM n(CR)**

genereaza un sistem CP/M reconfigurat pentru o memorie de "n" Kiloocteti (n trebuie sa fie cuprins intre 16 si 64) si il lanseaza in executie fara a-l salva pe disc.

(3) **MOVCPM * *(CR)**

construieste o imagine memorie relocata pentru configuratia de memorie curenta, dar lasa imaginea-memorie in memorie, pentru o viitoare operatie SYSGEN sau SAVE.

(4) **MOVCPM n *(CR)**

construieste o imagine-memorie relocata pentru un sistem cu "n" Kiloocteti si lasa aceasta imagine in memorie pentru o viitoare operatie SYSGEN sau SAVE.

Dupa o comanda MOVCPM forma (3) sau (4) se genereaza o noua versiune de CP/M in memorie, versiune ce poate fi salvata pe disc printr-o operatie SYSGEN sau SAVE. La consola va apare mesajul:

```
READY FOR "SYSGEN" OR
"SAVE kk CPMxx.COM"
```

unde "xx" este dimensiunea in Kocteti (adica kiloocteti) a memoriei pentru care a fost reconfigurat CP/M-ul, iar "kk" este dimensiunea imaginii sistemului intr-un fisier de tip .COM. Utilizatorul poate lansa atunci:

SYSGEN(CR)

pentru a incepe generarea sistemului. Dialogul in continuare este:

```
SOURCE DRIVE NAME (OR RETURN TO SKIP)
```

se raspunde cu (CR) pentru a evita operatia de citire CP/M de pe disc (deoarece sistemul este deja in memorie in urma comenzii MOVCPM).

DESTINATION DRIVE NAME (OR RETURN TO REBOOT)

se raspunde cu "B" pentru a scrie noul sistem pe discul din unitatea de disc "B".

DESTINATION ON B, THEN TYPE RETURN

se introduce in unitatea de disc "B" noul disc si se tasteaza (CR).

SAVE kk CPMxx.COM(CR)

Aceasta comanda este necesara cind operatia are loc intr-un cadru nestandard, in care componenta BIOS trebuie sa fie modificata pentru o configuratie particulara de periferice.

Exemple de comenzi valide ale programului MOVCPM:

MOVCPM 48(CR)

construieste o versiune de 48K a CP/M-ului si il lanseaza in executie.

MOVCPM 48 *(CR)

construieste in memorie o versiune de 48K a CP/M-ului in vederea inregistrarii ulterioare pe disc. Raspunsul este:

READY FOR "SYSGEN" OR
"SAVE" 32 CPM48.COM".

MOVCPM *(CR)

construieste o versiune de memorie maxima a CP/M-ului si il lanseaza in executie.

6.4.8 Comanda PIP

PIP este un program care implementeaza operatiile de baza pentru conversii de suport necesare pentru crearea, listarea, inregistrarea, copierea si concatenarea fisierelor pe disc. Se lanseaza in executie prin una din formele:

(1) **PIP(CR);**
(2) **PIP linie-de-comanda(CR).**

In ambele cazuri, programul PIP va fi incarcat in zona TPA si se va lansa in executie. In cazul (1) PIP citeste si executa mai multe linii de comanda introduse direct de la consola. PIP semnalizează faptul ca asteapta introducerea unei linii de comanda prin tiparirea la consola a prompterului "*". Programul PIP se termina atunci cind se introduce, pe post de linie de comanda, caracterul (CR). Forma (2) a comenzii PIP este echivalenta cu prima, cu exceptia faptului ca PIP preia si executa o singura linie de comanda dupa care actiunea sa se termina. In acest caz, PIP nu mai afiseaza la consola prompterul "*".

Formatul unei linii de comanda este:

destinatie=sursa-1,sursa-2,...sursa-n(CR)

unde:

destinatie - este numele unui fisier sau dispozitiv periferic care va receptiona datele;
sursa-1 - reprezinta unul sau mai multe fisiere sau
... dispozitive periferice, care se copiaza de la
sursa-n stinga la dreapta in fisierul/dispozitivul destinatie.

Atunci cind in linia de comanda se specifica mai multe fisiere sursa (n>1), se presupune ca fiecare din aceste fisiere contine numai caractere ASCII, sfirsitul lor fiind marcat de un "caracter de sfirsit de fisier standard" (CTRL/Z). Linia de comanda poate contine atit caractere minuscule, cit si majuscule, minusculele fiind automat transformate in majuscule. Lungimea ei poate fi de maxim de 128 caractere (se poate utiliza caracterul CTRL/E pentru a forta un [CR] fizic pentru liniile ce depasesc latimea consolei).

Fisierul destinatie si fisierele sursa pot fi definite prin specificatori-individuali precedati optional de numele unitatii de disc. In absenta numelui unitatii de disc, se considera implicit numele discului instalat.

Daca intr-o linie de comanda se specifica un singur fisier destinatie si mai multe fisiere sursa, aceasta echivaleaza cu o operatie de concatenare a fisierelor sursa, cu depunerea rezultatului in fisierul destinatie. In conditiile in care fisierul destinatie exista deja pe unitatea de disc specificata sau implicita, acesta este sters. Aparitia unei conditii de eroare inhiba operatia de stergere a fisierului destinatie, in cazul in care acesta exista deja.

Sa vedem citeva exemple de linii de comanda PIP valide.

```
X=Y(CR)
```

copiaza fisierul cu numele Y intr-un fisier cu numele X pe aceeasi unitate de disc. Fisierul Y ramine neschimbat.

```
X=Y,Z(CR)
```

concateneaza fisierele cu nume Y si Z si le copiaza intr-un fisier cu numele X pe aceeasi unitate de disc. Fisierele Y si Z ramin neschimbate.

```
X.ASM=Y.ASM,Z.ASM,W.ASM(CR)
```

creeaza fisierul X.ASM pornind de la concatenarea fisierelor Y, Z si W, toate de tip ASM.

```
B:A.U=B:B.V,C.W,D.X(CR)
```

concateneaza fisierul cu numele B.V de pe unitatea de disc "B" cu fisierele C.W si D.X de pe discul instalat si creeaza un fisier cu numele A.U pe unitatea de disc "B".

Comanda PIP accepta linii de comanda prescurtate in vederea conversiei de fisiere intre unitatile de disc. Iata citeva exemple de linii de comanda prescurtate:

```
x:=specificator-multiplu(CR);
x:=y:specificator-multiplu(CR);
specificator-individual=y:(CR);
x:specificator-individual=y:(CR).
```

Prima comanda copiaza toate fisierele de pe discul instalat care satisfac specificatorul-multiplu pe unitatea-X (X = A, B, ...P).

A doua comanda este echivalenta cu prima, cu deosebirea ca unitatea de disc pe care se gasesc fisierele sursa este Y:.

A treia comanda este echivalenta cu comanda:

specificator-individual=y:specificator-individual(CR)

care copiaza un fisier de pe unitatea de disc y: intr-un fisier pe discul instalat.

A patra comanda este echivalenta cu a treia cu deosebirea ca este specificata si unitatea de disc pe care se va gasi fisierul destinatie.

Trebuie remarcat faptul ca in toate cazurile fisierele sursa si destinatie trebuie sa fie diferite. Daca in comanda apare un specificator-multiplu, PIP listeaza, pe masura ce efectueaza operatiile de copiere, numele tuturor fisierelor transferate. Daca un fisier avind acelasi nume cu fisierul destinatie exista deja, acesta este sters si inlocuit cu fisierul copiat (daca operatia s-a desfasurat corect).

Urmatoarele comenzi PIP exemplifica operatiile valide de copiere a fisierelor de pe un disc pe altul:

B:=*.COM(CR)

copiaza toate fisierele de pe discul instalat care au extensia COM pe unitatea de disc B:.

A:=B:ZAP.*(CR)

copiaza toate fisierele de pe unitatea de disc B: care au numele ZAP si orice extensie pe unitatea de disc A:.

ZAP.ASM=B:(CR)

echivalent cu ZAP.ASM=B:ZAP.ASM.

B:ZOT.COM=A:(CR)

echivalent cu B:ZOT.COM=A:ZOT.COM.

B:=A:GAMMA.BAS(CR)

identic cu B:GAMMA.BAS=A:GAMMA.BAS.

Comanda PIP autorizeaza deasemenea utilizarea in linia de comanda a numelor de periferice fizice si logice acceptate de CP/M. Numele de periferice acceptate sint cele prezentate in cadrul comenzii STAT, precum si o serie de nume de periferice specifice comenzii PIP. Perifericele logice acceptate in cadrul comenzii STAT sint:

CON: echipament tip "consola";
RDR: echipament tip "cititor";
PUN: echipament tip "perforator";
LST: echipament tip "lista".

Nota: Utilizarea numelor pentru perifericele fizice trebuie facuta in concordanta cu versiunea de BIOS existenta in microcalculatorul gazda (vezi si observatiile din comanda STAT).

De retinut faptul ca periferiCul fizic "BAT:" nu este inclus, deoarece aceasta asignare nu este folosita decit pentru a arata ca unitatile **RDR:** si **LST:** trebuie sa fie utilizate pentru functiile de I/E ale consolei.

Unitatile CON, PUN, LST si RDR sint toate definite in cadrul componentei BIOS din CP/M si pot fi de asemenea modificate pentru orice configuratie de I/E particulara ("Maping"-ul perifericului fizic curent este definit de IOBYTE; vezi in "Manualul de interfata CP/M" - sectiunea 6.7 - studiul acestei functii). PeriferiCul destinatie trebuie sa fie capabil sa primeasca date (datele nu pot fi trimise la un cititor de cartele) iar perifericele de intrare trebuie sa fie capabile sa transmita date (de la un periferic de tip LST nu pot fi citite date).

Numele perifericelor suplimentare ce pot fi utilizate in comenzile PIP sint:

NUL: - trimite 40 de caractere null (caracterul ASCII "#00") la perifericul destinatie;

EOF: - trimite un "caracter de sfirsit de fisier standard CP/M" (CTRL/Z) la perifericul destinatie (caracterul este trimis automat la sfirsitul fiecarui transfer de date ASCII, realizat prin intermediul comenzii PIP);

INP: - este o sursa de intrare in comanda PIP speciala, ce poate fi inclusa chiar in programul PIP. PIP obtine date de intrare caracter cu caracter apelind cu CALL locatia #103, cu reintoarcerea datelor in locatia #109 (bitul de paritate trebuie sa fie zero);

OUT: - este un periferic destinatie special ce poate fi inserat chiar in comanda PIP. PIP apeleaza cu CALL locatia #106 cu datele in registrul C pentru fiecare caracter de transmis. De retinut ca locatiile de la #109 la #1FF, care apartin imaginii-memorie a programului PIP, nu sint utilizate, ele putind fi inlocuite cu rutine speciale utilizind, DDT-ul sau ZSID-ul;

PRN: - identic cu **LST:** cu deosebirea ca tine cont de TAB-uri (pozitionate din 8 in 8 coloane), numeroteaza liniile, efectueaza salturi la pagina noua dupa fiecare grup de 60 de linii tiparite si face un salt initial la pagina noua.

In cadrul comenzilor PIP pot apare atit nume de periferice, cit si nume de fisiere. In toate cazurile, perifericul specificat este citit pina la sfirsitul fisierului, marcat cu **CTRL/Z** pentru fisierele ASCII sau cu sfirsit real de fisier pentru fisierele pe disc non-ASCII. Datele de la fiecare periferic sau fisier sint concatenate de la stinga la dreapta pina ce ultima sursa de date a fost citita. Perifericul sau fisierul de destinatie este scris utilizind datele fisierelor sursa, la sfirsit adaugindu-se, pentru fisierele ASCII, un caracter de sfirsit de fisier (CTRL/Z). De retinut ca daca destinatia este un fisier pe disc, atunci se creeaza intii un fisier temporar (cu extensie **$$$**) si acest fisier isi schimba numele conform numelui fisierului destinatie numai dupa ce operatia de copiere s-a terminat normal. Fisierele cu extensia COM sint considerate intotdeauna ca fiind fisiere non ASCII. Operatia de copiere poate fi oprita in orice moment prin apasarea unei taste (ex: RUBOUT). PIP va raspunde cu mesajul "ABORTED" pentru a indica intreruperea operatiei. De retinut ca daca o operatie oarecare este oprita sau daca apare o eroare in timpul executiei, PIP suspenda toate comenzile in curs de asteptare in conditiile utilizarii comenzii SUBMIT.

Comanda PIP realizeaza o functie speciala daca destinatia este un fisier pe disc de tip HEX si daca sursa este o unitate periferica externa, cum ar fi cititorul de banda. In acest caz, programul PIP se asigura ca fisierul sursa contine un fisier in format hexa, cu valori hexazecimale permise si inregistrari de verificare. Cind este detectata o inregistrare incorecta, PIP transmite un mesaj de eroare la consola si asteapta o actiune de corectie (se va da banda inapoi cu aproximativ 30 de centimetri si se va reciti). Daca banda nu poate fi corect citita, se va continua operatia prin tastarea unui caracter (CR), inregistrarile neincluse fiind ulterior inserate cu **ED**. Daca perifericul sursa este **RDR:** atunci PIP accepta ca "sfirsitul de fisier" sa fie introdus de la consola (se va tasta CTRL/Z).

Exemple:

PIP LST:=X.PRN(CR)

copiaza fisierul **X.PRN** la dispozitivul de tip LST.

PIP(CR)

lansarea programului PIP in vederea executiei unor linii de comanda, semnalizate prin caracterul "*".

CON:=X.ASM,Y.ASM,Z.ASM(CR)

concateneaza trei fisiere de tip ASM si le copiaza la perifericul de tip **CON:**.

(CR)

un caracter (CR) incheie executia programului PIP.

PIP PUN:=NUL:,X.ASM,EOF:,NUL:(CR)

trimite 40 de null-uri la perifericul de tip **PUN:**, apoi copiaza la perifericul **PUN:** fisierul X.ASM, urmat de un sfirsit de fisier (CTRL/Z) si de alte 40 de null-uri.

Utilizatorul poate de asemenea specifica unul sau mai multi parametri PIP, introdusi intre paranteze drepte si separati sau nu prin spatii. Fiecare parametru influenteaza operatia de copiere, iar lista lor trebuie sa succeada imediat perifericul sau fisierul afectat. In general, fiecare parametru poate fi urmat optional de o valoare zecimala intreaga (cu exceptia parametrilor S si Q). Parametrii valizi care pot apare in cadrul unei comenzi PIP sint urmatorii:

B - specifica mod de transfer "bloc". Datele sint transferate in buffer de catre programul PIP, pina ce este receptionat de la perifericul sursa un caracter ASCII "X-OFF" (CTRL/S). Acest parametru permite transferul datelor de la un periferic de intrare care lucreaza la nivel de bloc (exemplu: banda magnetica, caseta magnetica). Dupa receptia unui caracter "X-OFF", programul PIP goleste bufferele discului si se pregateste sa primeasca alte date de intrare. Cantitatea de date inregistrate in buffer depinde de capacitatea memoriei sistemului gazda (PIP va emite un mesaj de eroare daca este depasita capacitatea buffer-ului);

Dn - sterge caracterele care depasesc coloana "n" in transferul datelor de la sursa la destinatie. Acest parametru este utilizat foarte frecvent pentru trunchierea liniilor lungi care sint trimise la o imprimanta sau consola;

E - transmite in ecou la consola toate operatiile de transfer, pe masura ce sint executate;

F - filtreaza avansurile de pagina (caracterele FORM-FEED) inserate intr-un fisier. Poate fi utilizat simultan si parametrul P pentru inserarea de caractere FORM-FEED;

H - transfera date hexa: verifica daca datele supuse transferului sint in format hexa. Caracterele neesentiale care se gasesc intre inregistrari hexa sint sterse in timpul copierii. Consola va fi solicitata pentru realizarea de corectii in cazul aparitiei de erori;

I - ignora inregistrarile de tip ":00" in cadrul transferului de fisiere in format hexa. Parametrul I semnifica automat si parametrul H;

L - transforma majusculele in minuscule;

N - adauga un numar de linie la fiecare linie transferata in fisierul destinatie. Prima linie se numeroteaza cu 1 si incrementarea se face permanent cu 1. Zerourile nesemnificative sint suprimate, iar numarul de linie este urmat de caracterul ":". Daca se specifica parametrul N2, atunci vor fi incluse si zerourile nesemnificative si, de asemenea, va fi inserat si un caracter TAB dupa numarul de linie. Caracterul TAB va fi tratat daca se foloseste parametrul T;

O - permite transferul de fisiere obiect (non ASCII); sfirsitul normal de fisier CP/M este ignorat;

Pn - include salturi la pagina noua dupa fiecare grup de "n" linii (cu un salt de pagina initial). Daca n=1 sau este omis, vor avea loc salturi de pagina dupa fiecare grup de 60 de linii. Daca este utilizat parametrul F, suprimarea caracterelor FORM-FEED se face inainte ca noile salturi de pagina sa fie inserate;

Qsir^z - intrerupe operatia de copiere dintr-un fisier sau de la un dispozitiv de intrare, atunci cind este intilnit sirul de caractere "sir" (terminat prin CTRL/Z);

Ssir^z - incepe copierea din fisierul sursa sau de la dispozitivul de intrare atunci cind intilneste sirul de caractere "sir" (terminat prin CTRL/Z). Parametrii S si Q pot fi utilizati pentru a copia o portiune particulara a unui fisier (de exemplu o subrutina). Sirul de caractere "sir" este supus intotdeauna operatiei de copiere (indiferent de camanda Q sau S).

Nota: Sirurile de caractere "sir" care urmeaza parametrilor S si Q sint transformate in majuscule de catre componenta CCP daca este utilizata forma (2) de lansare a programului PIP. Forma (1) de lansare a programului PIP nu realizeaza aceasta conversie.

(1) PIP(CR)
(2) PIP linie-de-comanda(CR)

Tn - tine cont, in timpul transferului de la sursa la destinatie, de caracterele TAB (CTRL/I) existente. Fiecare TAB reprezinta salt in coloana "n";

U - transforma in timpul copierii minusculele in majuscule;

V - verifica daca datele au fost copiate corect prin recitirea inregistrarilor (in acest caz destinatia trebuie sa fie un fisier pe disc);

Z - anuleaza bitul de paritate la citirea datelor pentru fiecare caracter ASCII;

Gn - permite copierea unui/unor fisiere apartinind unui alt utilizator (n) in fisier/fisiere apartinind utilizatorului curent;

Nota: In general, comanda PIP realizeaza operatii numai asupra fisierelor apartinind utilizatorului curent;

W - se aplica atunci cind se doreste sa nu mai apara la consola mesajul:

DESTINATION FILE IS R/O, DELETE (Y/N)?

si sa se realizeze automat inlocuirea vechiului fisier.

In mod normal, daca un fisier destinatie dintr-o comanda PIP exista deja si este protejat la scriere (R/O), la consola apare mesajul:

DESTINATION FILE IS R/O, DELETE (Y/N)?

si se asteapta optiunea "Y" din partea utilizatorului, pentru a-l inlocui. Daca raspunsul este diferit de "Y", apare la consola mesajul:

** NOT DELETED **

si fisierul destinatie ramine nealterat (nu se efectueaza transferul de date);

R - permite ca si fisierele care au atributul SYS sa fie luate in consideratie in cadrul unei comenzi PIP (in mod normal, comanda PIP ignora aceste fisiere). Parametrul R semnifica automat si parametrul W. Utilizarea parametrului R face ca, in operatiile de copiere, atributele fisierelor de tip SYS (daca exista) sa se pastreze.

Exemple de comenzi PIP valide care utilizeaza parametri in cadrul transferului de date:

```
PIP X.ASM=B:[v](CR)
```

copiaza fisierul X.ASM de pe unitatea de disc B pe discul instalat si verifica daca datele au fost copiate corect.

```
PIP LPT:=X.ASM[nt8u](CR)
```

copiaza fisierul X.ASM la imprimanta, numeroteaza fiecare linie, tine cont de TAB-uri (din 8 in 8 coloane) transforma minusculele in majuscule.

```
PIP PUN:=X.HEX[i],Y.ZOT[h](CR)
```

copiaza fisierul X.HEX pe un periferic de tip **PUN:** ignorind inregistrarile de tip ":00" de sfirsit din cadrul fisierului **X.HEX.** Continua apoi transferul de date citind fisierul **Y.ZOT,** care contine inregistrari hexa, incluzind si inregistrarile de tip ":00" pe care le contine fisierul **Y.ZOT.**

```
PIP X.LIB=Y.ASM[sSUBR1:^Z^QJMP^IL3^Z](CR)
```

copiaza fisierul Y.ASM in fisierul **X.LIB.** Copierea incepe cind este detectat sirul "SUBR1:" si se opreste dupa intilnirea sirului "JMP(TAB)L3".

```
PIP PRN:=X.ASM[p50](CR)
```

copiaza fisierul X.ASM la un periferic de tip **LST:,** numerotind fiecare linie, tinind cont de caracterele TAB (din 8 in 8 coloane) si facind salt la pagina noua dupa fiecare 50 de linii.

Parametrul [nt8p60] este parametrul implicit pentru dispozitivul PRN:, iar parametrul [p50] precizat in comanda modifica parametrul implicit [p60].

```
USER 4(CR)
PIP A:=A:*.*[g2](CR)
```

copiaza toate fisierele de pe discul **A:,** apartinind utilizatorului 2, tot pe discul A:, dar in zona corespunzatoare utilizatorului curent (in cazul nostru, utilizatorul 4). Pentru a utiliza aceasta comanda este necesar ca programul PIP sa fie inregistrat in zona utilizatorului 4. Acest lucru se realizeaza prin secventa:

```
USER 0
DDT PIP.COM
DDT VERS 2.2
NEXT  PC
1E00 0100
G0
USER 4
SAVE 30 PIP.COM.
```

```
STAT A:*.* $R/O(CR)
PIP A:=B:*.COM[w](CR)
```

copiaza toate fisierele cu extensie COM de pe discul **B:,** pe discul **A:,** suprascriind toate fisierele cu acelasi nume existente pe **A:,** indiferent daca erau R/O sau nu.

```
STAT B:ED.COM $SYS(CR)

B:ED.COM set to SYS
STAT B:ED.COM $R/O(CR)

B:ED.COM set to R/O
```

```
PIP ED.COM=B:[r](CR)
```

copiaza fisierul ED.COM, de tip SYS si R/O, de pe discul **B:** pe discul **A:,** pastrindu-i atributele.

6.5 Mesaje de eroare BDOS

Exista patru situatii de eroare pe care componenta BDOS le detecteaza in timpul lucrului cu fisiere. Cind una din aceste conditii a fost detectata, BDOS afiseaza mesajul:

Bdos Err ON x: error

in care "x" este numele unitatii de disc si "error" este unul din urmatoarele 4 mesaje de eroare:

Bad Sector;
Select;
File R/O;
R/O.

Mesajul "Bad Sector" apare ca urmare a detectarii, de catre interfata de disc a unei erori de I/E. Aceasta eroare poate fi datorata unei incorecte functionari a interfetei, unui volum disc extrem de uzat sau starii de neoperationalitate a unei unitati (nu exista disc in unitatea respectiva, discul existent in unitate nu este formatat corespunzator, usa unitatii de disc este deschisa, sau nu este pusa sub tensiune unitatea de discuri). Daca se constata ca sistemul CP/M afiseaza frecvent acest mesaj de eroare, trebuie verificata starea interfetei de disc flexibil si conditiile de mediu in care microcalculatorul lucreaza. Mesajul poate apare si datorita unor incompatibilitati intre unitatile de discuri. Indiferent de cauza care a generat mesajul de eroare, utilizatorul poate continua lucrul tastind:

CTRL/C - pentru a reincarca sistemul CP/M;
(CR) - pentru a ignora sectorul eronat.

NOTA: Tastarea caracterului (CR) poate duce la distrugerea structurii logice a discului (de exemplu: daca apare intr-o operatie de scriere in fisierul "director").

Mesajul "Select" apare cind se incearca adresarea unei unitati neincluse in sistem la generare. In acest caz, valoarea lui "x" din mesajul de eroare da numele unitatii selectate eronat.

Mesajul "File R/O" apare atunci cind s-a incercat o stergere, o redenumire sau o modificare a atributelor unui fisier R/O (read/only). In acest caz, trebuie intii modificat atributul R/O al fisierului, in R/W (read/write) prin comanda STAT si apoi reluata comanda care a produs mesajul de eroare. Sistemul se reincarca automat dupa primul caracter introdus de la consola.

Mesajul "Read Only" apare atunci cind se incearca o scriere pe o unitate de disc care a fost protejata la scriere prin comanda STAT sau care a fost desemnata ca R/O de catre BDOS (ori de cite ori se monteaza intr-o unitate un alt volum de disc fara a se reincarca sistemul CP/M sau a se initializa sistemul CP/M acel volum disc va fi desemnat de BDOS ca R/O). Dupa aparitia acestui mesaj de eroare, sistemul CP/M asteapta introducerea unui caracter de la consola. Indiferent de caracterul introdus, imediat dupa receptionarea lui, sistemul CP/M va realiza automat o reincarcare a sa.

6.6 Structura octetului "IOBYTE" interpretata standard de catre sistemul CP/M

LIST (LST)	PUNCH (PUN)	READER (RDR)	CONSOLE (CON)
B7 B6	B5 B4	B3 B2	B1 B0
00 = TTY:	00 = TTY:	00 = TTY:	00 = TTY:
01 = CRT:	01 = PTP:	01 = PTR:	01 = CRT:
10 = LPT:	10 = UP1:	10 = UR1:	10 = BAT:
11 = UL1:	11 = UP2:	11 = UR2:	11 = UC1:

unde B7, B6, ...B0 sint bitii de date ai octetului IOBYTE.

6.7 Manual de interfata CP/M

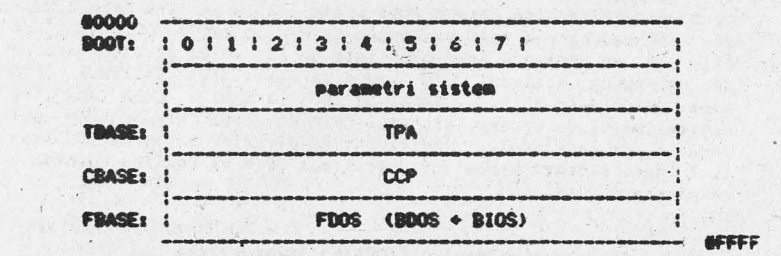
Acest manual descrie organizarea sistemului CP/M (inclusiv organizarea memoriei) si punctele de intrare in sistem. Se vor prezenta informatiile necesare pentru scrierea de programe executabile sub CP/M, programe ce utilizeaza facilitatile de I/E si de lucru cu discul oferite de sistem.

6.7.1 Organizarea CP/M

Sistemul CP/M este alcatuit din punct de vedere logic din urmatoarele patru parti:

- **BIOS** - sistemul de I/E de baza, care ofera interfata cu perifericele;
- **BDOS** - sistemul de exploatare a discurilor, care ofera primitivele de acces la disc;
- **CCP** - procesorul de comenzi-consola;
- **TPA** - zona pentru programe tranzitorii.

Componentele BIOS si BDOS sint grupate intr-un singur program numit FDOS, care are un punct de intrare unic. Componenta CCP este un program distinct, care utilizeaza programul FDOS pentru a oferi o interfata flexibila intre utilizator si informatiile existente pe disc. TPA este o zona de memorie (de exemplu zona de memorie care nu este utilizata de FDOS si CCP) in care se executa comenzile tranzitorii CP/M si programele-utilizator de aplicatii. Organizarea memoriei intr-un sistem standard CP/M este:



De obicei adresa BOOT este egala cu 0000, adresa TBASE este egala cu BOOT + 0100 = 0100, iar adresele CBASE si FBASE depind de tipul sistemului CP/M.

Zona adreselor 0000-0007 este rezervata pentru:

#0000-#0002 - salt la rutina de reinitializare a sistemului CP/M, existenta in BIOS (JMP WBOOT);
#0003 - octetul IOBYTE;
#0004 - numarul utilizatorului curent si al discului instalat;
#0005-#0007 - salt la punctul de intrare in FDOS, respectiv in BDOS (JMP FBASE).

Observatii: a. adresa #0005 este PUNCT DE INTRARE din programe tranzitorii in rutinele sistemului CP/M (in BDOS);
b. adresa prezenta in locatiile #0006-#0007 poate fi folosita pentru a determina dimensiunea maxima a memoriei disponibile (presupunind ca se reacopera componenta CCP);
c. adresa #0003 este rezervata pentru octetul IOBYTE (configuratia de I/E curenta);
d. adresa #0004 este rezervata pentru a stoca numarul utilizatorului curent si numarul discului instalat; octetul de la aceasta adresa are forma:

numar utilizator curent	numar disc instalat
0000-1111 (0-15)	0000 - 1111 (A) (P)

6.7.2 Executia programelor tranzitorii

Programele tranzitorii sint comenzi tranzitorii CP/M si programe-utilizator de aplicatii.

Orice program tranzitoriu se incarca de pe disc in zona TPA si se executa dupa cum va fi prezentat in continuare. Utilizatorul comunica cu componenta CCP (deci cu sistemul CP/M) prin introducerea, dupa fiecare prompter CP/M (>) a unei linii de comanda. Fiecare linie de comanda are una din urmatoarele forme:

(1) **comanda(CR)**
(2) **comanda specificator-fisier1(CR)**
(3) **comanda specificator-fisier1 specificator-fisier2(CR)**

unde "comanda" este numele unei comenzi CP/M rezidente (de exemplu: ERA, DIR, TYPE, etc.) sau numele unei comenzi CP/M tranzitorii sau numele unui program-utilizator. Daca "comanda" este numele unei comenzi CP/M, atunci aceasta comanda este executata imediat. In caz contrar, CCP cauta pe discul specificat (indicat inainte de comanda) sau pe discul instalat, un fisier cu numele:

comanda.COM

Daca un astfel de fisier este gasit, atunci se presupune ca el reprezinta imaginea-memorie a unui program care se executa in zona TPA si care, implicit, se incarca in memorie incepind de la adresa TBASE. Componenta CCP incarca fisierul tip ".COM" de pe disc in memorie, incepind de la adresa TBASE si ii preda controlul printr-o instructiune de tip "CALL". La sfirsitul executiei programului, controlul poate reveni in CCP (printr-o instructiune de tip "RET") sau in CP/M (printr-o instructiune "JMP BOOT"). Daca se doreste ca la sfirsitul executiei programului controlul

sa revina in CCP, atunci programul trebuie sa nu suprascrie zona CBASE-FBASE. In caz contrar, programul poate sa foloseasca memoria pina la adresa FBASE-1.

Daca in linia de comanda exista unul sau doi specificatori de fisier, atunci componenta CCP pregateste si unul sau doua "blocuri de control fisier" (FCB), in zona de memorie rezervata pentru "parametri sistem". Aceste FCB-uri sint construite in formatul impus de FDOS pentru accesul la fisiere (vezi sectiunea 7.4).

Programele tranzitorii pot folosi:

- facilitatile CP/M de I/E pentru a comunica cu consola si cu dispozitivele periferice;
- subsistemul de lucru cu discul, pentru accesul la fisiere rezidente pe acest suport.

Accesul din programe tranzitorii la sistemul de I/E al CP/M se face prin transmiterea catre sistemul CP/M, prin punctul de intrare in FDOS (existent la adresa BOOT + #0005), a unui numar de rutina si a unei adrese pentru informatii specifice rutinei. Dupa executia rutinei, FDOS intoarce o valoare ce indica modul de desfasurare a operatiei (operatie desfasurata corect sau codul de eroare [numeric], daca aceasta a esuat).

6.7.3 Conventii pentru apelul functiilor de sistem CP/M

Sistemul CP/M pune la dispozitia utilizatorilor o serie de rutine care pot fi apelate in cadrul programelor tranzitorii. Rutinele se impart in doua categorii:

- rutine de I/E pentru periferice simple;
- rutine de I/E pentru lucrul cu fisiere pe disc.

Rutinele de I/E pentru periferice simple sint:

- citire caracter de la consola;
- scriere caracter la consola;
- citire/scriere directa la consola;
- citire caracter de la dispozitivul tip "READER";
- scriere caracter la dispozitivul tip "PUNCH";
- scriere caracter la dispozitivul tip "LIST";
- citire/modificare octet IOBYTE;
- tiparire la consola a unui sir de caractere;
- citire buffer consola;
- citire stare consola.

Rutinele de I/E pentru lucrul cu fisiere pe disc sint:

- creare fisier;
- deschidere fisier;
- inchidere fisier;
- cautare in "director";
- modificare nume fisier;
- stergere fisier;
- citire secventiala sau directa a unui fisier;
- scriere secventiala sau directa a unui fisier;
- modificare atribute fisier;
- initializare "adresa DMA";
- initializare stare sistem discuri.

In sectiunea 6.7.5 este prezentata lista completa a rutinelor CP/M disponibile.

Accesul la rutinele FDOS se realizeaza prin transmiterea in:

- registrul "C" a numarului rutinei;
- perechea de registre "D&E" a unor parametri necesari rutinei.

Rutinele FDOS pot avea ca iesiri valori pe un octet (in registrul "A") sau pe doi octeti (in perechea de registre "H&L").

Observatii: a. Pentru rutinele care au ca iesiri valori pe doi octeti, aceste valori se gasesc si in registrele "A" si "B" (de exemplu (A) = L si (B) = H);

b. Conventiile de apel al rutinelor CP/M respecta standardele PL/M de comunicare parametri;

c. Exista rutine CP/M care folosesc doar registrul "E" pentru transmiterea unor parametrii necesari apelului lor;

d. Exista rutine CP/M care nu necesita parametrii (apelul lor presupune doar transmiterea, prin registrul "C", a numarului rutinei);

e. Exista rutine CP/M care nu au iesiri.

Rezulta ca secventa standard necesara pentru apelul unor rutine CP/M este:

```
BOOT    EQU     0
BDOS    EQU     BOOT+5
        .
        .
        .
        LD      C,numar-rutina
      [ LD      D,parametrii-specifici-rutinei ]
      [ LD      E,parametru-specific-rutinei ]
        CALL    BDOS        ; apel rutina prin
                            ; punctul de intrare in FDOS
```

Observatie: Liniile cuprinse intre [] reprezinta linii optionale, dependente de tipul rutinei.

S-a aratat in sectiunea 7.2 ca dupa incarcarea de pe disc in memorie a unui program tranzitoriu, componenta CCP ii preda acestuia controlul printr-o instructiune de tip "CALL". Executia programului tranzitoriu incepe avind SP-ul (registrul stivei, stack-pointerul) pozitionat pe o stiva cu o capacitate de 8 nivele (16 octeti), in care exista inscrisa doar adresa de revenire in CCP (7 nivele sint inca libere). Desi aceasta stiva nu este de obicei folosita de catre programele tranzitorii (majoritatea acestora rezervindu-si o stiva proprie si revenind in CCP printr-o instructiune de tip "JMP BOOT"), totusi este util de cunoscut faptul ca ea este suficient de mare pentru a realiza apeluri de rutine CP/M. Acest lucru este posibil intrucit componenta FDOS comuta SP-ul pe o stiva locala, la fiecare intrare intr-o rutina de sistem, neafectind astfel stiva initiala a programului. Programul in limbaj de asamblare de mai jos reprezinta un exemplu in acest sens, el realizind citirea unor caractere de la consola, pina la intilnirea unui caracter "*", care determina intoarcerea controlului in CCP:

```
BOOT    EQU     0
BDOS    EQU     BOOT+5      ;punct de intrare standard
                            ;in rutinele CP/M
```

```
CONIN   EQU     1       ;rutina "Console Input"
        ORG     100H    ;adresa de baza pentru TPA
NEXTC:
        LD      C,CONIN ;pregatire apel rutina CONIN
        CALL    BDOS    ;citire caracter de la consola
                        ;cu preluarea caracterului
                        ;in registrul "A"
        CPI     '*'     ;test pentru sfirsit
                        ;de prelucrare
        JNZ     NEXTC   ;reluare prelucrare
        RET             ;revenire in CCP
        END
```

6.7.4 Particularitati in utilizarea rutinelor CP/M de lucru cu fisiere pe disc

Pentru lucrul cu discul flexibil sistemul CP/M implementeaza, pe fiecare volum disc, o structura de fisiere identificate prin nume. Fiecare unitate de disc este, din punct de vedere logic, distincta, avind o zona rezervata pentru "director" si o alta zona pentru fisierele de date. Fiecare fisier are asociat un identificator alcatuit din:

- codul pentru selectarea unitatii de disc (o litera A...P);
- numele (alcatuit din 1...8 caractere ASCII diferite de spatiu);
- extensia (tipul) fisierului (alcatuita din 0...3 caractere ASCII diferite de spatiu).

Extensiile definesc categoria generica din care face parte un anumit fisier, in timp ce numele identifica in mod unic fisierul in cadrul categoriei respective. Astfel, sistemul CP/M utilizeaza urmatoarele extensii standard:

- **ASM**: pentru fisiere sursa in limbaj de asamblare tratabile cu asamblorul ASM sau MAC;
- **PRN**: pentru fisiere listing;
- **HEX**: pentru fisiere hexa;
- **BAS**: pentru fisiere sursa in limbaj BASIC;
- **INT**: pentru fisiere cod-obiect intermediar;
- **COM**: pentru fisiere cod-obiect direct executabil;
- **REL**: pentru fisiere cod-obiect relocabil;
- **COB**: pentru fisiere sursa in limbaj COBOL;
- **FOR**: pentru fisiere sursa in limbaj FORTRAN;
- **MAC**: pentru fisiere sursa in limbaj de asamblare tratabile cu asamblorul M80;
- **BAK**: pentru fisiere ce reprezinta versiuni anterioare intr-un proces de editare texte;
- **$$$**: pentru fisiere temporare.

Fisierele sursa sint tratate ca o secventa de caractere ASCII, in care fiecare "linie" din fisier se termina prin secventa de caractere (CR;LF, echivalent cu #0D;#0A). Astfel, o inregistrare CP/M (de 128 de octeti) poate contine mai multe linii de text sursa. Sfirsitul unui fisier ASCII este indicat prin caracterul CTRL/Z (#1A) sau prin "sfirsitul fizic" de fisier, detectat de catre rutina CP/M de citire. Caracterele CTRL/Z existente intr-un fisier cod-obiect (de exemplu, in fisiere tip COM) sint ignorate, sfirsitul de fisier fiind detectat de rutina CP/M de citire.

Orice fisier CP/M este o secventa de maximum 65536 inregistrari, de cite 128 octeti fiecare, numerotate de la 0 la 65535. Desi din punct de vedere logic inregistrarile intr-un fisier sint contigue, ele pot fi dispersate pe disc. Fisierelor li se aloca spatiu disc in mod dinamic, pe masura crearii lor.

Fiecare fisier este, intern, impartit in segmente de cite 16 KB, denumite "extensii logice". In cadrul fiecarei "extensii logice" exista 128 de inregistrari (128 * 128 B = 16 KB) numerotate de la 0 la 127 (#00 - #7F). Se observa ca in cadrul unei "extensii logice" contorul de inregistrari poate fi reprezentat pe 8 biti. Informatiile privind fiecare "extensie logica" a unui fisier ocupa spatiu in "directorul" discului respectiv. O "extensie logica" (16 KB) este formata din mai multe blocuri de alocare. Un bloc de alocare reprezinta spatiul disc minim ce poate fi alocat unui fisier. Un bloc de alocare are minimum 1 KB si maximum 16 KB; dimensiunea blocului de alocare este stabilita la generarea sistemului CP/M. Pentru utilizarea rutinelor CP/M de lucru cu fisiere pe disc trebuie respectate urmatoarele conventii:

- informatiile de identificare a oricarui fisier se transmit catre rutinele FDOS intr-un format standard, si anume sub forma unui "bloc de control fisier" (File Control Block = FCB); dimensiunea FCB depinde de tipul accesului la fisier (este de 33 de octeti pentru acces secvential si de 36 de octeti pentru acces direct); adresa FCB-ului se transmite in general prin registrele D&E;
- orice operatie de citire/scriere date intr-un fisier se realizeaza asupra unei inregistrari de 128 de octeti;
- adresa de inceput a zonei de memorie (de 128 de octeti) utilizata ca buffer in operatiile de citire/scriere se numeste "adresa DMA"; aceasta adresa nu se transmite ca parametru, ea fiind initializata de catre sistemul CP/M sau de catre o rutina CP/M speciala (rutina 26);
- se numeste "disc selectat" acea unitate de disc care a fost activata prin:
  - actiunea componentei CCP (discul instalat prin CCP este in momentul lansarii unui program in TPA "disc selectat");
  - prin rutina CP/M de selectare disc (rutina 14).
- se numeste "disc activ" acea unitate de disc, care de la ultima initializare/reinitializare a sistemului CP/M, sau de la ultima operatie de initializare stare sistem discuri (rutina 13), a facut obiectul unei selectii de tip:
  - explicit (prin CCP sau prin rutina CP/M de selectare disc (rutina 14));
  - implicit (printr-o rutina de deschidere sau creare a unui fisier cu octet 00 din FCB diferit de zero).

Programele tranzitorii pot folosi zona #005C-#007F (36 octeti) pentru unul din FCB-urile necesare. Aceasta zona are avantajul ca este initializata de CCP inainte de lansarea programului. Zona #0080-#00FF (128 octeti) poate fi folosita drept "buffer" in operatiile de intrare/iesire cu fisiere. CCP initializeaza "adresa DMA" cu #0080, deci daca un singur "buffer" este suficient se recomanda folosirea acestei zone.

Structura standard a unui FCB, dupa octetii care il compun, este urmatoarea:

0 - codul unitatii de disc pe care se gaseste fisierul respectiv;

#00 - pentru discul selectat;
#01 - pentru unitatea "A";
.
.
.
#10 - pentru unitatea "P".

1-8 - numele fisierului exprimat in ASCII (caractere majuscule, cu bitul B7=0); daca numele fisierului are mai putin de 8 caractere, atunci el trebuie completat la dreapta cu blancuri;

9-11 - extensia (tipul) fisierului exprimat in ASCII (caractere majuscule); daca extensia are mai putin de 3 caractere, atunci acest cimp trebuie completat la dreapta cu blancuri.

Daca fisierul este protejat la scriere (R/O) atunci B7 din octetul 9 este egal cu 1; altfel, acest bit este egal cu 0.

Daca fisierul este invizibil (SYS) atunci B7 din octetul 10 este 1; altfel, acest bit este egal cu 0.

12 - Numarul curent al "extensiei logice" a fisierului; de obicei acest octet este sters cu zero de utilizator;

13 - rezervat pentru sistem;

14 - rezervat pentru sistem; acest octet este sters cu zero atunci cind se executa operatii de tip OPEN, MAKE, SEARCH;

15 - contor de inregistrari in cadrul "extensiei logice" curente (ia valori intre #00 si #7F); acest cimp este completat de catre sistem;

16-31 - sint rezervati pentru sistem;

32 - numarul inregistrarii din "extensia logica" curenta; se foloseste in accesul secvential la fisiere; in mod normal acest octet este sters cu zero de catre utilizator inainte de deschiderea fisierului;

33-35 - reprezinta un parametru optional folosit numai in accesul direct la fisiere. El indica numarul inregistrarii de scris/citit (are valori intre #0000 si #FFFF cu posibilitate de depasire in octetul 35). Octetii 33 si 34 reprezinta o valoare pe 16 biti cu partea cea mai putin semnificativa in octetul 33 si cea mai semnificativa in octetul 34.

Fiecare fisier care trebuie exploatat prin CP/M trebuie sa aiba un FCB propriu, care sa furnizeze, pentru fiecare operatie cu fisierul, informatii privind numele si alocarea acestuia. Orice acces la un fisier implica initializarea de catre utilizator (programator) a FCB-ului corespunzator, respectiv prin inscrierea in octetii 00-11 ai specificatorului fisierului si prin umplerea cu #00 a restului de octeti (12-32/35). Informatiile din FCB-urile corespunzatoare fisierelor de pe un disc se gasesc inregistrate in "directorul" discului respectiv si sint aduse in memoria interna inainte ca utilizatorul sa inceapa lucrul asupra fisierului/fisierelor (vezi rutinele OPEN, MAKE). Copia din memorie a FCB-ului este actualizata pe masura ce au loc operatii asupra fisierului, iar la terminarea lucrului cu acesta ea este inregistrata pe disc (vezi rutina CLOSE).

Atunci cind o comanda (program) se lanseaza in executie prin:

comanda specificator-fisier1(CR)
comanda specificator-fisier1 specificator-fisier2(CR)

componenta CCP construieste (dupa cum s-a aratat in sectiunea 6.2) primii 16 octeti din doua FCB-uri, pornind de la specificatorul/specificatorii de fisier prezenti in linia de comanda (dupa numele comenzii). Automat, CCP completeaza (daca este cazul) numele si extensia fisierelor cu blancuri. Primul FCB este construit la adresa #005C si poate fi folosit ca atare pentru operatii ulterioare asupra fisierului "specificator-fisier1". Al doilea FCB este construit in octetii 16-31 din primul FCB (adica de la adresa #006C) si trebuie sa fie mutat intr-o alta zona de memorie inainte de utilizarea lui.

Daca, de exemplu, utilizatorul introduce comanda:

PROGNAME B:X.ZOT Y.ZAP(CR)

fisierul PROGNAME.COM de pe discul instalat va fi incarcat in zona TPA, iar blocul de control de la adresa #005C va fi initializat astfel:

octetul 00 = #02 (cod unitate "B");
octetii 01-08 = 'X ';
octetii 09-11 = 'ZOT';
octetii 12-15 = #00;
octetul 16 = #00 (cod disc selectat, care in acest caz este chiar discul instalat);
octetii 17-24 = 'Y ';
octetii 25-27 = 'ZAP';
octetii 28-31 = #00.

Programatorul trebuie sa salveze continutul celui de-al doilea FCB (cei 16 octeti incepind de la adresa #006C) inainte de a deschide fisierul corespunzator primului FCB (de la adresa #005C), intrucit prin deschiderea acestuia informatiile referitoare la cel de-al doilea fisier vor fi sterse (suprascrise) de catre sistem (de catre rutina OPEN).

Daca intr-o linie de comanda CP/M nu apare nici un specificator-fisier atunci zonele #005D-#0067 si #006D-#0077 vor contine blancuri.

Componenta CCP asigura automat transformarea minusculelor in majuscule.

Dupa receptionarea unei linii de comanda CCP pastreaza la adresa #0080 un buffer pentru consola, in care exista o copie a continutului liniei de comanda, exceptind numele comenzii. Astfel, pentru exemplul considerat anterior, bufferul de la adresa #0080 va avea urmatorul continut:

octetul 00 = #0E (numarul de caractere utile din linia de comanda exceptind numele comenzii);
octetul 01 = ' ';
octetii 02-08 = 'B:X.ZOT';
octetul 09 = ' ';
octetii 10-14 = 'Y.ZAP'.

Este sarcina utilizatorului de a extrage informatiile din acest buffer, inainte de a executa orice operatie asupra unui fisier, operatie prin care aceasta zona este suprascrisa (initial "adresa DMA" este egala cu #0080, adica tocmai adresa de inceput a acestui buffer consola).

6.7.5 Prezentarea rutinelor CP/M

```
-------------------------------------------------------------
|RUTINA 0  : Reinitializare sistem CP/M                     |
|            (System Reset)                                 |
|Intrari   : registrul C    : #00                           |
-------------------------------------------------------------
```

Efect: intoarce controlul din programul utilizator in CP/M; aceasta functie are acelasi efect ca "JMP BOOT".

```
-------------------------------------------------------------
|RUTINA 1  : Citire caracter de la consola                  |
|            (Console Input)                                |
|Intrari   : registrul C    : #01                           |
|Iesiri    : registrul A    : caracter ASCII                |
-------------------------------------------------------------
```

Efect: preia un caracter de la consola si-l transmite in registrul "A". Toate caracterele tiparibile si in plus (CR)(LF) si (BS)(CTRL/H) sint transmise in ecou la consola. De asemenea, caracterul CTRL/I (TAB) muta cursorul in urmatoarea pozitie de tabulare. Restul de caractere netiparibile nu sint transmise in ecou la consola. Rutina asteapta un timp nelimitat pina cind se tasteaza un caracter la consola.

```
-------------------------------------------------------------
|RUTINA 2  : Scriere caracter la consola                    |
|            (Console Output)                               |
|Intrari   : registrul C    : #02                           |
|            registrul E    : caracter ASCII                |
-------------------------------------------------------------
```

Efect: transmite la consola caracterul specificat prin registrul "E". Caracterele "TAB" (CRTL/I) sint expandate. Caracterul CTRL/S introdus de la consola este interpretat drept stop defilare. Reluarea defilarii dupa CTRL/S se face cu orice caracter diferit de CTRL/C. CTRL/C dupa CTRL/S reinitializeaza sistemul CP/M.

```
-------------------------------------------------------------
|RUTINA 3  : Citire caracter de la cititor logic            |
|            (Reader Input)                                 |
|Intrari   : registrul C    : #03                           |
|Iesiri    : registrul A    : caracter ASCII                |
-------------------------------------------------------------
```

Efect: preia un caracter de la dispozitivul **RDR:** curent si-l depune in registrul "A". Rutina asteapta un timp nelimitat preluarea caracterului de la **RDR:**.

```
-------------------------------------------------------------
|RUTINA 4  : Scriere caracter la perforator logic           |
|            (Punch Output)                                 |
|Intrari   : registrul C    : #04                           |
|            registrul E    : caracter ASCII                |
-------------------------------------------------------------
```

Efect: transmite la dispozitivul **PUN:** curent caracterul specificat prin registrul "E".

```
-------------------------------------------------------------
|RUTINA 5  : Scriere caracter la imprimanta logica          |
```

```
|            (List Output)                          |
|Intrari   : registrul C    : #05                   |
|            registrul E    : caracter ASCII        |
```

Efect: transmite la dispozitivul LST: curent caracterul specificat prin registrul "E".

```
|RUTINA 6  : Citire/Scriere directa la consola      |
|            (Direct Console I/O)                   |
|Intrari   : registrul C    : #06                   |
|                                                   |
|            registrul E    : #FF pentru intrare    |
|                             sau caracter ASCII    |
|                             pentru iesire         |
|Iesiri    : registrul A    : caracter ASCII sau #0 |
```

Efect: daca registrul "E" este egal cu #0FF, atunci rutina realizeaza citirea (fara ecou) a unui caracter de la consola. Registrul "A" va contine codul caracterului ASCII introdus sau #00, daca nu s-a introdus nici un caracter.

Rutina nu asteapta nelimitat introducerea unui caracter de la consola (ea intoarce imediat (A)=#00, daca in registrul de interfata al consolei nu exista nici un caracter disponibil). Este indicat ca utilizatorul sa astepte prin program introducerea unui caracter de la consola.

Daca registrul "E" contine codul unui caracter ASCII, atunci rutina realizeaza scrierea la consola a caracterului respectiv.

Rutina 6 nu trebuie sa fie folosita impreuna cu alte rutine CP/M care realizeaza intrari/iesiri cu consola (rutinele 1, 2, 9, 10 si 11).

```
|RUTINA 7  : Citire IOBYTE                          |
|            (Get I/O Byte)                         |
|Intrari   : registrul C    : #07                   |
|Iesiri    : registrul A    : valoare curenta IOBYTE |
```

Efect: citeste octetul de configuratie I/E si il plaseaza in registrul "A". Pentru semnificatia acestui octet vezi sectiunea.

```
|RUTINA 8  : Modificare IOBYTE                      |
|            (Set I/O Byte)                         |
|Intrari   : registrul C    : #08                   |
|            registrul E    : valoare pentru IOBYTE |
```

Efect: scrie continutul registrului "E" in octetul de configuratie I/E, modificind astfel asignarea dispozitivelor logice curente.

```
|RUTINA 9  : Tiparire sir de caractere la consola   |
|            (Print String)                         |
|Intrari   : registrul C    : #09                   |
|            registrele D&E : adresa sir            |
```

Efect: tipareste la consola sirul de caractere ASCII a carui adresa de inceput este specificata in registrele "D&E". Tiparirea se termina atunci cind s-a intilnit caracterul "$". Rutina trateaza caracterele TAB (CTRL/I) intilnite, mutind cursorul in urmatoarea pozitie de tabulare. La fel ca in rutina 2, se face verificare pentru caracterul CTRL/S (stop defilare).

RUTINA 10	Citire buffer consola (Read Console Buffer)	
Intrari	registrul C	#0A
	registrele D&E	adresa buffer

Efect: rutina permite citirea unei linii introduse de la consola cu transferarea continutului ei intr-o zona de memorie a carei adresa de inceput este data in registrele "D&E".

O linie editata la consola se considera terminata atunci cind s-a introdus caracterul (CR) sau caracterul (LF) sau atunci cind s-a depasit capacitatea bufferului consolei, specificata de utilizator in primul octet din buffer. Rutina aduce in bufferul a carui adresa este data in registrele "D&E" urmatorul continut:

- octetul 00; numarul maxim de caractere din bufferul consolei (cu valori intre 1 si 255); acest cimp este initializat de catre utilizator inaintea apelului rutinei 10;
- octetul 01; numarul real de caractere introduse in linie (fara (CR) si (LF)); valoare intoarsa de sistem;
- octetii 02-n; caracterele din linia de editare (c1,c2,c3,...,cn).

Daca numarul de caractere din linia de editare este mai mic decit numarul maxim de caractere din buffer, atunci dupa ultimul caracter citit din linia de editare (i.e. caracterul "cn") si pina la pozitia corespunzatoare ultimului caracter posibil in buffer, vor exista in buffer o serie de caractere fara semnificatie pentru utilizator. Ele reprezinta un rest neinitializat din bufferul consolei.

In timpul introducerii de la consola a liniei sint active, pentru corectii, urmatoarele caractere de editare ale sistemului CP/M:

RUBOUT/DEL - sterge din bufferul de intrare si reda in ecou ultimul caracter din buffer; stergere caracter pentru terminale cu hirtie;

CTRL/C - reincarcarea sistemului de operare;

CTRL/E - forteaza sfirsitul fizic al unei linii; cursorul se pozitioneaza pe inceputul liniei, dar linia nu se transmite decit atunci cind se tasteaza (CR);

CTRL/H - sterge ultimul caracter din bufferul de intrare si de pe ecran; stergere pentru terminale cu tub catodic;

CTRL/J - este echivalent unui caracter (LF) si reprezinta sfirsitul unei linii;

CTRL/M - este echivalent unui caracter (CR) si reprezinta sfirsitul unei linii;

CTRL/R - tipareste la consola, pe linia imediat urmatoare, continutul curent al bufferului de intrare; prin acest caracter se poate vizualiza

continutul curent al unei linii in care s-au efectuat corectii prin RUBOUT (DEL);

CTRL/U - sterge bufferul de intrare, afiseaza caracterul "#" pe linia curenta si trece la linie noua; anulare linie pentru terminale cu hirtie;

CTRL/X - sterge bufferul de intrare si caracterele corespunzatoare de pe ecran; anulare linie pentru terminale cu tub catodic;

CTRL/P - inverseaza starea indicatorului "hard copy". Daca acest indicator este activ, toate caracterele care se afiseaza la consola prin rutinele 2 si 9 sint trimise si la imprimanta logica. Un nou CTRL/P, introdus in timpul executiei rutinei 10, readuce lucrurile la normal.

RUTINA 11	: Citire stare consola (Get Console Status)	
Intrari	: registrul C	: #0B
Iesiri	: registrul A	: stare consola

Efect: rutina verifica daca s-a introdus un caracter de la consola. Daca in registrul de interfata al consolei exista un caracter disponibil, atunci rutina intoarce in registrul "A" valoarea #FF. In caz contrar, in registrul "A" se va afla valoarea #00.

RUTINA 12	: Citire versiune sistem (Get Version Number)	
Intrari	: registrul C	: #0C
Iesiri	: registrele H&L	: numar de versiune

Efect: rutina intoarce in registrele "H&L" o valoare egala cu numarul de versiune al sistemului CP/M sub care se lucreaza, respectiv (H)=#00 iar (L)=numarul de versiune (ex: #22 pentru versiunea 2.2). Intrucit in versiunea 1.4 de CP/M nu erau implementate functiile de acces aleator la fisiere, rutina 12 trebuie apelata in toate programele care fac acces aleator. Daca numarul de versiune intors este mai mic decit #20, programul trebuie terminat dupa ce se tiparesc anumite replici, mai putin principiale.

RUTINA 13	: Initializare stare sistem discuri (Reset Disk System)	
Intrari	: registrul C	: #0D

Efect: rutina dezactiveaza logic toate unitatile de disc (le sterge atributul R/O), asigneaza ca disc selectat unitatea "A" si stabileste ca "adresa DMA" adresa #0080. Rutina poate fi folosita atunci cind o aplicatie necesita schimbari de volume disc fara a se reinitializa sistemul CP/M (prin CTRL/C).

RUTINA 14	: Selectare disk (Select Disk)	
Intrari	: registrul C	: #0E

	registrul E	: numar unitate selectata

Efect: rutina desemneaza unitatea specificata in registrul "E" ca "disc selectat". Numarul unitatii de disc se specifica prin valorile: #00 pentru unitatea "A", #01 pentru unitatea "B",...#0F pentru unitatea "P". In urma executiei rutinei, unitatea specificata in registrul "E" este trecuta in starea "activ" (disc activ) care incarca "directorul" volumului respectiv; unitatea ramine in aceasta stare pina la o noua initializare sau reinitializare a sistemului CP/M sau pina la o noua operatie de "initializare stare sistem discuri" (rutina 13). Daca in timp ce o unitate este "activa" se fac schimbari de volume disc, atunci automat unitatea este desemnata de catre sistem ca R/O (vezi si rutina 28).
NOTA: Toate FCB-urile care au primul octet egal cu #00 se refera implicit la fisiere care se gasesc pe discul selectat.

RUTINA 15	: Deschidere fisier (Open File)	
Intrari	: registrul C	: #0F
	registrele D&E	: adresa FCB
Iesiri	: registrul A	: cod "director"

Efect: rutina realizeaza activarea unui fisier care se gaseste in "directorul" discului specificat prin octetul 00 din FCB si care apartine utilizatorului curent. Adresa FCB-ului fisierului de deschis este data prin registrele "D&E".

Programul FDOS cauta in directorul discului specificat o intrare identica cu valoarea octetilor 1-12 din FCB.
NOTA: In FCB octetii 12 si 32 trebuie stersi cu zero de catre utilizator, inaintea apelului rutinei 15. Daca programul FDOS gaseste o astfel de intrare, atunci informatiile din "director" corespunzatoare ei sint copiate in octetii 1 - 31 din FCB, permitindu-se astfel accesul la fisier pentru operatii ulterioare de citire/scriere.

Rutina intoarce in registrul "A" o valoare 0-3, daca operatia de deschidere s-a efectuat corect si o valoare egala cu 255 (#FF), daca aceasta a esuat.

Programatorul nu trebuie sa efectueze operatii asupra unui fisier decit dupa ce s-a realizat corect deschiderea sa.

Exista posibilitatea ca in cadrul FCB-ului, in octetii 1-11 sa apara un specificator-multiplu de fisier, adica sa apara caractere "?" (care inlocuiesc orice caracter in pozitia respectiva). In acest caz, programul FDOS cauta in "director" prima intrare care corespunde specificatorului-multiplu de fisier din FCB. Daca functia 13 se termina cu succes, programul FDOS va inlocui specificatorul-multiplu din FCB cu specificatorul-individual corespunzator fisierului gasit.

RUTINA 16	: Inchidere fisier (Close File)	
Intrari	: registrul C	: #10
	registrele D&E	: adresa FCB
Iesiri	: registrul A	: cod "director"

Efect: rutina realizeaza reversul rutinelor 15 (OPEN) si 22 (MAKE). Astfel, presupunind ca FCB-ul a carui adresa este spe-

cificata in registrele "D&E" a fost activat anterior printr-o rutina de "deschidere fisier" (rutina 15) sau de "creare fisier" (rutina 22), rutina de "inchidere fisier" inregistreaza FCB-ul curent in "directorul" discului specificat, actualizind astfel intrarea din "director" corespunzatoare fisierului respectiv.

Rutina intoarce in registrul "A" o valoare egala cu 0-3 daca operatia de inchidere s-a desfasurat corect, sau o valoare egala cu 255 (#FF) daca numele fisierului din FCB nu a fost gasit in "director".

Inchiderea fisierelor care au fost exploatate doar in citire este optionala. Numai fisierele in care s-au efectuat operatii de scriere trebuie inchise (pentru a actualiza in "director" informatiile referitoare la acele fisiere).

Daca in FCB-ul fisierului de inchis apare un specificator multiplu (i.e. caractere "?"), atunci rutina va efectua cautarea in "director" asa cum face rutina 15.

RUTINA 17	Cauta in "director" prima intrare (Search for First)	
Intrari	registrul C	#11
	registrele D&E	adresa FCB
Iesiri	registrul A	cod "director"

Efect: rutina cauta in "director" prima intrare care corespunde valorilor octetilor 0-12 din FCB-ul a carui adresa este data in registrele "D&E". Rutina intoarce in registrul "A" valoarea 255 (#FF) daca nu a gasit o astfel de intrare, sau o valoare cuprinsa intre 0-3 daca a gasit-o. Daca in "director" a fost gasita o intrare identica cu specificatorul-fisierului din FCB, atunci zona de memorie a carei adresa este "adresa DMA" va fi completata cu o inregistrare de "director" (128 octeti) si anume cu acea inregistrare din "director" care contine intrarea respectiva. Adresa relativa a intrarii, in cadrul inregistrarii de "director", este egala cu (A) * 32 (i.e. continutul registrului "A" rotit spre stinga cu 5 biti sau "ADD A" de 5 ori). Pe baza acestei adrese relative, programele de aplicatii pot extrage din bufferul care contine inregistrarea de "director", informatiile necesare din intrarea gasita.

Daca FCB-ul contine un specificator-multiplu (i.e. apar caractere "?" in pozitiile 1-12), atunci rutina intoarce PRIMA intrare din "director" care satisface specificatorul. Daca octetul 00 din FCB contine caracterul "?", atunci rutina intoarce automat prima intrare din "directorul" discului selectat (indiferent de numarul utilizatorului caruia ii apartine intrarea respectiva, indiferent de continutul acestei intrari si indiferent daca intrarea este stearsa sau nu).

RUTINA 18	Cauta urmatoarea intrare (Search for Next)	
Intrari	registrul C	#12
Iesiri	registrul A	cod "director"

Efect: aceasta rutina este similara rutinei 17, cu exceptia faptului ca "directorul" discului specificat nu se investigheaza de la inceputul sau (ca in toate celelalte rutine), ci se cauta intrarea corespunzatoare FCB-ului incepind de la ultima intrare din "director" gasita.

identica cu FCB-ul specificat.

RUTINA 19	Stergere fisier (Delete File)	
Intrari	registrul C	#13
	registrele D&E	adresa FCB
Iesiri	registrul A	cod "director"

Efect: rutina realizeaza stergerea unuia sau mai multor fisiere, specificate prin FCB-ul a carui adresa este data in registrele "D&E". FCB-ul poate contine un specificator-individual de fisier sau un specificator-multiplu de fisier (pot apare caractere "?" in zona de nume sau de extensie a fisierului, dar nu si in zona pentru numele unitatii de disc pe care se gaseste fisierul - asa cum se putea in rutinele 17 si 18).

Rutina intoarce in registrul "A" valoarea 255 (#FF), daca fisierul/fisierele specificate in FCB nu au fost gasite, si o valoare 0-3, daca operatia de stergere s-a efectuat normal.

RUTINA 20	Citire secventiala (Read Sequential)	
Intrari	registrul C	#14
	registrele D&E	adresa FCB
Iesiri	registrul A	octet de stare

Efect: presupunind ca FCB-ul a carui adresa este specificata in registrele "D&E" a fost activat printr-o rutina de "deschidere fisier" (rutina 15) sau de "creare fisier" (rutina 22), rutina "citire secventiala" realizeaza citirea din fisier a urmatoarei inregistrari de 128 de octeti si transferarea ei in memorie, intr-o zona a carei adresa este "adresa DMA". Numarul inregistrarii din cadrul "extensiei logice" curente este specificat prin octetul 32 din FCB. Dupa citire, valoarea acestui octet va fi automat incrementata cu 1. Daca valoarea rezultata in octetul 32 depaseste 127 (#7F) atunci urmatoarea "extensie logica" a fisie-

rului este deschisa automat si octetul 32 ia valoarea #00, fiind astfel pregatit pentru urmatoarea operatie de citire. Daca operatia de citire s-a efectuat normal, atunci registrul "A" va avea valoarea #00; in caz contrar, adica atunci cind nu mai exista date in fisier (s-a atins sfirsitul fisierului!), registrul "A" va avea o valoare diferita de #00.

RUTINA 21	Scriere secventiala (Write Sequential)	
Intrari	registrul C	#15
	registrele D&E	adresa FCB
Iesiri	registrul A	octet de stare

Efect: presupunind ca FCB-ul a carui adresa este specificata in registrele "D&E" a fost activat printr-o operatie de "deschidere fisier" (rutina 15) sau "creare fisier" (rutina 22) anterioara, rutina "scriere secventiala" realizeaza scrierea in fisier a unei inregistrari de 128 de octeti. Inregistrarea de scris este luata

din memorie, de la o adresa egala cu "adresa DMA" si este plasata in fisier in pozitia data de valoarea octetului 32 din FCB (numarul inregistrarii in cadrul "extensiei logice" curente). Dupa scrierea inregistrarii in fisier, continutul octetului 32 din FCB este automat incrementat cu 1. Daca in urma incrementarii rezulta o depasire (o valoare mai mare ca 127 (i.e. #7F)), atunci este deschisa urmatoarea "extensie logica" a fisierului si octetul 32 din FCB este initializat cu #00, in vederea unor operatii de scriere ulterioare. Operatia de "scriere secventiala" poate avea loc si in cadrul unor fisiere deja create corect, caz in care inregistrarile ce se scriu se vor suprapune peste cele existente, practic inlocuindu-le pe cele vechi.

Rutina intoarce in registrul "A" valoarea #00 daca operatia de scriere a decurs normal, sau o valoare diferita de #00 daca operatia de scriere a esuat datorita lipsei de spatiu pe disc.

RUTINA 22	: Creare fisier		
	(Make File)		
Intrari	: registrul C	: #16	
	registrele D&E	: adresa FCB	
Iesiri	: registrul A	: cod "director"	

Efect: rutina are acelasi efect ca si rutina "deschidere fisier" (rutina 15), cu exceptia faptului ca, in acest caz, FCB-ul trebuie sa contina numele unui fisier care nu exista in "directorul" discului specificat.

Programul FDOS creaza intrarea din "director" corespunzatoare FCB-ului si initializeaza atit FCB-ul cit si "directorul" discului, fortind lungimea fisierului pe 0.

NOTA: Programatorul trebuie sa evite duplicarea numelor fisierelor in "director", respectiv trebuie sa se asigure ca in "director" nu exista un alt fisier cu nume identic cu cel al fisierului de creat. In acest scop, este indicat ca el sa efectueze anterior rutinei 22 o operatie de "stergere fisier" (rutina 19).

Rutina 22 intoarce in registrul "A" o valoare 0-3 daca operatia s-a desfasurat normal sau o valoare 255 (#FF) daca nu mai exista spatiu in "directorul" discului. Rutina 22 are ca efect secundar si activarea FCB-ului, astfel incit nu mai este necesara o operatie ulterioara de "deschidere fisier".

RUTINA 23	: Schimbare nume fisier		
	(Rename File)		
Intrari	: registrul C	: #17	
	registrele D&E	: adresa FCB	
Iesiri	: registrul A	: cod "director"	

Efect: rutina realizeaza schimbarea numelui unui fisier. Rutina utilizeaza FCB-ul adresat prin registrele "D&E" astfel:

- primii 16 octeti din FCB reprezinta numele vechi al fisierului;
- ultimii 16 octeti din FCB reprezinta numele nou al fisierului;
- octetul 00 din FCB reprezinta codul unitatii pe care se gaseste fisierul de redenumit (octetul 16 din FCB este considerat #00).

Rutina intoarce in registrul "A" o valoare 0-3 daca operatia s-a desfasurat normal, sau valoarea 255 (#FF) daca nu s-a gasit in "directorul" discului specificat un fisier cu nume identic cu cel al fisierului de redenumit.

RUTINA 24	Citire vector unitati disc active (Return Log-in Vector)	
Intrari	registrul C	#18
Iesiri	registrele H&L	vectorul unitatilor de disc active

Efect: rutina analizeaza care din unitatile de disc A-P, este "activa", respectiv care din aceste unitati a fost activata:

- explicit printr-o rutina de "selectare disc" (rutina 14);
- implicit printr-o operatie de deschidere/creare fisier (cu valoare diferita de #00 in octetul 00 din FCB).

Pentru unitatile de disc active, rutina intoarce o valoare logica "1", iar pentru cele care nu sint active o valoare logica "0". Bitul B0 din registrul "L" reprezinta starea unitatii "A", iar bitul B7 din registrul "H" reprezinta starea unitatii "P". Astfel, prin registrele "H&L" (respectiv B&A) rutina intoarce un vector ce indica starea tuturor unitatilor A-P.

RUTINA 25	Citire numar disc selectat (Return Current Disk)	
Intrari	registrul C	#19
Iesiri	registrul A	numarul discului selectat

Efect: rutina intoarce in registrul "A" numarul "discului selectat". Acest numar este cuprins intre #00 pentru unitatea "A" si #0F pentru unitatea "P".

RUTINA 26	Modificare "adresa DMA" (Set DMA Adress)	
Intrari	registrul C	#1A
	registrele D&E	adresa DMA

Efect: rutina permite modificarea "adresei DMA", adica a adresei bufferului de 128 octeti folositi in operatiile de citire/scriere fisiere. In general, "adresa DMA" stabilita la initializarea sau reinitializarea CP/M precum si dupa o operatie de "initializare stare sistem discuri" (rutina 13), este adresa #0080. Rutina permite comutarea acestei adrese pe orice alta adresa (data in registrele "D&E"), permitind astfel localizarea bufferului de 128 de octeti in orice zona de memorie.

Rutina stabileste "adresa DMA" ca fiind egala cu adresa specificata in registrele "D&E". Noua valoare pentru "adresa DMA" este valabila pina la:

- o initializare sau reinitializare a sistemului CP/M;
- un alt apel al rutinei 26;
- o operatie de "initializare stare sistem discuri" (rutina 13).

```
-------------------------------------------------------
|RUTINA 27 : Citire adresa vector de alocare          |
|          (Get Alloc Addr)                           |
|Intrari   : registrul C    : #1B                     |
|Iesiri    : registrul H&L  : adresa vector alocare   |
-------------------------------------------------------
```

Efect: rutina intoarce in registrele "H&L" adresa vectorului de alocare asociat discului selectat. Sistemul CP/M pastreaza in memorie, pentru fiecare unitate "activa", un vector de alocare. Acest vector poate fi folosit pentru a determina dimensiunea spatiului-disc ramas liber pe un volum (vezi comanda tranzitorie STAT).

Informatiile cuprinse in vectorul de alocare asociat unei unitati de disc care a fost desemnata ca R/O de catre CP/M (in urma schimbarii unui volum disc fara initializarea sistemului CP/M sau fara o operatie de "initializare stare sistem discuri" (rutina 13)) sint false.

```
-------------------------------------------------------
|RUTINA 28 : Setare atribut R/O unitate disc          |
|          (Write Protect Disk)                       |
|Intrari :   registrul C    : #1C                     |
-------------------------------------------------------
```

Efect: rutina desemneaza temporar discul selectat ca disc R/O. Orice incercare de scriere pe acel disc, pina la o initializare sau reinitializare a sistemului CP/M sau pina la o operatie de "initializare stare sistem discuri" (rutina 13), va produce mesajul:

BDOS ERR on d:R/O

```
-------------------------------------------------------
|RUTINA 29 : Citire vector de unitati R/O             |
|          (Get Read/Only Vector)                     |
|Intrari   : registrul C    : #1D                     |
|Iesiri    : registrele H&L : vectorul de unitati R/O |
-------------------------------------------------------
```

Efect: rutina intoarce in registrele "H&L" un vector ce indica unitatile de disc care sint desemnate ca R/O in acel moment. Bitul B0 din registrul "L" corespunde unitatii "A", iar bitul B7 din registrul "H" corespunde unitatii "P". O valoare logica "1" indica faptul ca unitatea respectiva este R/O.

O unitate de disc devine R/O dupa un apel al rutinei 28 sau in urma schimbarii volumului disc din acea unitate (sistemul CP/M, in acest caz, desemneaza automat unitatea respectiva ca R/O).

```
-------------------------------------------------------
|RUTINA 30 : Modificare atribute fisier               |
|          (Set File Attributes)                      |
|Intrari   : registrul C    : #1E                     |
|          : registrele D&E : adresa FCB              |
|Iesiri    : registrul A    : cod "director"          |
-------------------------------------------------------
```

Efect: rutina permite modificarea atributelor R/O si SYS ale unui fisier specificat in FCB-ul a carui adresa este data in

registrele "D&E". FCB-ul trebuie sa contina un specificator-individual de fisier. Noile atribute ale fisierului se specifica prin:

- bitul B7 din octetul 09 din FCB ("1" reprezinta fisier protejat la scriere (R/O));
- bitul B7 din octetul 10 din FCB ("1" reprezinta fisier invizibil (SYS)).

Rutina cauta in "director" o intrare care corespunde octetilor 1-11 din FCB: comparatia se face ignorind valorile bitilor B7 din octetii 1-11 din FCB si din "director". Daca o astfel de intrare este gasita rutina modifica intrarea din "director" corespunzatoare. Rutina intoarce in registrul "A" o valoare 0-3 pentru cazul in care operatia s-a desfasurat corect, sau o valoare egala cu 255 (#FF) pentru cazul in care nu fost gasita o astfel de intrare.

RUTINA 31	Citire adresa "bloc parametri disc" (Get Disk Parms Addr)	
Intrari	registrul C	#1F
Iesiri	registrele H&L	adresa bloc parametri disc

Efect: rutina intoarce in registrele "H&L" adresa "blocului de parametri ai discului", bloc care este rezident in BIOS. Aceasta adresa poate fi folosita:

- pentru a extrage din zona respectiva informatii privind parametrii discului (informatii necesare pentru a fi afisate sau pentru a se realiza, pe baza lor, calcule);
- pentru a modifica, prin program, parametrii discului; de obicei programele de aplicatii nu folosesc rutina 31 in acest scop.

RUTINA 32	Citire/modificare numar utilizator (Set/Get User Code)	
Intrari	registrul C	#20
	registrul E	#FF pentru citire numar utilizator pentru modificare
Iesiri	registrul A	numar utilizator daca E a fost #FF

Efect: rutina permite citirea numarului utilizatorului curent (daca (E)=#FF) si intoarcerea acestui numar in registrul "A" sau modificarea numarului utilizatorului curent, in functie de valoarea curenta a registrului "E" (modulo 16). Numarul utilizatorului curent variaza intre #00 si #0F.

RUTINA 33	Citire directa (Read Random)	
Intrari	registrul C	#21
	registrele D&E	adresa FCB
Iesiri	registrul A	octet de stare

Efect: rutina este similara rutinei "citire secventiala" (rutina 20) cu exceptia faptului ca nu se citeste din fisier inregistrarea de 128 de octeti cu numarul specificat in octetul 32 din FCB, ci inregistrarea al carei numar este dat in octetii 33 si 34 din FCB. Octetii 33 si 34 din FCB reprezinta o valoare pe 16 biti cuprinsa intre #0000-#FFFF cu partea cea mai putin semnificativa in octetul 33 si cea mai semnificativa in octetul 34. Octetul 35 trebuie sa fie #00 intrucit o valoare diferita de #00 indica o "depasire" dincolo de sfirsitul fisierului.

Citirea directa necesita in prealabil deschiderea "extensiei logice" cu numarul 0 a fisierului (prima "extensie logica" a fisierului), operatie care se realizeaza prin rutina 15.

Daca operatia de citire directa s-a efectuat corect, atunci:

- registrul "A" va avea valoarea #00;
- inregistrarea citita din fisier se va gasi depusa in memorie la "adresa DMA";
- valorile octetilor 12 (numarul "extensiei logice" curente) si 32 (numarul inregistrarii in cadrull "extensiei logice curente) vor fi automat modificate in functie de numarul inregistrarii citite (octetii 33 si 34);
- valoarea octetului 32 nu va fi incrementata cu 1 (ca in rutina 20).

Dupa o operatie de "citire directa" pot fi folosite operatii de "citire secventiala/scriere secventiala". Programatorul insa trebuie sa tina cont de faptul ca prima operatie de "citire secventiala/scriere secventiala" se va aplica asupra aceleiasi inregistrari care s-a preluat prin "citire directa" (intrucit octetul 32 nu a fost incrementat cu 1). Se poate, printr-o "citire secventiala" falsa, incrementa octetul 32 din FCB, astfel incit operatiile de "citire/scriere secventiala" urmatoare sa se aplice asupra inregistrarilor care urmeaza celei preluate prin "citire directa".

Daca operatia de "citire directa" s-a aplicat asupra ultimei inregistrari dintr-o "extensie logica", nu se realizeaza automat deschiderea "extensiei logice" urmatoare (ca in rutina 20).

Daca operatia de "citire directa" nu s-a efectuat corect, atunci registrul "A" va contine codul de eroare, care poate fi:

- 01; citirea unei inregistrari nescrise;
- 03; imposibilitate de inchidere a "extensiei logice" curente (trebuie redeschisa sau recitita "extensia logica" numarul 0 a fisierului);
- 04; acces la o "extensie logica" a fisierului care nu a fost creata;
- 06; octetul 35 este diferit de #00 (incercare de cautare in afara limitelor fizice ale fisierului).

In general, codurile de eroare diferite de #00 pot fi interpretate ca "lipsa de date".

RUTINA 34	: Scriere directa (Write Random)		
Intrari	: registrul C	: #22	
	registrele D&E	: adresa FCB	
Iesiri	: registrul A	: octet de stare	

Efect: rutina este identică cu rutina de "citire directa" cu exceptia faptului ca o inregistrare de 128 de octeti aflata in memorie la "adresa DMA" este scrisa pe disc. Inregistrarea se va scrie in fisier in pozitia corespunzatoare numarului ei (octetii 33 si 34). Daca acestei pozitii nu ii fusese alocat spatiu, atunci rutina realizeaza aceasta alocare inainte de scriere.

In urma unei operatii de "scriere directa" valorile octetilor 12 si 32 se modifica, dar octetul 32 nu se incrementeaza cu 1. Toate observatiile referitoare la rutina 33 sint valabile si pentru rutina 34.

Daca operatia de "scriere directa" s-a efectuat corect, atunci registrul "A" va avea valoarea #00; in caz contrar el va contine codul de eroare. Codurile de eroare posibile sint cele de la rutina 33 plus codul 05 care indica imposibilitatea scrierii datelor intrucit nu a mai fost spatiu in "director" pentru crearea unei noi "extensii logice".

RUTINA 35	: Determinare lungime fisier	
	(Compute File Size)	
Intrari	: registrul C : #23	
	registrele D&E : adresa FCB	
Iesiri	: lungimea virtuala a fisierului plasata	
	in octetii 33, 34 si 35 din FCB	

Efect: rutina necesita ca FCB-ul adresat prin registrele "D&E" sa aiba 36 de octeti si sa contina un specificator-individual de fisier. Rutina cauta in "director" informatii privind fisierul specificat in FCB si completeaza octetii 33, 34 si 35 cu o valoare egala cu numarul corespunzator primei inregistrari de pe disc, care urmeaza dupa sfirsitul fizic al fisierului. Astfel, octetii 33, 34 si 35 reprezinta "lungimea fisierului", lungime care poate fi:

- lungime reala a fisierului daca fisierul a fost creat secvential;
- lungime virtuala a fisierului (daca fisierul a fost creat in acces direct si exista "gauri" in alocarea fisierului).

Daca octetul 35 are valoare egala cu #01 rezulta ca fisierul contine numarul maxim de inregistrari posibile: 65536.

Rutina poate fi folosita pentru a adauga inregistrari intr-un fisier. Prin apelul ei se determina numarul de ordine al primei inregistrari neocupate de dupa sfirsitul fizic al fisierului, numar de ordine ce poate fi folosit in continuare de catre o secventa de operatii de "scriere directa".

RUTINA 36	: Determinare numar inregistrare	
	(Set Random Record)	
Intrari	: registrul C : #24	
	registrele D&E : adresa FCB	
Iesiri	: numarul inregistrarii plasat	
	in octetii 33, 34 si 35 din FCB	

Efect: rutina intoarce in octetii 33, 34 si 35 din FCB numarul inregistrarii curente dintr-un fisier care a fost citit/scris secvential. Rutina poate fi folosita astfel:

- pentru determinarea numarului de ordine al unor inregistrari dintr-un fisier, care contin o anumita "cheie". In acest caz, fisierul se parcurge secvential in citire, se verifica daca inregistrarea citita contine "cheia" cautata si daca o contine se apeleaza rutina 36 pentru a determina "numarul de ordine" al inregistrarii respective. Acest numar de ordine se stocheaza si apoi se continua investigarea secventiala a fisierului. La sfirsitul prelucrarii se va dispune de o lista a tuturor numerelor inregistrarilor care contin "cheia" respectiva. Pe baza acestei liste, utilizatorul poate citi direct inregistrarile care il intereseaza.
- atunci cind se doreste trecerea de la accesul secvential intr-un fisier la accesul direct. In acest caz, dupa ce un fisier a fost exploatat secvential pina la un anumit punct, se apeleaza rutina 36 pentru a determina "numarul de ordine" al inregistrarii curente. Pe baza acestui numar de ordine se pot realiza, in continuare, operatii de citire/scriere directa, operatii ce se aplica de la un anumit punct selectat din fisier in continuare.

```
-------------------------------------------------------
|RUTINA 37 : Dezactivare discuri                      |
|            (Reset Drive)                            |
|Intrari   : registrul C    : #25                     |
|            registrele D&E : vector discuri          |
-------------------------------------------------------
```

Efect: dezactiveaza unitatile de disc specificate in vectorul definit prin continutul registrelor "D&E" si acorda acestor unitati atributul R/W. Bitul B0 din registrul "E" corespunde unitatii "A" iar bitul B7 din registrul "D" corespunde unitatii "P". O valoare logica "1" in vectorul definit reprezinta o optiune-utilizator de "dezactivare" a unitatii respective.

Rutina se foloseste, de obicei, pentru a modifica atributul R/O, care a fost asociat unei unitati de disc prin apelul rutinei 28.

Discul selectat nu poate fi dezactivat prin aceasta rutina, ci numai printr-o rutina 13.

```
-------------------------------------------------------
|RUTINELE 38 si 39 sint rezervate pentru dezvoltari   |
|                  ulterioare ale sistemului.         |
-------------------------------------------------------
```

```
-------------------------------------------------------
|RUTINA 40 : Scriere directa cu umplere cu 0          |
|            (Write Random with Zero Fill)            |
|Intrari   : registrul C    : #28                     |
|            registrele D&E : adresa FCB              |
|Iesiri    : registrul A    : octet de stare          |
-------------------------------------------------------
```

Efect: rutina este similara rutinei 34 (scriere directa) cu exceptia faptului ca inainte de a se scrie o inregistrare, intr-un bloc nealocat, acesta este automat umplut cu zerouri. Astfel, toate "gaurile" dintr-un fisier creat in acces direct vor fi recunoscute prin continutul lor (zerouri).

6.7.6 Breviarul principalelor rutine CP/M

Nr. rut HEX	Denumire rutina	Intrari	Iesiri
0	1	2	3
0	Reinitializare sistem CP/M	C=#00	-
1	Citire caracter de la consola	C=#01	A=caracter ASCII
2	Scriere caracter la consola	C=#02 E=caracter ASCII	-
3	Citire caracter de la dispozitivul "Reader" curent	C=#03	A=caracter ASCII
4	Scriere caracter la dispozitivul "Punch" curent	C=#04 E=caracter ASCII	-
5	Scriere caracter la dispozitivul "List" curent	C=#05 E=caracter ASCII	-
6	Citire/Scriere directa la consola	C=#06 E=#FF =caracter ASCII	A=caracter ASCII sau =octet stare
7	Citire octet IOBYTE	C=#07	A=valoare octet IOBYTE
8	Modificare octet IOBYTE	C=#08 E=valoare pentru octetul IOBYTE	-
9	Tiparire la consola a unui sir de caractere	C=#09 D&E=adresa sir	-
0A	Citire buffer consola	C=#0A D&E=adresa buffer	-
0B	Citire stare consola	C=#0B	A=stare consola
0C	Citire versiune sistem	C=#0C	H&L=numar de versiune
0D	Initializare stare sistem discuri	C=#0D	-
0E	Selectare disc	C=#0E E=numar unitate selectata	-

0F	Deschidere fisier	C=#0F D&E=adresa FCB	A=octet stare
10	Inchidere fisier	C=#10 D&E=adresa FCB	A=octet stare
11	Cauta in "director" prima intrare	C=#11 D&E=adresa FCB	A=octet stare
12	Cauta in "director" urmatoarea intrare	C=#12	A=octet stare
13	Stergere fisier	C=#13 D&E=adresa FCB	A=octet stare
14	Citire secventiala	C=#14 D&E=adresa FCB	A=octet stare
15	Scriere secventiala	C=#15 D&E=adresa FCB	A=octet stare
16	Creare fisier	C=#16 D&E=adresa FCB	A=octet stare
17	Schimbare nume fisier	C=#17 D&E=adresa FCB	A=octet stare
18	Citire vector de unitati-disc active	C=#18	H&L=vectorul de unitati disc active
19	Citire numar disc selectat	C=#19	A=numar disc selectat
1A	Modificare "adresa DMA"	C=#1A D&E=adresa DMA	-
1B	Citire adresa vector de alocare	C=#1B	H&L=adresa vector de alocare
1C	Setare atribut R/O pentru o unitate de disc	C=#1C	-
1D	Citire vector de unitati R/O	C=#1D	H&L=vectorul de unitati R/O
1E	Modificare atribute fisier	C=#1E D&E=adresa FCB	A=octet stare
1F	Citire adresa "bloc de parametri disc"	C=#1F	H&L=adresa blocului de parametri disc
20	Citire/Modificare numar utilizator curent	C=#20 E=#FF =numar utilizator curent	A=numar utilizator sau nimic

21	Citire directa	C=#21 D&E=adresa FCB	A=octet stare
22	Scriere directa	C=#22 D&E=adresaFCB	A=octet stare
23	Determinare lungime fisier	C=#23 D&E=adresa FCB	lungime virtuala in octetii 33, 34 si 35 din FCB
24	Determinare numar inregistrare	C=#24 D&E=adresa FCB	numarul inregistrare
25	Dezactivare discuri	C=#25	A=#00
26	Rezervata		
27	Rezervata		
28	Scriere directa cu umplere cu zero	C=#28	A=octet stare

6.8 Programe utilitare de baza sub CP/M

6.8.1 Editor - ED

6.8.i.1 Prezentare generala

ED este un editor de texte contextual, care lucreaza sub sistemul de operare CP/M si este utilizat pentru crearea si modificarea fisierelor sursa. Lansarea lui in executie se face prin una dintre comenzile:

(1) **ED specificator-individual(CR);**
(2) **ED specificator-individual nume-unitate(CR).**

Forma (1) permite crearea unui fisier sursa (ASCII) pe disc, sau modificarea unui fisier existent deja pe disc. Prin aceasta forma, atit versiunea anterioara editarii, cit si fisierul rezultat in urma editarii se pastreaza pe aceeasi unitate de disc (unitate pe care se gaseste fisierul de editat).
Exemplu:

ED X.Y(CR)

editeaza fisierul **X.Y** de pe discul instalat si pastreaza versiunea anterioara editarii in fisierul X.BAK, iar fisierul rezultat din editare in **X.Y**. (Daca fisierul **X.Y** nu exista, el se creeaza in urma editarii).

Forma (2) permite utilizarea simultana a doua unitati de disc:

- o unitate, pe care se gaseste fisierul de editat si pe care se va pastra versiunea anterioara editarii;
- o unitate, pe care se va gasi fisierul rezultat din editare.

Daca unitatea specificata (pentru fisierul rezultat) este declarata R/O, atunci la consola va apare mesajul:

```
Bdos Err On x: R/O
```

unde "x" este numele unitatii specificate in comanda ED; procesul de editare se va intrerupe. Dupa apasarea oricarui caracter se face o reinitializare a sistemului, care deprotejeaza toate unitatile. Dupa verificarea discurilor montate si o noua reinitializare, comanda se reintroduce.
Exemplu:

ED X.Y B:(CR)

editeaza fisierul **X.Y** de pe discul "A", in urma editarii rezultind fisierele:

```
X.BAK          (pe discul "A")
               versiunea anterioara;
X.Y            (pe discul "B")
               rezultatul editarii.
```

Daca fisierul supus editarii exista deja si avea atributul R/O, la consola apare mesajul:

```
** FILE IS READ/ONLY **
```

care atentioneaza utilizatorul ca fisierul nu poate fi modificat, ci doar vizualizat. Daca fisierul exista deja si avea atributul SYS, la consola apare mesajul:

```
"SYSTEM" FILE NOT ACCESSIBLE
```

si procesul de editare se intrerupe. In ambele cazuri, utilizatorul trebuie sa modifice atributele fisierului de editat, prin comanda STAT.

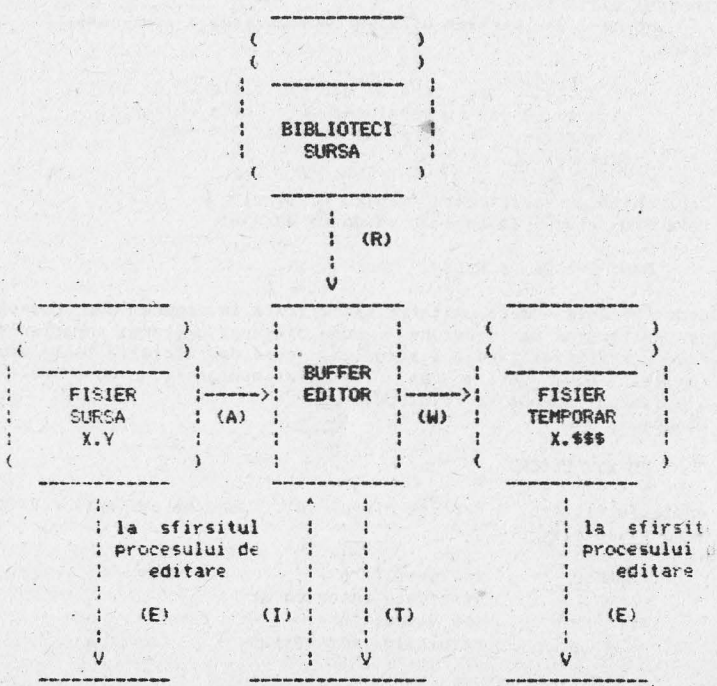
In forma (2), daca pe unitatea specificata in comanda exista deja un fisier cu nume identic cu cel al fisierului de editat, apare la consola mesajul:

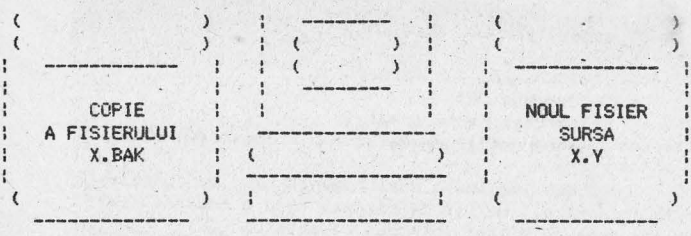
```
FILE EXISTS, ERASE IT
```

si procesul de editare se intrerupe. Utilizatorul trebuie sa utilizeze comanda ERA pentru a sterge intii acest fisier si apoi sa reia procesul de editare.

Editorul ED lucreaza intern cu un fisier temporar pe disc, cu nume identic cu cel al fisierului de editat si extensie "$$$", fisier ce se va gasi pe aceeasi unitate cu fisierul de editat (forma (1)) sau pe unitatea specificata in comanda (forma (2)).

Functionarea editorului ED este prezentata in figura urmatoare:





Nota: programul ED accepta de la consola atit caractere ASCII minuscule cit si caractere ASCII majuscule.

Procesul de editare consta in:

- transferarea continutului (integral sau partial) al fisierului sursa (de editat) in buffer-ul de memorie al editorului (daca acest fisier exista !);
- modificarea continutului buffer-ului de memorie prin comenzi ED (inserare text, modificare text, etc.);
- scrierea continutului (partial sau total) al buffer-ului de memorie in fisierul temporar;
- transferarea, la sfirsitul procesului de editare, a continutului integral al buffer-ului de memorie si a acelei parti din fisierul sursa care n-a fost citita (daca e cazul !) tot in fisierul temporar;
- modificarea numelui fisierului sursa in "x.BAK" si a fisierului temporar in "x.y".

Editorul ED lucreaza numai cu fisiere sursa (fisiere alcatuite din caractere ASCII). Toate operatiile de editare (modificare) se realizeaza asupra buffer-ului de memorie. Capacitatea buffer-ului este dependenta de versiunea de CP/M (ea poate fi cunoscuta prin intermediul comenzii V). Editorul ED lucreaza cu doua entitati:

- caractere;
- linii; (o linie este o succesiune de caractere, de orice lungime, terminata prin caracterele (CR)(LF)).

In buffer, liniile sint dispuse una dupa alta. Pentru a putea parcurge ordonat buffer-ul de memorie, acesta are asociat un "INDICATOR la nivel de caracter" (IC), care poate fi deplasat sub controlul utilizatorului. Indicatorul nu este vizibil la consola si el poate sa se gaseasca in una din pozitiile:

- inaintea primului caracter existent in buffer;
- dupa ultimul caracter existent in buffer;
- intre doua caractere consecutive existente in buffer.

Utilizatorul poate, in orice moment, sa afle pozitia IC-ului si respectiv sa modifice aceasta pozitie. IC-ul nu poate fi deplasat (prin comenzi utilizator) in afara limitelor buffer-ului de memorie (el se opreste cel mult pe una din limite). Prin conventie se numeste "linie curenta" linia in care se gaseste la un moment dat indicatorul.

6.8.1.2 Comenzi ED

Dialogul cu programul ED se face de la consola, prin inter-

mediul comenzilor ED, care pot fi:

- individuale, sub forma:
  comanda(CR);
- inlantuite, sub forma:
  comanda(1)comanda(2) . . . comanda(n)(CR).

Lungimea maxima a unei comenzi (sir de comenzi) ED este de 128 caractere. Un sir de comenzi (sau o comanda) este executata numai dupa ce a fost introdus caracterul (CR). In timpul introducerii unei comenzi (sir de comenzi) se pot folosi, pentru corectii, caracterele de editare ale sistemului CP/M (ex: (RUBOUT), (CTRL/U), (CTRL/R), etc). Toate caracterele de tip (CTRL/x) introduse in comenzile ED (cu exceptia caracterelor de editare ale sistemului CP/M) apar afisate la consola sub forma "^x".

Comenzile ED au forma generala:

[+/-n]nume-comanda(CR)

unde:

n - este un intreg, exprimat in zecimal, cu valori intre 0 si 65535;
nume-comanda - este o litera.

Exista urmatoarele conventii valabile pentru scrierea comenzilor ED:

- daca "n" este omis intr-o comanda in care el poate figura, se considera implicit n=1;
- daca semnul ("+/-") apare specificat in sintaxa unei comenzi si este omis la utilizare, se considera implicit egal cu "+";
- oriunde apare "n" in sintaxa unei comenzi, el poate fi inlocuit prin caracterul "#", care are semnificatia de "n=65535".

O particularitate a editorului ED este aceea ca el asigura accesul la texte atit prin context, cit si pe baza numarului de linie. Fiecare linie are un numar asociat, exprimat sub forma:

nnnnn:

numar cuprins intre 1 si 65535 ce poate fi referit in comenzi. De exemplu, forma:

nr-linie:

este echivalenta cu comanda: pozitioneaza-te in linia "nr-linie" iar forma:

:nr-linie comanda

este echivalenta cu: executa comanda de un numar de ori egal cu diferenta intre "nr-linie" si "numarul absolut al liniei curente".

O comanda de tipul:

nr-linie-1::nr-linie-2 comanda

este echivalenta cu: executa comanda incepind din linia cu numar egal cu "nr-linie-1" de k ori (unde k = nr-linie-2 - nr-linie-1).

Prima linie din buffer-ul de memorie are intotdeauna numarul 1 iar numerele de linie cresc din 1 in 1. Numerele de linie insotesc textul numai in buffer-ul editorului, ele nepropagindu-se in fisierul temporar/rezultat. Numerele de linie pot fi folosite pentru a precede orice comanda ED standard. Exista si posibilitatea de a anula numerotarea automata a liniilor (vezi comanda V). Comenzile ED se impart in:

- comenzi de I/E;
- comenzi de editare texte;
- comenzi de lucru cu biblioteci sursa;
- comenzi generale.

6.8.1.2.1 Comenzi de I/E

Realizeaza transferul de linii din fisierul sursa in buffer-ul de memorie, sau din buffer-ul de memorie in fisierul temporar (si eventual final). Modul de lucru al acestor comenzi este prezentat in figura urmatoare:

Organizarea buffer-ului de memorie

PS - indicator al urmatoarei linii care va fi citita din fisierul sursa (comanda A);

PM - indicator de inceput al zonei de memorie disponibila in buffer;

PT - indicator de inceput al zonei din fisierul temporar disponibila la momentul respectiv.

Comenzile de I/E sint:

nA(CR) - citeste "n" linii din fisierul sursa (incepind de la pozitia PS) si le transfera in buffer-ul de memorie (incepind din pozitia PM). Incrementeaza corespunzator PS si PM cu "n". Nu afecteaza pozitia IC-ului.

OA(CR) - citeste linii din fisierul sursa pina cind umple o jumatate din capacitatea buffer-ului de memorie. Incrementeaza corespunzator indicatorul PS. Nu afecteaza pozitia IC-ului.

#A(CR) - citeste linii din fisierul sursa pina cind se umple buffer-ul de memorie sau se ajunge la sfirsit de fisier sursa. Nu afecteaza pozitia IC-ului.

Nota:Daca intr-o comanda A se depaseste capacitatea buffer-ului de memorie, la consola este semnalata o eroare. Utilizatorul va

trebui sa foloseasca comanda W pentru a-si elibera buffer-ul. Depasirea capacitatii buffer-ului poate apare si in mijlocul unei linii, caz in care restul liniei va fi citit (preluat) prin urmatoarele comenzi A:

nW(CR) - scrie primele "n" linii din buffer in fisierul temporar (incepind din pozitia PT) si deplaseaza linia "n+1" la inceputul buffer-ului. Incrementeaza PT cu "n" si decrementeaza PM cu "n". Nu afecteaza pozitia IC-ului.

OW(CR) - scrie linii din buffer, in fisierul temporar, pina cind buffer-ul este cel putin jumatate gol.

E(CR) - termina procesul de editare si preda controlul sistemului CP/M (copiaza tot textul din buffer in fisierul temporar, copiaza toate liniile netratate din fisierul sursa in fisierul temporar si schimba numele fisierelor (asa cum a fost descris anterior)).

H(CR) - prin aceasta comanda fisierul temporar devine noul fisier sursa, buffer-ul de memorie este golit si se creaza un nou fisier temporar (cu PT pozitionat la inceputul acestuia). Comanda "H" este echivalenta comenzilor:

E(CR) **(comanda ED)**
ED x.y(CR) **(reapel program)**

IC se pozitioneaza pe inceputul buffer-ului.

O(CR) - revine la fisierul sursa original. Buffer-ul de memorie este golit, fisierul temporar este sters iar PS revine la pozitia 1 din fisierul sursa. Prin aceasta comanda se realizeaza anularea unor comenzi de editare efectuate anterior si se reia de la inceput procesul de editare. IC se pozitioneaza pe inceputul buffer-ului.

Q(CR) - abandoneaza lucrul cu editorul, fara a modifica fisierul sursa si preda controlul sistemului CP/M.

Nota: Comenzile E, H, O, Q trebuie introduse NUMAI ca o comanda individuala. Comenzile O si Q solicita la consola un raspuns (Y/N), ele fiind efective numai atunci cind raspunsul este "Y".

6.8.1.2.2 Comenzi de editare texte

Aceste comenzi se aplica numai asupra buffer-ului de memorie. Ele sint:

- comenzi la nivel de caracter;
- comenzi la nivel de linie;
- comenzi asupra sirurilor de caractere.

6.8.1.2.2.1 Comenzi la nivel de caracter

+/-B(CR)

deplaseaza IC la inceputul buffer-ului de memorie (daca s-a specificat semnul "+") sau la sfirsitul buffer-ului de memorie (daca s-a specificat semnul "-").

+/-nC(CR)

deplaseaza IC-ul in buffer peste "n" caractere (catre inceputul buffer-ului daca este specificat semnul "-" si catre sfirsitul acestuia in caz contrar).

+/-nD(CR)

sterge "n" caractere din buffer aflate inaintea IC-ului (daca s-a specificat semnul "-") sau aflate dupa acesta, in caz contrar.

6.8.1.2.2.2 Comenzi la nivel de linie

+/-nL(CR)

deplaseaza IC-ul in buffer peste "n" linii (spre inceputul buffer-ului daca semnul este "-" si spre sfirsitul acestuia, in caz contrar).

Nota: Daca n=0 atunci IC-ul este deplasat la inceputul liniei curente. Daca n este diferit de 0 atunci este deplasat intii IC-ul la inceputul liniei curente si apoi este mutat peste "n" linii.

+/-nK(CR)

sterge "n" linii din buffer, utilizind IC-ul ca punct de referinta. Daca IC nu se gaseste la inceputul liniei, atunci:

- daca s-a specificat semnul "+" se sterg (n - 1) linii aflate dupa linia curenta si toate caracterele din linia curenta aflate intre IC si sfirsitul liniei ((CR)(LF));
- daca s-a specificat semnul "-" se sterg caracterele din linia curenta aflate inaintea IC-ului si (n) linii anterioare liniei curente;
- daca n = 0 se sterg caracterele din linia curenta aflate inaintea IC-ului.

Comanda nu afecteaza pozitia IC-ului.

+/-nT(CR)

tipareste la consola continutul a "n" linii din buffer. Pot apare urmatoarele situatii:

- daca n este 0, tipareste continutul liniei curente pina la IC;
- daca n este 1, tipareste continutul liniei curente de la IC pina la sfirsitul liniei;
- daca n este mai mare ca 1 si semnul este "+", tipareste caracterele din linia curenta aflate dupa IC si (n - 1) linii care urmeaza liniei curente;
- daca n este mai mare ca 1 si semnul este "-", tipareste "n" linii precedente liniei curente si caracterele din linia curenta, pina la IC.

Nu afecteaza pozitia IC-ului. Comanda poate fi intrerupta (in timpul executiei) prin tastarea oricarui caracter.

+/-n(CR)

este echivalenta cu comanda +/-nLT, adica deplaseaza IC-ul peste "n" linii si tipareste continutul liniei curente.

nX(CR)

transfera "n" linii incepind de la linia curenta spre sfirsitul buffer-ului, intr-un fisier temporar denumit X$$$$$$$.LIB, care

este activ doar pe durata unui proces de editare. Utilizatorul poate transfera mai multe segmente de text din buffer in fisierul temporar (X$$$$$$$.LIB), utilizind succesiv comanda "X". In urma unei comenzi "X", liniile transferate nu se sterg automat din buffer. Daca n = 0 atunci comanda are ca efect stergerea continutului fisierului temporar (X$$$$$$$.LIB).

+/-nP(CR)

muta IC in buffer, peste "n" pagini (spre sfirsitul buffer-ului daca este "+" si spre inceputul lui daca este "-") si tipareste pagina respectiva. O pagina are 23 de linii. 'OP' tipareste 23 de linii fara sa mute IC-ul.

6.8.1.2.2.3 Comenzi asupra sirurilor de caractere

I(CR)
linia 1(CR)
linia 2(CR)
.
.
.
linia k(CR)
(CTRL/Z)

introduce de la consola, in buffer, incepind din pozitia curenta a IC, un text (alcatuit din mai multe caractere/linii). Fiecare linie se termina cu caracterul (CR), in buffer inregistrindu-se automat perechea de caractere (CR)(LF). Inserarea se termina atunci cind se tasteaza caracterul (CTRL/Z). IC ramine pozitionat in buffer dupa ultimul caracter introdus. In timpul inserarii textului se pot face corectii cu ajutorul caracterelor de editare ale sistemului CP/M (ex:(RUBOUT), (CTRL/R), etc.).

Itext(CTRL/Z)

este identica cu forma anterioara (text este o succesiune de caractere/linii).
Nota: Lungimea oricarei linii (de comanda sau text) nu trebuie sa depaseasca 128 de caractere. Daca acest lucru se intimpla, se forteaza sfirsit de linie dupa caracterul 128. Aceasta este singura regula care restringe lungimea unui sir de caractere din comenzile I, F, S, N si J.

Itext(CR)

este identica cu forma anterioara, cu deosebirea ca dupa ultimul caracter din text se mai introduce automat perechea de caractere (CR)(LF).
Nota: In acest caz sfirsitul unei linii se indica prin introducerea caracterului (CTRL/L).

nFc1c2...ck(CR)
sau
nFc1c2...ck(CTRL/Z)

cauta in buffer, incepind din pozitia curenta a IC pina la sfirsitul buffer-ului, sirul de caractere identice cu "c1c2...ck". Cautarea se efectueaza de "n" ori si daca s-a gasit al "n"-lea sir identic cu "c1c2...ck", atunci IC-ul se pozitioneaza dupa

caracterul "ck" din acest sir. In caz contrar, IC-ul nu se deplaseaza din pozitia initiala. Sirul "c1c2...ck" poate contine si caracterul (CTRL/L) care inlocuieste perechea (CR)(LF).

nSc1c2...ck(CTRL/Z)d1d2...dm(CR)
sau
nSc1c2...ck(CTRL/Z)d1d2...dm(CTRL/Z)

cauta in buffer, incepind din pozitia curenta a IC pina la sfirsitul buffer-ului, siruri de caractere identice cu "c1c2...ck" si le substitue prin sirul "d1d2...dm". Operatia de substitutie are loc de "n" ori sau pina se termina buffer-ul.

nNc1c2...ck(CR)
sau
nNc1c2...ck(CTRL/Z)

este similara cu comanda "F", cu deosebirea ca prin aceasta comanda se cauta a "n"-a aparitie a sirului "c1c2...ck" in tot fisierul sursa. Comanda se executa astfel:

- Se cauta in buffer-ul curent, incepind din pozitia curenta a IC-ului, a "n"-a aparitie a unui sir identic cu "c1c2...ck".
- Daca s-a gasit, atunci IC ramine pozitionat dupa caracterul "ck" din acest ultim sir.
- Daca nu s-a gasit, atunci se executa automat o comanda "#W" (se transfera tot continutul buffer-ului in fisierul temporar) si se citesc automat linii din fisierul sursa pina s-a umplut cel putin o jumatate din buffer sau pina s-a sfirsit fisierul sursa. Procesul de cautare continua si asupra noului buffer. Daca s-a gasit a "n"-a aparitie a sirului cautat, indicatorul IC ramine pozitionat dupa caracterul "ck" din sir. In caz contrar, comanda continua in acelasi mod, pina cind tot fisierul sursa a fost in intregime transferat in fisierul temporar.

Jc1c2...ck(CTRL/Z)d1d2...dm(CTRL/Z)e1e2...en(CR)
sau
Jc1c2...ck(CTRL/Z)d1d2...dm(CTRL/Z)e1e2...en(CTRL/Z)

aceasta comanda consta in cautarea sirului "c1c2...ck" incepind de la pozitia curenta a IC spre sfirsitul buffer-ului. Daca acest sir a fost gasit se insereaza, in acel punct, sirul "d1d2...dm" si se pozitioneaza IC dupa caracterul "dm". Apoi se sterg toate caracterele care existau intre IC si sirul "e1e2...en" (exclusiv acest sir), IC raminind pozitionat dupa caracterul "dm". Daca sirul "e1e2...en" nu este gasit atunci nu se efectueaza nici o stergere.

6.8.1.2.3 Comenzi de lucru cu biblioteci sursa

Aceste comenzi permit includerea unor biblioteci sursa in procesul de editare. Comanda este:

(1) **Rnume(CR)**
(2) **R(CR)**

unde "nume" este numele unui fisier sursa pe disc care are extensie "LIB".

Comanda "R" are ca efect citirea fisierului specificat si includerea lui in buffer, incepind din pozitia curenta a IC. Forma (2) este folosita atunci cind fisierul care se citeste este fisierul X$$$$$$$.LIB creat cu ajutorul comenzii "X". De remarcat faptul ca prin comanda "R", fisierul cu extensie "LIB" poate fi citit de mai multe ori.

6.8.1.2.4 Comenzi generale

nMc1c2...ck(CR)

unde "c1c2...ck" este sir de comenzi ED. Permite executia de "n" ori a sirului de comenzi ED "c1c2...ck". Sirul "c1c2...ck" nu trebuie sa contina o alta comanda "M". Daca n = 0 sau n = 1 atunci sirul de comenzi se executa de un numar nelimitat de ori pina cind apare o conditie de eroare (de exemplu: atunci cind s-a atins sfirsitul buffer-ului intr-o comanda "F").

+/-V(CR)

permite ca ED sa numeroteze liniile din buffer (daca este +V sau V = optiune implicita) sau ca ED sa nu numeroteze liniile din buffer (daca este -V).

0V(CR)

tipareste la consola mesajul:

nr-1 / nr-2

unde:

- nr-1 = numarul de octeti liberi din buffer (in zecimal);
- nr-2 = numarul total de octeti ai buffer-ului (dimensiunea buffer-ului exprimata tot in zecimal).

+/-U(CR)

transforma minusculele in majuscule (daca este U sau +U) sau nu face nici o transformare (daca este -U). Implicit, editorul considera ca fiind activa comanda -U. O comanda U ramine activa pina la o noua comanda -U. Daca este activa o comanda -U si se introduc cu litere mici comenzi de tipul I, F, S, N sau J, atunci sirurile de caractere implicate in aceste comenzi vor putea contine atit caractere minuscule cit si caractere majuscule (nefacindu-se nici o transformare asupra lor!). Daca insa, este -U si se introduc cu litere mari comenzi de tipul I, F, S, N, sau J, atunci, indiferent de tipul caracterelor (minuscule sau majuscule) din sirurile implicate in aceste comenzi, editorul automat le va trata ca siruri de caractere scrise cu majuscule. Daca este activa o comanda U, atunci, indiferent de caracterele cu care se introduc comenzile ED (sau sirurile de caractere din comenzile ED), editorul va transforma automat toate minusculele in majuscule.

6.8.1.3 Mesaje de eroare ED

In caz de eroare, ED tipareste mesajul:

BREAK "x" AT @

unde:

"x" - este un indicator de eroare;
@ - este comanda in care a aparut eroarea.

Indicatorul de eroare poate fi:

? - comanda nerecunoscuta sau o comanda E, H, Q sau O nu este singura intr-o linie de comanda;
> - buffer-ul de memorie este plin (trebuie utilizata una din comenzile "D", "K", "N", "S" sau "W" pentru a-l goli) sau sirurile de caractere din una din comenzile "F", "N" sau "S" sint prea lungi;
- ED nu poate gasi sirul specificat intr-o comanda F, S, sau N;
- comanda a fost intrerupta prin introducerea unui caracter de la consola;
O - fisierul cu extensia "LIB", specificat intr-o comanda "R" nu este gasit.

Daca la scrierea fisierului temporar se depaseste capacitatea discului, apare mesajul:

DISK OR DIRECTORY FULL

si se reinitializeaza sistemul. Utilizatorul trebuie sa se asigure inainte de a incepe editarea ca exista spatiu suficient pe discul destinatie.

Daca se detecteaza o eroare "CRC" intr-un fisier, apare mesajul:

Bdos Err on d: Bad Sector

unde "d" este numele unitatii de disc pe care a aparut eroarea.
Nota: "CRC" reprezinta o serie de informatii de "control ciclic redundant", care insotesc fiecare inregistrare dintr-un fisier si care sint puse de catre sistemul CP/M in momentul scrierii unui fisier. Aceste informatii sint folosite in operatiile de citire a fisierului, pentru a verifica daca datele au fost preluate corect.
La aparitia unei astfel de erori, utilizatorul poate sa ignore eroarea (tastind la consola orice caracter diferit de (CTRL/C)), sau sa reincarce sistemul CP/M (tastind (CTRL/C)) si sa reia procesul de editare pornind de la fisierul cu extensia "BAK" (daca acesta exista).
Nota: Ignorarea erorii trebuie sa fie urmata de o vizualizare a buffer-ului, pentru a se verifica daca datele citite au fost corect introduse in buffer.

Reincarcarea sistemului CP/M trebuie urmata de o secventa de comenzi CP/M de tipul:

TYPE X.BAK(CR)

unde "X" era numele fisierului care se edita in momentul aparitiei erorii, pentru a verifica daca fisierul "BAK" contine integral versiunea anterioara a fisierului care se edita.

ERA X.Y(CR)

pentru a sterge fisierul initial (care putea fi afectat de aban-

donarea procesului de editare).

REN X.Y=X.BAK(CR)

pentru a restaura fisierul initial.

ED X.Y(CR)

pentru a relua editarea pornind de la versiunea anterioara.

6.8.1.4 Caractere de control disponibile in ED

caracterul de control	functia realizata
(CTRL/C)	reincarcarea sistemului CP/M
(CTRL/E)	introduce un sfirsit fizic de linie la consola (in buffer nu se introduce nimic)
(CTRL/H)	sterge ultimul caracter introdus (pentru CRT)
(CTRL/I)	caractere de tabulare (coloanele 1, 9, 17,...)
(CTRL/L)	caracter ce inlocuieste perechea (CR)(LF) in cadrul sirurilor ce apar in comenzi de cautare/substitutie siruri
(CTRL/M)	echivalent cu (CR)
(CTRL/R)	afiseaza la consola continutul curent al bufferului consolei
(CTRL/U)	sterge linia introdusa de la consola (pentru TTY)
(CTRL/X)	sterge linia introdusa de la consola (pentru CRT)
(CTRL/Z)	terminator de sir de caractere
(RUBOUT)(DEL)	sterge ultimul caracter introdus (pentru TTY)

6.8.1.5. Comenzi disponibile in ED

comanda	functia realizata
nA	introduce in buffer linii din fisierul sursa
+/-B	inceput/sfirsit de buffer
+/-nC	muta IC peste "n" caractere
E	sfirsit proces de editare si inchidere fisiere (sfirsit normal al unei operatii de editare)
nF	cauta sir
H	sfirsit proces de editare, inchidere si redeschidere fisier
I	inserare text
J	plaseaza siruri prin juxtapunere
+/-nK	sterge linii
+/-nL	muta IC peste "n" linii
nM	comenzi repetitive
nN	cauta un sir parcurgind automat tot fisierul sursa
O	revine la fisierul sursa original
+/-nP	muta IC peste "n" pagini si tipareste pagina respectiva
Q	abandoneaza procesul de editare fara a afecta fisierele
R	citeste date dintr-o biblioteca sursa
nS	substitutie sir
+/-nT	tipareste "n" linii

nW	scrie linii din buffer in fisierul temporar	
+/-U	transforma minusculele in majuscule	
+/-n	muta IC peste "n" linii si tipareste (+/-nL-T)	
nZ	introduce o "intirziere" de aproximativ "n" secunde	

6.8.2 Asamblor - ASM

6.8.2.1 Prezentare generala

ASM este un asamblor care lucreaza sub sistemul de operare CP/M si care trateaza fisiere sursa in limbaj de asamblare rezidente pe disc, in urma procesului de asamblare rezultind un fisier cod-obiect (cod-masina 8080) in format hexa (compatibil INTEL) si un fisier listing.

Lansarea in executie a programului ASM se face prin una din comenzile:

(1) **ASM [nume-unitate]nume-fisier(CR)**
(2) **ASM nume-fisier.p1p2p3(CR)**

In ambele cazuri, programul ASM presupune ca fisierul de asamblat (fisier sursa in limbaj de asamblare 8080) se gaseste pe disc, sub un nume egal cu "nume-fisier" si cu extensie standard ".ASM".

Forma (1) precizeaza ca asamblorul va trata ca fisier sursa fisierul **nume-fisier.ASM** aflat pe discul instalat sau pe unitatea specificata, in scopul producerii, la iesire, au uneia din urmatoarele versiuni de fisier rezultat:

(1) **fisierul cod obiect** - (format hexa), (pe discul instalat sau pe unitatea specificata); forma generala:
nume-fisier.HEX.
(2) **fisierul listing** - (pe discul instalat sau pe unitatea specificata); forma generala:
nume-fisier.PRN.

Forma (2) precizeaza ca asamblorul va trata ca fisier sursa fisierul **nume-fisier.ASM** aflat pe unitatea specificata prin parametrul p1 si sa genereze sau nu fisierele cod-obiect/listing, in functie de optiunea utilizatorului, exprimata prin parametrii p2 si p3. Semnificatia parametrilor este:

p1 - poate fi o litera (A - P), care indica numele unitatii de disc pe care se gaseste fisierul sursa;
p2 - poate fi:- numele unitatii de disc pe care se va genera fisierul cod-obiect (A - P);
- litera "Z" care indica faptul ca se solicita suprimarea generarii fisierului cod-obiect.
p3 - poate fi:- numele unitatii de disc pe care se va genera fisierul listing (A - P);
- litera "X" care indica faptul ca se solicita obtinerea fisierului listing la consola;
- litera "Z" care indica faptul ca se solicita suprimarea generarii fisierului listing.

De exemplu comanda:

ASM PROG.BBX(CR)

cere asamblarea programului aflat in fisierul sursa B:PROG.ASM, generarea codului obiect in fisierul B:PROG.HEX si listare pe consola.

6.8.2.2 Formatul fisierului sursa

Un program in limbaj de asamblare, acceptat de ASM, este alcatuit dintr-o succesiune de linii sursa care au formatul general:

numar-linie eticheta cod-operatie operand ;comentariu

O linie poate contine oricare din elementele specificate in formatul general, sau toate aceste elemente. Fiecare linie se termina prin caracterele (CR)(LF). O linie poate contine una sau mai multe instructiuni/directive, separate intre ele prin caracterul "!". Este deci, acceptata forma generala:

nr.linie eticheta instr(1)!instr(2)!... instr(n) (CR)(LF)

unde instr(i) contine:

cod-operatie operanzi [;comentariu]

Nota:Caracterul "!" este tratat de catre ASM drept sfirsit logic de linie (sfirsit de instructiune).

O linie care incepe cu caracterul ";" sau "*" in coloana 1 este considerata "linie comentariu" si nu este tratata de asamblor.

6.8.2.2.1 Cimpul "numar linie"

Este un cimp optional ce reprezinta numarul liniei sursa (exprimat in zecimal). Asamblorul ignora acest cimp.

6.6.2.2.2 Cimpul "eticheta"

Este un cimp optional, care are forma:

(1) **eticheta**
(2) **eticheta:**

unde "eticheta" este un sir de maximum 16 caractere alfanumerice, cu primul caracter obligatoriu alfabetic. Acest cimp poate contine si caracterul "$", care nu este tratat de ASM (il ignora), el folosind doar pentru cresterea lizibilitatii etichetelor (ex: nume$lung sau data$foarte$lunga).

6.8.2.2.3 Cimpul "cod operatie"

Acest cimp poate contine o mnemonica 8080 sau o directiva a asamblorului ASM. Directivele acceptate de ASM sint:

ORG - stabileste originea programului (adresa incepind de la

care asamblorul va genera cod-obiect);
END - indica sfirsitul unui program sursa in limbaj de asamblare (si, optional, adresa de lansare automata in executie a programului);
EQU - atribuie valori unui simbol;
SET - atribuie valori temporare unui simbol;
IF - directiva pentru asamblare conditionata;
ENDIF - sfirsit asamblare conditionata;
DB - generare date de lungime egala cu un octet fiecare;
DW - generare date de lungime egala cu doi octeti fiecare;
DS - rezerva o zona de memorie.

6.8.2.2.4 Cimpul "operand"

Acest cimp contine, in general, expresii alcatuite din operanzi si operatori. Expresiile sint evaluate de asamblor in timpul procesului de asamblare ca valori pe 16 biti.

6.8.2.2.4.1 Operanzi

Operanzii pot fi:

-**etichete**; pot fi etichete definite de utilizator in program, sau etichete carora li s-au atribuit valori prin directivele EQU sau SET.
-**constante numerice**; pot fi exprimate in binar (B), octal (O sau Q), zecimal (D) sau hexazecimal (H). Orice constanta numerica reprezinta o valoare pe 16 biti. Orice constanta numerica se scrie sub forma:

c1c2...cnb

unde c(i) sint cifre apartinind bazei de numeratie "b", iar "b" reprezinta baza in care este exprimata constanta (literele B, O, Q, D sau H). In absenta lui "b", se considera "b"="D". O constanta hexa trebuie sa inceapa intotdeauna cu o cifra. Orice constanta este transformata de asamblor intr-o constanta binara, reprezentata pe 16 biti. Daca reprezentarea constantei depaseste 16 biti, automat se trunchiaza la dreapta. In scrierea constantelor se pot folosi caracterele "$" (ignorate de asamblor) pentru a creste lizibilitatea (ex: 33FF44H).
-**constante sir**; sint siruri de caractere ASCII incluse intre caracterele apostrof ('). O constanta sir trebuie sa fie scrisa pe o singura linie (fizica !) si sa aiba maximum 64 caractere. Caracterul ' poate apare intr-o constanta sir prin dublarea lui. Valoarea unui caracter dintr-un sir este egala cu codul ASCII corespunzator lui.
-**cuvinte rezervate**; acestea sint nume de registre generale sau coduri de instructiuni masina care au valori prestabilite. Astfel, exista urmatoarele cuvinte rezervate:

A	cu valoare asociata	7
B	cu valoare asociata	0
C	cu valoare asociata	1
D	cu valoare asociata	2
E	cu valoare asociata	3
H	cu valoare asociata	4
L	cu valoare asociata	5
M ((HL))	cu valoare asociata	6

SP cu valoare asociata 6
PSW (F) cu valoare asociata 6

grupul dintre primul nivel de paranteze constituie reprezentarea in limbaj de asmblare Z-80.

-**instructiuni 8080 cu valoare asociata**: codul-masina corespunzator tipului instructiune respectiv (operanzii instructiunii fiind implicit inlocuiti cu valoarea 0 (zero)). De exemplu, LD A,MOV se asambleaza la fel cu LD A,40H).

-**$**; adresa la care va fi generata instructiunea/directiva curenta. Caracterul $ trebuie sa apara separat prin blancuri de identificatorii din context.

6.8.2.2.4.2 Operatori

Operatorii pot fi:

aritmetici: + - * / MOD ( ) + (unar) - (unar)
logici: NOT AND OR XOR SHL SHR

Semnificatia acestor operatori este:

operator	utilizare	semnificatie
+	a + b	adunare fara semn a lui "a" cu "b"
-	a - b	scadere fara semn a lui "b" din "a"
+ (unar)	+ b	identic cu "b"
- (unar)	- b	identic cu "0 - b"
*	a * b	inmultire fara semn a lui "a" cu "b"
/	a / b	impartire intreaga (fara semn) a lui "a" la "b"
MOD	a MOD b	restul impartirii intregi a lui "a" la "b"
NOT	NOT b	complementul fata de 1 al valorii lui "b" (toti bitii "0" din "b" devin "1" si toti bitii "1" din "b" devin "0")
AND	a AND b	"si"logic intre "a" si "b"
OR	a OR b	"sau" logic intre "a" si "b"
XOR	a XOR b	"sau-exclusiv"logic intre "a"si"b"
SHR	a SHR b	valoarea care rezulta prin deplasarea la dreapta a valorii "a" cu "b" pozitii binare (pozitiile ramase libere se completeaza cu zerouri).
SHL	a SHL b	valoarea care rezulta prin deplasarea la stinga a valorii "a" cu "b" pozitii binare (pozitiile ramase libere se completeaza implicit cu zerouri).

In fiecare din cazurile prezentate, "a" si "b" reprezinta operanzi (etichete, constante numerice, cuvinte rezervate sau siruri de 1 sau 2 caractere ASCII) sau subexpresii incluse intre paranteze.
Exemple:

```
10+20  10H+37Q  A1/3  (A2+4) SHR 3  ('A' AND 5FH)+'0'
('B'+B) OR (PSW+M)        (1+(2+C)) SHR (A-(B+1))
```

Evaluarile tuturor expresiilor se realizeaza in faza de asamblare (ca operatii pe 16 biti fara semn). Astfel, (-1) este evaluat ca (0 - 1), adica rezulta valoarea #0FFFF.

Valoarea calculata a unei expresii trebuie sa corespunda lungimii operanzilor implicati in instructiuni. Daca, de exemplu, o expresie este utilizata intr-o instructiune ADI, atunci primii 8 biti ai valorii expresiei (cei mai semnificativi) tre-

buie sa fie egali cu zero. Astfel, instructiunea:

```
ADI     -1
```

va produce un mesaj de eroare, intrucit valoarea #0FFFF nu poate fi reprezentata pe 8 biti, in timp ce instructiunea:

```
ADI     (-1) AND 0FFH
```

este acceptata de ASM, intrucit anuleaza primii 8 biti ai expresiei.

Evaluarea expresiilor se face astfel:

- intii parantezele cele mai interioare;
- apoi, in cadrul unei paranteze sau in absenta acestora, in ordinea: * / MOD SHL SHR - + NOT AND OR XOR

Exemple:

```
a*b+c = (a*b)+c
a+b*c = a+(b*c)
a MOD b*c SHL d = ((a MOD b)*c) SHL d
a OR b AND NOT c+d SHL e = a OR (b AND (not (c+(d SHL e))))
(a OR b) AND (not c)+d SHL e=(a OR b)AND((NOT c)+(d SHL e))
```

6.8.2.2.5 Cimpul "comentariu"

Acest cimp contine orice sir de caractere care urmeaza caracterului ";" si care se termina la sfirsitul liniei curente (caracterele (CR)(LF)), sau la sfirsitul logic al liniei (caracterul "!"). Acest cimp este citit de asamblor, este listat in fisierul listing, dar nu este interpretat de asamblor.

6.8.2.3 Mesaje de eroare ASM

Erorile detectate de asamblor apar in prima coloana a fisierului listing. Ele apar tiparite si la consola, pe masura ce se genereaza fisierul listing (astfel, nu este obligatoriu sa se listeze fisierul listing pentru aflarea eventualelor erori in programul sursa).

Semnalarea erorilor se face prin urmatoarele coduri:

D - eroare de data: un element din datele instructiunii nu poate fi plasat in zona de date specificata;
E - eroare de expresie: expresia nu poate fi evaluata in timpul asamblarii;
L - eticheta eronata: eticheta nu poate apare in acest context (poate fi eticheta dubla);
N - neimplementat: directiva este recunoscuta de asamblor dar nu este tratata de catre acesta (este neimplementata inca);
O - depasire: expresia este prea complexa pentru a putea fi evaluata (trebuie simplificata !);
P - eroare de faza: eticheta nu are aceeasi valoare in doi pasi succesivi de parcurgere de catre asamblor a programului sursa;
R - eroare de registru: valoarea specificata ca registru nu este compatibila cu tipul de instructiune;
S - eroare de sintaxa: expresia este incorect construita;
U - eticheta nedefinita: specificatorul fisierului sursa, in

comanda, este fie incorect introdus, fie tipul fisierului nu este ASM;

V - eroare de valoare: operandul intilnit in expresie este incorect construit.

Asamblorul ASM semnaleaza erori datorate unor comenzi incorecte de apel al asamblorului. Aceste mesaje de eroare sint:

NO SOURCE FILE PRESENT

asamblorul nu poate sa gaseasca fisierul sursa specificat.

NO DIRECTORY SPACE

fisierul "director" al discului este plin; trebuie sterse fisierele care nu mai sint utile si apoi reapelat asamblorul.

SOURCE FILE NAME ERROR

numele fisierului sursa specificat in comanda este eronat (de exemplu contine caracterul "?" sau "*").

SOURCE FILE READ ERROR

fisierul sursa specificat in comanda nu poate fi citit corect de catre asamblor; trebuie utilizata comanda TYPE pentru a determina locul din fisier in care apare eroarea.

OUTPUT FILE WRITE ERROR

fisierul/fisierele de iesire (cod-obiect si/sau listing) nu pot fi scrise pe disc (in majoritatea cazurilor din lipsa de spatiu-disc; trebuie eliberat spatiul-disc si reluat procesul de asamblare).

CANNOT CLOSE FILE

fisierul de iesire nu poate fi inchis: trebuie verificat daca exista disc in unitate si daca acesta nu este R/O; aceasta este o eroare fatala care determina terminarea executiei programului ASM.

6.8.2.4 Directive acceptate de ASM

[eticheta] ORG expresie

determina generarea codului obiect incepind de la adresa egala cu valoarea expresiei (de obicei aceasta valoare este sub CP/M egala cu #100 care reprezinta inceputul zonei TPA). Daca eticheta exista, ii atribuie valoarea "expresiei".

[eticheta] END [expresie]

indica sfirsitul unui program sursa in limbaj de asamblare (liniile aflate dupa aceasta directiva nu vor fi tratate de asamblor). Daca "expresie" este prezenta, atunci ea va fi evaluata si va reprezenta "adresa de lansare in executie" a programului (aceasta "adresa" va fi inscrisa in ultima inregistrare a fisierului hexa generat de asamblor). Daca "expresie" nu este prezenta, atunci "adresa de lansare in executie" a programului va fi

#0000.

```
etichetа    EQU     expresie
```

evalueaza "expresie" si atribuie valoarea simbolului "eticheta". Simbolul "eticheta" nu trebuie sa mai apara ca eticheta in programul respectiv.

```
etichetа    SET     expresie
```

evalueaza "expresie" si atribuie aceasta valoare simbolului "eticheta". Simbolul "eticheta" va avea valoarea atribuita prin directiva SET pina in momentul in care i se va atribui o noua valoare, printr-o alta directiva SET.

```
        IF      expresie
linie sursa 1
linie sursa 2
  .
  .
  .
linie sursa n
    ENDIF
```

evalueaza "expresie" si daca valoarea rezultata este diferita de zero, atunci instructiunile din liniile 1 - n vor fi asamblate. Daca valoarea rezultata este zero, atunci instructiunile din liniile 1 - n vor fi ignorate de asamblor (nu se va genera cod pentru ele).

```
[eticheta]  DB      exp(1),...exp(n)
```

unde exp(i) este o expresie a carei valoare incape pe 8 biti (valoarea expresiei are primii 8 biti nuli) si genereaza valorile corespunzatoare expresiilor "exp(1)",..."exp(n)" pe cite un octet fiecare. "Exp(i)" pot fi si constante-sir, cu lungime maxima de 64 caractere. Fiecare caracter din constanta-sir va fi generat pe cite un octet (in cod ASCII).

```
[eticheta]  DW      exp(1),...exp(n)
```

unde "exp(i)" este o expresie a carei valoare incape pe 16 biti. Directiva genereaza valorile corespunzatoare expresiilor "exp(1)",..."exp(n)" pe cite doi octeti fiecare. Daca "exp(i)" este o constanta-sir, atunci lungimea ei maxima este de 2 caractere.

```
[eticheta]  DS      expresie
```

evalueaza "expresie" si rezerva o zona de memorie de "n" octeti (cu "n" egal cu valoarea expresiei).

6.8.3 Depanator - ZSID

6.8.3.1 Prezentare generala

Programul ZSID permite testarea si depanarea interactiva a programelor. Se lanseaza in executie prin una din comenzile:

```
(1) ZSID(CR)
```

(2) **ZSID [dispozitiv:]nume-fisier.HEX(CR)**
(3) **ZSID [dispozitiv:]nume-fisier.COM(CR)**

unde:

dispozitiv este numele unitatii de disc (A - P) pe care se afla programul (fisierul);
nume-fisier este numele programului care va fi incarcat in memorie si testat.

Formele (2) si (3) realizeaza, in plus fata de forma (1), si incarcarea in memorie a programului (fisierului) specificat. Toate formele de apel ZSID realizeaza incarcarea, de pe disc, a programului ZSID, in locul componentei CCP (vezi si sectiunile de interfata CP/M").

Dialogul cu programul ZSID se face prin comenzi introduse de la consola. Dupa lansarea in executie a programului ZSID si ori de cite ori ZSID asteapta o comanda, la consola apare semnul "-".

Comenzile ZSID au forma generala:

nume-comanda[parametri](CR)

unde:
nume-comanda este o litera;
parametri pot fi 0, 1, 2 sau 3 valori hexazecimale separate intre ele prin blanc sau ",". Fiecare parametru poate contine 1 - 4 cifre hexa. Parametrii mai lungi de 4 cifre hexa sint trunchiati la dreapta.

In timpul introducerii oricarei comenzi sint active caracterele de editare ale sistemului CP/M (ex: RUBOUT, CTRL/U, CTRL/R, etc.).

Orice comanda ZSID poate contine maximum 32 de caractere (al 33-lea caracter, automat inserat de catre ZSID, este (CR)).

Executia programului ZSID poate fi oprita in orice moment prin introducerea caracterului CTRL/C sau prin intrerupere pe nivelul zero (reinitializare CP/M). Controlul trece in sistemul CP/M si imaginea-memorie a programului care a fost testat/depanat cu ajutorul ZSID poate fi salvata pe disc cu ajutorul comenzii:

SAVE n [unitate:]nume-fisier.COM

unde:

n este numarul de pagini de memorie (de cite 256 octeti fiecare) care trebuie salvate pe disc ("n" este exprimat in zecimal);
unitate si **nume fisier** reprezinta numele unitatii de disc si al fisierului in care se va salva imaginea-memorie.

Nota:"n" se calculeaza pornind de la valoarea octetului cel mai semnificativ din adresa afisata de ZSID, ca adresa a primei locatii de memorie disponibila dupa ce programul de testat/depanat cu ZSID-ul a fost incarcat in memorie. De exemplu, daca ZSID afiseaza mesajul:

```
NEXT  PC
12A6 0100
```

atunci, 12A6 va fi adresa primei locatii de memorie disponibila, iar "n" va fi egal cu 18 (adica #12).

6.8.3.2 Comenzi ZSID

Programul ZSID permite:

- incarcarea de pe disc a unui program (fisier) de testat/depanat;
- dezasamblarea programului cod-masina existent intr-o zona de memorie;
- vizualizarea si/sau modificarea continutului unor locatii de memorie si/sau registre ale unitatii centrale;
- corectarea unui program cod-obiect existent in memorie prin introducerea, la adrese specificate, a unor instructiuni in limbaj de asamblare (programul ZSID realizeaza automat asamblarea acestora si introducerea codului-masina corespunzator in memorie);
- lansarea in executie a unui program existent in memorie si controlul executiei acestuia.

In prezentarea sintaxei/efectului comenzilor ZSID se vor folosi urmatorii termeni:

adresa

valoare hexazecimala (maximum 4 cifre hexa) ce indica o adresa absoluta de memorie.

constanta

valoare hexazecimala reprezentabila pe un octet (maximum 2 cifre hexa).

stare sistem

valoarea la un moment dat a registrelor generale (A, B, C, D, E, H, L), a registrelor SP si PC si a indicatorilor de conditie (Z, S, P, C, AC).

Nota: Codificarea folosita de ZSID pentru aceste elemente este:

C - indicatorul C (carry);
Z - indicatorul Z (zero);
M - indicatorul S (sign);
E - indicatorul P (parity);
I - indicatorul AC (auxiliary carry);
A - registrul A;
B - registrele B si C;
D - registrele D si E;
H - registrele H si L;
S - registrul SP;
P - registrul PC.

adresa de afisare

adresa de memorie incepind de la care va fi vizualizat continutul memoriei.

adresa de dezasamblare

adresa de memorie incepind de la care va incepe procesul de

dezasamblare.
Nota:Operatia de dezasamblare consta in scrierea textului-sursa corespunzator continutului unor locatii de memorie (se presupune ca locatiile de memorie contin un program cod-masina). Exemplu:

```
23        INC  HL
210001    LD   HL,0100H
010304    LD   BC,0403H
```

6.8.3.2.1 Comanda D (display)

Forma generala:

(1) **D(CR)**
(2) **Dadresa(CR)**
(3) **Dadresa1,adresa2(CR)**

Afiseaza la consola continutul unei zone de memorie. Forma (1) afiseaza la consola, incepind de la "adresa de afisare curenta", 16 linii a cite 16 octeti. Initial, adresa de afisare curenta este #100. Aceasta "adresa de afisare" se modifica dupa fiecare comanda D (ia o valoare egala cu "adresa ultimei locatii afisate printr-o comanda D anterioara" + 1).
Forma (2) afiseaza la consola 16 linii a cite 16 octeti incepind de la "adresa".
Forma (3) afiseaza la consola continutul zonei de memorie cuprinsa intre "adresa1" si "adresa2".
Formatul standard de afisare este:

```
aaaa bb bb bb bb bb bb bb bb bb bb bb bb bb bb bb cccccccccccc
```

unde:**aaaa** - este o valoare hexazecimala ce indica adresa de inceput a zonei de memorie vizualizata;
bb - este continutul in hexa al fiecarei locatii de memorie (incepind de la adresa "aaaa");
cc...c - este textul ASCII asociat configuratiei hexa bb...bb.

Nota:Pentru codurile care corespund unor caractere netiparibile va apare caracterul "." In sirul "cc...c" vor apare (daca terminalul accepta acest lucru) si caractere minuscule si caractere majuscule, in functie de codurile "bb".
Afisarile prea lungi, pot fi oprite prin tastarea caracterului RUBOUT (DEL).

6.8.3.2.2 Comanda F (fill)

Forma generala:

Fadresa1,adresa2,constanta(CR)

Umple zona de memorie cuprinsa intre "adresa1" si "adresa2" cu constanta specificata.

6.8.3.2.3 Comanda M (move)

Forma generala:

Madresa1,adresa2,adresa3(CR)

Muta continutul zonei de memorie cuprinsa intre "adresa1" si adresa2" in zona care incepe la "adresa3".

6.8.3.2.4 Comanda S (substitute)

Forma generala:

Sadresa(CR)

Permite vizualizarea si/sau modificarea unor locatii de memorie incepind de la "adresa". Comanda afiseaza automat mesajul:

aaaa bb

unde "aaaa" este adresa locatiei curente iar "bb" este continutul acestei locatii. Daca utilizatorul introduce caracterul (CR) va apare afisata adresa si continutul urmatoarei locatii de memorie. Daca utilizatorul introduce o constanta (2 cifre hexa) si (CR), atunci va fi modificat continutul locatiei respective cu valoarea introdusa. Comanda S se termina atunci cind utilizatorul introduce caracterul "." sau atunci cind s-a introdus eronat o constanta.

6.8.3.2.5 Comanda X (examine)

Forma generala:

(1) **X(CR)**
(2) **Xlitera(CR)**

permite vizualizarea si/sau modificarea starii curente a sistemului. Forma (1) afiseaza la consola:

CxZxMxExIx A=bb B=dddd D=dddd H=dddd S=dddd P=dddd instructiune

unde:

x, bb, dddd - reprezinta valorile curente ale indicatorilor de conditie si registrelor unitatii centrale;
instructiune - reprezinta textul sursa corespunzator instructiunii (cod-masina) de la adresa data in registrul P (PC-ul curent).

Forma (2) permite vizualizarea/modificarea continutului unui indicator de conditie sau al unui registru. In aceasta forma "litera" este una din literele (C, Z, M, E, I, A, B, D, H, S, P), litera i indica numele indicatorului/registrului de vizualizat/modificat. In urma unei astfel de comenzi apare tiparit numele elementului implicat in comanda si valoarea lui curenta. Utilizatorul poate introduce:

(1) **(CR)** - se termina comanda fara nici o modificare;
(2) **valoare(CR)** - se modifica continutul elementului specificat si se termina comanda X, simbolul **valoare** putind fi:
- 0 sau 1, daca se modifica un indicator de conditie;
- 2 cifre hexa, daca se modifica A;

- 4 cifre hexa, daca se modifica registrele B...P.

6.8.3.2.6 Comanda H (hexa)

Forma generala:

Hadresa1,adresa2(CR)

Comanda calculeaza si afiseaza suma si diferenta hexa intre adresa1 si adresa2.

6.8.3.2.7 Comanda G (go)

Forma posibile:

(1) **G(CR)**
(2) **G,adresa2(CR)**
(3) **G,adresa2,adresa3(CR)**
(4) **Gadresa1(CR)**
(5) **Gadresa1,adresa2(CR)**
(6) **Gadresa1,adresa2,adresa3(CR)**

unde "adresa2", "adresa3" sint adrese de "puncte de intrerupere" (breakpoint) in executia unui program.

Comanda realizeaza lansarea in executie a unui program existent in memorie.

Executia programului incepe de la:

- valoarea curenta a registrului PC (formele 1, 2, 3);
- "adresa1" (formele 4, 5, 6).

Executia programului se termina atunci cind:

- continutul registrului PC este egal cu "adresa2" sau "adresa3" (formele 2, 3, 5, 6);
- s-a executat instructiunea RST 7 (care intoarce controlul in ZSID);
- s-a executat o instructiune:

  RST 0, sau
  JP 0, sau
  CALL 0

  (care intorc controlul in CP/M cu reinitializarea sistemului CP/M)

Formele (2), (3), (5) si (6) permit executia controlata a unor programe (se specifica in comanda G adresa urmatorului punct de intrerupere). Oprirea executiei programului, ca urmare a atingerii unei adrese de punct de intrerupere sau ca urmare a excutiei unei instructiuni RST 7 are ca efect aparitia la consola a mesajului:

*adresa

unde "adresa" este valoarea PC-ului curent.
Starea sistemului ramine nemodificata si utilizatorul poate vizualiza/modifica continutul unor locatii de memorie/registre prin comenzile D, M, F, S, X.

Nota: 1. Exista o forma particulara a comenzii G si anume:

GO(CR)

Aceasta comanda intoarce controlul, din ZSID, in CP/M.

2. Daca un program se termina cu o instructiune RST 0 sau JP 0 sau CALL 0 este indicat sa se inlocuiasca aceasta instructiune (numai pe perioada testarii/depanarii programului) printr-o instructiune RST 7; prin aceasta, la sfirsitul executiei programului, controlul va fi intors tot in programul ZSID. Se poate evita efectuarea acestei modificari daca utilizatorul foloseste intotdeauna un "punct de intrerupere" pe adresa corespunzatoare instructiunii RST 0/JP 0/CALL 0.
3. In formele (3) si (6), daca s-a atins unul din punctele de intrerupere specificate prin "adresa2" si "adresa3", executia programului se suspenda, controlul trece in ZSID si celalalt punct de intrerupere prevazut (dar neatins!) este automat sters (nu ramine in continuare valabil!).

6.8.3.2.8 Comanda T (trace)

Forme posibile:

(1) T(CR)
(2) Tn(CR)

Aceasta comanda permite vizualizarea "starii curente a sistemului" (in formatul specific comenzii X) si executia uneia sau mai multor instructiuni din programul supus testarii.

In urma unei comenzi T, executia programului se opreste si la consola apare mesajul:

*adresa

unde "adresa" este valoarea PC-ului curent si anume adresa urmatoarei instructiuni de executat.

"Adresa curenta de afisare" (folosita de catre comanda D) devine egala cu continutul registrelor H si L, iar "adresa curenta de dezasamblare" (folosita de catre comanda L) devine egala cu "adresa". Utilizatorul poate vizualiza "starea curenta a sistemului", dupa executia comenzii T, prin comanda X.

Forma (1) a comenzii T permite vizualizarea starii curente a sistemului si executia unei instructiuni.

Forma (2) a comenzii T permite executia a "n" instructiuni (n exprimat in hexazecimal), cu vizualizarea la consola a starii sistemului inaintea fiecarei instructiuni. Aceasta comanda se termina dupa executia a "n" instructiuni, la intilnirea unui "punct de intrerupere" (RST 7) sau atunci cind s-a introdus de la consola caracterul RUBOUT (DEL), care forteaza un "punct de intrerupere" in timpul comenzii T.

Nota: Executia unui program cu comanda T se realizeaza de aproximativ 500 de ori mai lent decit executia normala a programului, intrucit ZSID preia controlul dupa fiecare instructiune executata.

6.8.3.2.9 Comanda U (untrace)

Forme posibile:

(1) **U(CR)**
(2) **Un(CR)**

Comanda U este similara ca functiune cu comanda T, cu exceptia faptului ca starea sistemului nu mai apare afisata la consola la fiecare pas intermediar (instructiune intermediara) ci doar inainte de executia primei instructiuni.

6.8.3.2.10 Comanda L (list)

Forme posibile:

(1) **L(CR)**
(2) **Ladr1(CR)**
(3) **Ladr1,adr2(CR)**

Aceasta comanda permite listarea textului sursa (in limbaj de asamblare) asociat continutului unei zone de memorie.

Formele (1) si (2) ale comenzii L permit listarea a 12 linii de text-sursa asociat zonei de memorie care incepe de la "adresa curenta de dezasamblare" (forma 1) sau de la "adr1" (forma 2). Prin forma (3) se listeaza textul-sursa asociat continutului zonei de memorie cuprinsa intre "adr1" si "adr2". Dupa executia comenzii L (orice forma), "adresa de dezasamblare" devine egala cu adresa primei locatii de memorie nedezasamblata; astfel "adresa de dezasamblare" este pregatita pentru executia ulterioara a altor comenzi L. Dupa intilnirea unui "punct de intrerupere", in timpul executiei unui program, "adresa de dezasamblare" devine egala cu valoarea curenta a registrului PC (vezi comenzile G si T).

Dupa lansarea in executie a programului ZSID, "adresa de dezasamblare" este initial stabilita la #100. Comanda L poate fi intrerupta in timpul executiei prin introducerea caracterului RUBOUT (DEL).

6.8.3.2.11 Comanda A (assembly)

Forma generala:

Aadr(CR)

Comanda permite introducerea in memorie, incepind de la adresa "adr", a unor instructiuni Z80. Instructiunile se introduc in format sursa (in limbaj de asamblare), ele continind mnemonice Z80 si operanzi: nume de registre sau constante numerice (valori absolute, exprimate implicit in hexa).

Comanda A realizeaza automat asamblarea instructiunilor introduse si inscrierea codului-masina generat in memorie. Tot ea afiseaza permanent la consola adresa de memorie la care va fi introdusa noua instructiune, specificata de utilizator.

Comanda A se termina atunci cind s-a introdus de la consola o linie vida. Dupa executia ei, utilizatorul poate revedea continutul zonei de memorie modificata, prin utilizarea comenzii L (pentru dezasamblare).
Exemplu:

-A109 (CR)
0109 JP C,10D(CR)

```
010C INC HL(CR)
010D (CR)
-
```

Nota: Comenzile A si L utilizeaza doua module ale programului ZSID (un asamblor si un dezasamblor) care se gasesc dispuse in memorie deasupra nucleului programului ZSID (la adrese mai mici decit acesta!). Zona de memorie asociata acestor module poate fi, in unele cazuri, reacoperita de catre programul supus testarii/depanarii (pentru programe de dimensiuni mari), fara ca utilizatorul sa fie avertizat printr-un mesaj la consola asupra faptului ca modulele ZSID respective nu mai sint disponibile. In aceasta situatie, utilizarea comenzilor L si A va produce aparitia la consola a caracterului "?", iar comenzile T si X nu vor mai contine textul-sursa asociat instructiunii cod-masina respective, ci chiar codul-masina respectiv (in format hexa).

6.8.3.2.12 Comanda I (input)

Forma generala:

Ispecificator-individual(CR)

Daca extensia fisierului specificat in comanda I este "HEX" sau "COM" atunci, in urma comenzii I, pot fi folosite comenzi de tip "R" (read) pentru incarcarea in memorie a programelor hexa sau absolute respective.

Comanda permite utilizatorului sa introduca un specificator-fisier (un nume de fisier) in "blocul de control fisier" (FCB-ul) rezervat standard de CP/M la adresa #005C (vezi si sectiunea de interfata CP/M). Acest FCB poate fi utilizat de catre programul supus testarii/depanarii, el fiind initializat, prin comanda I, asa cum l-ar fi initializat componenta CCP. Specificatorul de fisier, prezent in FCB, este folosit de asemenea si de catre programul ZSID, pentru citirea ulterioara a unor fisiere "HEX" sau "COM".

6.8.3.2.13 Comanda R (read)

Forme posibile:

(1) **R(CR)**
(2) **Radr(CR)**

Comanda R se utilizeaza impreuna cu comanda I (necesita o comanda I prealabila) si permite citirea de pe disc si incarcarea in memorie, in zona TPA, a fisierelor ".HEX" si ".COM" (fisiere ce contin un program de testat/depanat).

Incarcarea programelor in memorie se face la o adresa egala cu "adresa de incarcare a fiecarui bloc", (specificata in fiecare inregistrare dintr-un fisier de tip ".HEX") sau incepind de la adresa implicita #100 (pentru fisierele de tip ".COM"). Operatia de incarcare a programelor in memorie nu trebuie sa afecteze zona #0000-#00FF (prima pagina de memorie), rezervata pentru parametrii sistemului CP/M (vezi si sectiunile de interfata CP/M). Forma (2) a comenzii R face ca programul sa fie incarcat la adresa specificata pe suport la care se aduna "adr" (pentru fisierele tip ".HEX") sau incepind de la adresa #0100 + "adr" (pentru fisierele de tip ".COM"). Parametrul "adr" reprezinta,

deci, o valoare de "deplasare" pentru incarcarea in memorie a programelor. Forma (1) a comenzii R este similara formei (2), in acest caz "adr" fiind implicit considerata #0000.

Utilizatorul poate folosi, in urma unei comenzi I, o singura comanda R pentru incarcarea in memorie a programului de testat/depanat. Daca se doreste reincarcarea programului, trebuie repetata comanda I.

Orice fisier cu extensie "COM" se presupune a fi un program cod-obiect direct executabil (rezultat in urma unei comenzi LOAD sau SAVE) iar fisierul cu extensia "HEX" se presupune a fi un program cod-obiect in format hexa (rezultat, de exemplu, in urma unei comenzi ASM).

Comanda de apel ZSID de forma:

ZSID specificator-individual(CR)

este echivalenta comenzilor:

ZSID(CR)
Ispecificator-individual(CR)
R(CR)

Ori de cite ori se utilizeaza comanda R, programul ZSID raspunde cu:

a. mesajul:

NEXT PC
nnnn pppp

unde:

nnnn - adresa primei locatii de memorie disponibile dupa ce programul a fost incarcat;

pppp - valoarea curenta a PC-ului (#100 pentru fisierele tip ".COM" sau "adresa de lansare automata in executie a programului", preluata din ultima inregistrare a unui fisier tip ".HEX").

Acest mesaj indica faptul ca programul a fost incarcat corect.

b. mesajul de eroare:

?

care indica faptul ca fisierul specificat nu poate fi deschis sau ca a aparut eroare de "cifra de control" in citirea unui fisier de tip ".HEX".

6.8.4 Macroasamblorul de programe relocabile M80

6.8.4.1 Utilizarea macroasamblorului M80

Macroasamblorul M80 traduce programe sursa scrise in limbaj de asamblare 8080 sau Z80 intr-un format obiect relocabil, ce urmeaza a fi prelucrat de link-editorul L80.

Limbajul sursa acceptat contine un numar relativ mare de directive de asamblare, printre care directive pentru definire de macroinstructiuni, directive de asamblare conditionata, directive de control a listei de asamblare si altele.

Comanda de asamblare:

Sint posibile doua moduri de utilizare a asamblorului M80:

- pentru asamblarea unui singur program (modul) sursa se

introduce o singura linie de comanda, de forma:

M80 frel,fprn=fmac/opt1/opt2/...<CR>

- pentru asamblarea mai multor programe (module) sursa se incarca asamblorul prin comanda:

M80<CR>

si apoi se pot introduce mai multe linii de forma:

frel,fprn=fmac/opt1/opt2/...<CR>

ca raspuns la caracterul "*" ce arata ca asamblorul este gata sa primeasca o noua comanda.

Parametrii liniei de comanda au urmatoarea semnificatie:

frel - fisier obiect relocabil produs de asamblor (are tipul implicit .REL);
fprn - fisier de listare produs de asamblor (are tipul implicit .PRN, pe disc);
fmac - fisier sursa citit de asamblor (are tipul implicit .MAC).

Numele de fisiere fmac, frel, fprn urmeaza formatul general :

d:nume.tip **sau** **nume**

Dispozitivul suport "d" poate lipsi cind se refera la discul implicit, iar tipul fisierelor poate lipsi daca se respecta conventiile implicite. Discul suport al fisierului de intrare este preluat implicit si pentru fisierele de iesire.

Optiunile de asamblare "opt" sint:

/C - genereaza referinte incrucisate pe lista de asamblare;
/L - produce fisier de listare;
/H - afisarea in hexa a memoriei pe lista de asamblare;
/O - afisarea in octal a memoriei pe lista de asamblare;
/R - genereaza fisier obiect;
/Z - genereaza cod obiect pentru Z80.

Ca fisier de listare se pot utiliza urmatoarele nume de dispozitive periferice:

TTY: pentru consola sistem;
LST: pentru imprimanta.

Fisierele frel si fprn sint optionale; de asemenea tipurile fisierelor sursa si obiect pot fi omise in comanda (daca fisierele sursa sint de tip .MAC).

La consola se afiseaza intotdeauna liniile cu erori , chiar daca nu se cere lista de asamblare.

Exemple de comenzi de asamblare:

1) **M80 B:TESTM,TTY:=B:TESTM<CR>**

Se asambleaza programul din fisierul B:TESTM.MAC, se genereaza fisierul B:TESTM.REL si se listeaza la consola codul sursa.

2) **M80 =TEST.ASM<CR>**

Se face o verificare sintactica a programului din fisierul TEST.ASM, fara a genera fisier obiect (erorile sint afisate).

3) **M80 =B:TEST/R/L<CR>**

Se asambleaza fisierul B:TEST.MAC si se produc doua fisiere pe acelasi disc: TEST.REL si TEST.PRN. Aceasta comanda este echivalenta cu comanda:

M80 B:TEST,B:TEST=B:TEST<CR>

sau, mai complet:

M80 B:TEST.REL,B:TEST.PRN=B:TEST.MAC<CR>

4) **M80<CR>**
***=P1/R<CR>**
***=P2/R<CR>**
***CTRL/C**

Se asambleaza succesiv fisierele P1.MAC si P2.MAC, generind fisierele P1.REL si P2.REL. Iesirea din asamblor se face cu CTRL/C.

6.8.4.2 Limbajul sursa acceptat de macroasamblorul M80

Limbajul de asamblare acceptat de M80 este compatibil cu limbajele acceptate de alte asambloare pentru 8080, dar prezinta si unele particularitati ce sint prezentate in continuare.

Sintaxa programelor M80:

- O linie sursa poate avea maxim 132 de caractere si, in fata, un numar de linie, cu conditia ca fiecare cifra din acest numar sa aiba bitul 7 setat.
- Literele mici sint acceptate numai in comentarii si in constante alfanumerice.
- In cadrul etichetelor simbolice sint admise urmatoarele caractere speciale, asimilate cu simbolurile: $, ., ?, @, ←.
- Sint admise constante sau expresii in zona de cod mnemonic, ele fiind considerate ca operanzi ai unei directive DB implicite, spre exemplu:

```
zero:    0
```

este echivalenta cu:

```
zero:    DB   0
```

- O eticheta urmata de doua caractere ":" este tratata ca simbol public (global):

```
SUBR::   PUSH     H
```

este echivalenta cu:

```
         PUBLIC   SUBR
         ...
SUBR:    PUSH     H
```

- Un simbol urmat de doua caractere "#" este tratat ca simbol

extern:

```
CALL    MUL##
```

este echivalent cu:

```
EXTRN   MUL
  ...
CALL    MUL
```

- Constantele numerice pot fi:
  - binare (sufix B);
  - octale (sufix O sau Q);
  - hexazecimale (sufix H sau prefix X);
  - zecimale (sufix D sau nici un sufix).

- Constantele numerice prea mari sint automat trunchiate la un cuvint (16 biti).
- Sirurile de caractere pot fi delimitate prin ghilimele simple sau duble.
- Expresiile din zona operand pot contine:
  - operatorii aritmetici +, -, *, / si semnul _;
  - operatorii simbolici NUL, LOW, HIGH, MOD, SHR, SHL;
  - operatorii de relatie EQ, NE, LT, LE, GT, GE;
  - operatorii logici NOT, AND, OR, XOR;
  - paranteze rotunde.

- Operatorii simbolici, de relatie si logici trebuie separati de operanzi prin spatii.
- Se pot utiliza ca operanzi de un octet coduri simbolice de instructiuni:

```
MVI A,(JMP)
MVI C,MOV A,B
```

- Simbolurile sint de 4 tipuri:
  - absolute;
  - relative la segmentul de date;
  - relative la segmentul de program;
  - relative la blocul de comun;

  Expresiile permise cu aceste simboluri sint:

```
<oarecare> + <absolut>  -> <oarecare>
<oarecare> - <absolut>  -> <oarecare>
<oarecare> - <oarecare> -> <absolut>
```

- Simbolurile externe pot fi folosite in expresii aritmetice cu operatorii +, -.

6.8.4.3 Directivele macroasamblorului M80

6.8.4.3.1 Directive generale

```
ASEG
```

Inceput de segment absolut. Acesta directiva este echivalenta cu directiva CSEG insotita de optiunea /P la link-editare (vezi sectiunea 6.8.5.*).

```
COMMON  //
```

```
COMMON    /nume/
```

Inceput de bloc de date comun.

```
CSEG
```

Inceputul unui segment de cod. Directiva CSEG este implicita pentru orice programe, deci poate lipsi in general. Ea este necesara atunci cind segmentul de cod este precedat de un bloc comun, pentru delimitarea celor doua segmente.

```
DB   exp[,exp...]
DB   "sir"[,"sir"...]
```

Genereaza una sau mai multe constante de un octet, corespunzatoare expresiilor sau sirurilor de caractere din zona operand.

```
DC   "sir"
```

Genereaza un sir de caractere ASCII si pune 1 in bitul cel mai semnificativ (bitul 7) din ultimul caracter (pentru detectarea sfirsitului sirului de caractere).

```
DS   exp
```

Rezerva o zona de memorie de lungime egala cu valoarea expresiei "exp".

```
DSEG
```

Inceput de segment de date.

```
DW   exp[,exp...]
```

Genereaza una sau mai multe constante de un cuvint.

```
END  [exp]
```

Marcheaza sfirsitul unui program si, eventual, indica adresa de lansare in executie.

```
ENTRY   nume [,nume...]
PUBLIC  nume  [,nume...]
```

Aceste doua directive echivalente declara unul sau mai multe simboluri publice (puncte de intrare), care pot fi folosite si din alte module asamblate separat.

```
nume    EQU  exp
```

Atribuie numelui din zona eticheta valoarea expresiei din zona operand.

```
EXTRN nume[,nume...]
EXT nume [,nume...]
```

Aceste doua directive echivalente declara unul sau mai multe simboluri externe, definite in alte module dar folosite in modulul curent.

```
NAME      ('nume')
```

Atribuie un nume modulului curent (echivalenta cu TITLE).

ORG exp

Stabileste adresa de implantare pentru codul sau zonele de date ce urmeaza.

PAGE [exp]

Produce trecerea la o pagina noua in cursul listei de asamblare. O pagina are implicit lungimea de 50 de linii sau lungimea data de operandul directivei (10-255).

nume SET exp

Ca si directiva EQU, atribuie numelui din stinga valoarea expresiei din dreapta, dar permite redefinirea unui nume de mai multe ori in program.

SUBTTL "text"

Genereaza un subtitlu, de maxim 60 de caractere, imprimat automat pe fiecare pagina de listare.

TITLE "text"

Genereaza un titlu, imprimat pe prima linie din fiecare pagina a listei de asamblare; primele 6 caractere se considera drept nume al modulului obiect generat.

.COMMEN /text/

Reprezinta o alta posibilitate de a scrie comentarii in textul sursa; ca delimitatori de inceput si de sfirsit comentariu se poate folosi orice caracter (nu neaparat caracterul "/").

.PRINTX /text/

Afiseaza la consola textul respectiv in momentul asamblarii acestei directive. Textul poate fi incadrat de orice caractere identice.

.RADIX exp

Modifica baza de numeratie implicita pentru constantele numerice.

.REQUEST fisier[,fisier...]

Cere linkeditorului sa caute in bibliotecile indicate pentru rezolvarea referintelor externe intilnite in program.

.XLIST / .LIST

Opreste/reia listarea textului sursa in fisierul de listare.

6.8.4.3.2 Directive si conventii pentru definirea de macroinstructiuni

nume MACRO arg1,arg2,...

Inceput de macroinstructiune.

ENDM

Sfirsit de macroinstructiune.

EXITM

Termina expandarea unei instructiuni.

LOCAL arg1,arg2,...

Creaza cite un simbol unic pentru fiecare dintre argumentele formale continute in cursul expandarii unei macroinstructiuni. Etichetele astfel generate au forma **..nnnn** unde "n" este o cifra zecimala.

.LALL

Listeaza tot textul macro la toate expandarile.

.SALL

Listeaza numai codul obiect produs la o macroinstructiune.

.XALL

Interzice listarea macro-expandarilor.

Caracterul "&" se foloseste pentru concatenare de texte sau simboluri in cursul expandarii unei macroinstructiuni. Un argument formal introdus intre "" nu este substituit decit daca este precedat de "&".

Caracterul "!" utilizat inaintea unui caracter intr-un argument face ca acest caracter sa fie tratat ca un literal (sa fie copiat intocmai si nu substituit).

Parantezele ascutite ("< >") se folosesc tot pentru literali in cadrul argumentelor. Notatiile "!;" si "<;>" sint echivalente.

Comentariile dintr-o macro-definitie care sint precedate de ";;" nu sint memorate si deci nu mai apar la expandarea macroinstructiunii.

6.8.4.3.3 Directive de asamblare conditionata

IF NUL <ARGUMENT>

are ca rezultat fals daca, in timpul expandarii, primul caracter din argument este orice altceva decit ";" sau <CR>.

IF NUL arg

are un rezultat fals daca, in timpul expandarii, primul caracter din argument este orice altceva decit ";" sau <CR>.

Formatul general al unei asamblari conditionate este urmatorul:

IFxx arg
...
[ELSE

```
        ...]
    ENDIF
```

unde IFxx poate fi una dintre urmatoarele directive:

IF/IFT exp - adevarat daca <exp> este diferita de zero;
IFE/IFF exp - adevarat daca expresia <exp> este zero;
IF1 - adevarat pentru prima trecere a asamblarii;
IF2 - adevarat pentru a doua trecere a asamblarii;
IFNDEF simbol - adevarat daca <simbol> este nedefinit;
IFDEF simbol - adevarat daca simbol este definit sau este declarat extern;
IFB arg - adevarat daca argumentul este blanc (" ") (parantezele unghiulare sint necesare);
IFNB arg - adevarat daca argumentul este diferit de blanc.

Este permisa suprapunerea directivelor conditionale pe orice numar de nivele.

Argumentele directivelor conditionale trebuie sa fie cunoscute din prima trecere a asamblarii, deci sa fie definite inainte de a fi utilizate ca argumente.

6.8.4.3.4 Directive de asamblare repetata

```
    REPT    exp
```

Repeta instructiunile care urmeaza pina la o directiva ENDM de cite ori arata valoarea expresiei "exp" (valoare pe 16 biti).

```
    IRP     arg,<arg1,arg2,...>
```

Repeta instructiunile care urmeaza pina la o directiva ENDM de atitea ori cite argumente sint in lista dintre parantezele unghiulare, substituind de fiecare data argumentul formal cu argumentul urmator din lista. Secventa urmatoare:

```
    IRP     X,<1,2,3,4>
    DB      X
    ENDM
```

este echivalenta cu secventa:

```
    X   SET  0
        REPT 4
    X   SET  X+1
    DB  X
    ENDM
```

si genereaza 4 directive DB:

```
    DB  1
    DB  2
    DB  3
    DB  4
```

```
    IRPC    arg,"sir"
```

Repeta instructiunile care urmeaza pina la o directiva ENDM de atitea ori cite caractere contine sirul dat, inlocuind de fiecare data argumentul formal cu caracterul urmator din sir.

6.8.4.3.5 Conventii utilizate in lista de asamblare

Formatul unei linii din lista de asamblare este urmatorul:

[crf] [err] loc m xx xxxx linia sursa

unde:

crf - numar de referinta incrucisata (optional);
err - cod de eroare (o litera);
loc - adresa locatiei de memorie;
m - indicator mod de adresare;
xx... - continutul locatiei de memorie.

Indicatorul de mod "m" poate avea valorile urmatoare:

blank - adresa absoluta;
, - adresa relativa la segmentul de cod;
" - adresa relativa la segmentul de date;
! - adresa relativa in blocul comun;
***** - referinta externa.

In tabela de simboluri se folosesc urmatoarele notatii:

***** - simbol extern;
i - simbol public (punct de intrare);
c - nume bloc comun;
u - simbol nedefinit.

6.8.4.4 Coduri de eroare la asamblare

A (ARGUMENT ERROR)	- argument de directiva gresit;
C (CONDITIONAL NESTING ERROR)	- directive conditionale gresit suprapuse;
D (DOUBLE DEFINED SYMBOL)	- simbol dublu definit;
E (EXTERNAL ERROR)	- eroare la o referinta externa;
M (MULTIPLE DEFINED SYMBOL)	- simbol multiplu definit;
N (NUMBER ERROR)	- constanta numerica gresita;
O (OPCOD ERROR)	- cod numeric gresit sau alta eroare de sintaxa;
P (PHASE ERROR)	- eroare de faza la asamblare;
Q (QUESTIONABLE SYNTAX)	- linie terminata incorect;
R (RELOCATION ERROR)	- utilizare incorecta a unui simbol relocabil intr-o expresie;
U (UNDEFINED SYMBOL)	- simbol nedefinit;
V (VALUE ERROR)	- valoare gresita a unei constante.

6.8.4.5 Utilizarea subprogramelor din biblioteca FORLIB.REL

Biblioteca FORLIB.REL contine subprograme utilizate in mod normal de programele rezultate din compilari FORTRAN. O serie de subprograme din biblioteca sint de utilitate mai generala si pot fi apelate de programe scrise in limbaj de asamblare; aceste subprograme pot fi clasificate in trei categorii:

a) subrutine pentru operatii aritmetice;
b) subrutine pentru conversii de tip;
c) functii FORTRAN intrinseci (functii standard).

Conventiile de transmitere a argumentelor la aceste subprograme sint diferite de conventia de transmitere la apelarea de subprograme FORTRAN.

Subrutine pentru operatii aritmetice:

$AC = acumulator de virgula mobila in precizie simpla
($AC + 3 = adresa exponent);
$DAC = acumulator de virgula mobila in precizie dubla
($AC + 7 = adresa exponent).

Conventii

a) argumentul 1 in registre:
- intregi in HL;
- reali in $AC;
- dubla-precizie in $DAC.

b) argumentul 2 in registre sau in memorie, functie de tip:
- intregi in HL sau DE, daca HL contin primul argument;
- reali si dubla precizie in memorie, la adresa data de HL.

Functie	Nume	Arg.1	Arg.2
Adunare	$AA	R ($AC)	I (H,L)
	$AB	R ($AC)	R((H,L))
	$AQ	D ($DAC)	I (H,L)
	$AR	D ($DAC)	R((H,L))
	$AU	D ($DAC)	D((H,L))
Scadere	$SA	R ($AC)	I (H,L)
	$SB	R ($AC)	R((H,L))
	$SQ	D ($DAC)	I (H,L)
	$SR	D ($DAC)	R((H,L))
	$SU	D ($DAC)	D((H,L))
Inmultire	$M9	I (H,L)	I (D,E)
	$MA	R ($AC)	I (H,L)
	$MB	R ($AC)	R((H,L))
	$MQ	D ($DAC)	I (H,L)
	$MR	D ($DAC)	R((H,L))
	$MU	D ($DAC)	D((H,L))
Impartire	$D9	I (H,L)	I (D,E)
	$DA	R ($AC)	I (H,L)
	$DB	R ($AC)	R((H,L))
	$DQ	D ($DAC)	I (H,L)
	$DR	D ($DAC)	R((H,L))
	$DU	D ($DAC)	D((H,L))
Exponentiere	$ED	I (H,L)	I (D,E)
	$EA	R ($AC)	I (H,L)
	$EB	R ($AC)	R((H,L))
	$EQ	D ($DAC)	I (H,L)
	$ER	D ($DAC)	R((H,L))
	$EU	D ($DAC)	D((H,L))

Observatie: La operatia de impartire intreaga ($D9) registrele HL trebuie sa contina impartitorul, iar registrele DE trebuie sa contina deimpartitul. Dupa impartire, registrele HL contin citul intreg al impartirii, iar registrele DE contin restul impartirii intregi.

Subrutine pentru conversii de tip

Argumente:

- logic in A;
- intreg in HL;
- real in $AC;
- dubla-precizie in $DAC.

Functiile de conversie sint date in urmatorul tabel:

Nume	Argument	Rezultat	Tip conversie
$CA	I (H,L)	R ($AC)	intreg-real
$CC	I (H,L)	D ($DAC)	intreg-dubla-precizie
$CH	R ($AC)	I (H,L)	real-intreg
$CJ	R ($AC)	L (A)	real-logic
$CK	R ($AC)	D ($DAC)	real-dubla precizie
$CX	D ($DAC)	I (H,L)	dubla precizie-intreg
$CY	D ($DAC)	R ($AC)	dubla precizie-real
$CZ	D ($DAC)	L (A)	dubla precizie-logic

In tabele s-au folosit urmatoarele notatii:

I = intreg;
R = real;
D = dubla precizie;
L = logic.

Functii FORTRAN intrinseci (functii standard)

Parametrii din HL si DE (al treilea argument in BC) reprezinta adresele argumentelor.

Pentru MIN si MAX numarul de argumente se transmite in A.

Cu exceptia functiilor INP si OUT, nici o functie nu admite argumente de un octet. Ele trebuie convertite, altfel apar rezultate impredictibile.

6.8.5 Editorul de legaturi L80

Programul L80 este un editor de legaturi pentru module obiect relocabile in format .REL, rezultate din compilare FORTRAN (cu F80), din asamblare cu asamblorul M80, din compilare BASIC (cu BASCOM) sau din compilare COBOL.

Linkeditorul L80 poate lega impreuna mai multe module obiect aflate in fisiere de tip .REL cu module extrase din una sau mai multe biblioteci de subrutine relocabile, producind un singur program executabil care se incarca in memorie si care, la cerere, este scris intr-un fisier de tip .COM.

Bibliotecile prelucrate de L80 trebuie create cu un program bibliotecar special numit LIB80.

Listarea numelor modulelor dintr-o biblioteca se poate face cu bibliotecarul sau chiar cu linkeditorul L80, daca se leaga intre ele toate modulele din biblioteca si se foloseste optiunea

/M de afisare a simbolurilor globale.

Linkeditorul L80 cauta implicit intr-o biblioteca cu numele FORLIB.REL (necesara pentru modulele generate de compilatorul FORTRAN), dar se poate cere explicit cautarea in alte biblioteci pentru satisfacerea referintelor externe (optiunea /S).

Nu este prevazuta crearea de programe segmentate cu ajutorul linkeditorului L80.

6.8.5.1 Sintaxa comenzii de linkeditare

Prin comanda de lansare a linkeditorului se incarca in memorie programul L80 si, eventual, se transmit numele fisierelor de intrare si optiunile. Este posibil ca numele fisierelor si/sau optiunile sa fie introduse pe rind, dupa incarcarea sa in memorie.

Sfirsitul introducerii de module obiect si de optiuni este marcat prin optiunea /E (Exit) sau /G (Go), moment in care linkeditorul cauta in biblioteca implicita si/sau in bibliotecile date explicit, pentru a rezolva referintele externe ramase si pentru a incheia editarea.

In lipsa optiunii /N, linkeditorul incarca in memorie programul creat si nu creaza un fisier disc cu acest program.

Comanda de linkeditare poate avea una din formele:

```
L80 frel1/opt,frel2/opt,.../E<CR>
```

sau:

```
L80 frel1/opt,frel2/opt,...,fcom/N/E<CR>
```

cind se doreste obtinerea unui fisier obiect de tip .COM, sau:

```
L80<CR>
*frel1/opt<CR>
*frel2/opt<CR>
*/E<CR>
```

sau:

```
L80<CR>
*frel1/opt<CR>
*frelD/opt<CR>
    ...
*fcom/N
*E
```

cind se doreste obtinerea unui fisier "fcom" de tip .COM

Optiunile "opt" pot avea una din valorile urmatoare:

```
/D:<adresa>
```

Stabilire adresa de incarcare a segmentelor de date si de comun; adresa implicita a segmentului de date este 103H.

```
/E
/E:nume
```

Terminare linkeditare, eventual cu indicarea numelui simbolului global ce reprezinta adresa de lansare.

```
/G
/G:nume
```

Lansare automata in executie dupa linkeditare, eventual cu indi-

carea numelui adresei de lansare (simbol public).

/M

Listare inceput si sfirsit segment de date si de program, precum si numele simbolurilor globale (publice si externe).

/P:adresa

Stabilire adresa de incarcare a segmentului de cod din urmatorul modul obiect prelucrat (nu are efect asupra modulului curent). Adresa implicita a segmentului de cod este 100H.

/R

Relansarea de la inceput a linkeditorului, fara reincarcarea sa in memorie (se foloseste dupa o eroare la numele fisierelor de intrare).

fisier/S

Se cere linkeditorului sa caute in biblioteca indicata prin <fisier>, pentru rezolvarea referintelor externe.

/U

Listare inceput si sfirsit segment de date si de program, precum si numele simbolurilor globale nedefinite.

Observatii:

- Optiunile /D si /P pot lipsi; in acest caz linkeditorul plaseaza segmentul de date inaintea segmentului de cod la fiecare modul si introduce o instructiune de salt (JMP) peste segmentul de date. Daca lipseste segmentul de date, atunci segmentul de program urmeaza imediat instructiunii JMP ;
- Adresa de incarcare implicita este 100H;
- Daca se folosesc directive ORG intr-un program, atunci adresele continute in aceste directive sint tratate ca adrese relative la baza de incarcare implicita (100H) si nu ca adrese absolute;
- Adresa primei locatii libere dupa segmentul de date (sau dupa segmentul de cod daca lipseste segmentul de date) este introdusa de linkeditor in simbolul global $MEMRY, daca a fost definit un asemenea simbol in program.

6.8.5.2 Exemple de comenzi de linkeditare

1) L80 TEST,TEST/N/E<CR>

Se face relocarea programului obiect citit din fisierul TEST.REL, se creaza un fisier TEST.COM si se revine in sistem.

2) L80 TEST/G<CR>

Se face relocarea programului obiect citit din fisierul TEST.REL, se incarca programul absolut in memorie si se lanseaza in executie.

3) L80 G1,G2<CR>
   *GLIB/S<CR>

```
   *GRAF/N/E<CR>
```

Se leaga impreuna modulele citite din fisierele G1.REL si G2.REL cu subrutinele apelate din aceste module si extrase din biblioteca GLIB; se creaza fisierul GRAF.COM.

```
4) L80<CR>
   *G1<CR>
   *G2<CR>
   *GLIB/S<CR>
   *GRAF/N<CR>
   */E<CR>
```

Acelasi efect cu secventa din exemplul precedent.

6.8.6 Bibliotecarul LIB80

Programul bibliotecar LIB80 realizeaza urmatoarele functii:

- crearea de biblioteci relocabile pornind dela fisiere rezultate din compilari si asamblari diferite;
- listarea modulelor dintr-o biblioteca impreuna cu toate numele de simboluri globale;
- extragerea de module obiect din fisiere sau biblioteci in alte fisiere relocabile.

Apelarea bibliotecarului se fac prin comanda:

```
LIB80
```

Dupa incarcarea sa in memorie LIB80 afiseaza un asterisc (*) si asteapta comenzi. Formele unei comenzi LIB80 pot fi:

```
     *flib=frel1,frel2,../opt
sau:
     *flib/opt
sau:
     *flib=frel1
     *frel2
     *frel3
     ...
     */E
```

Numele fisierelor de intrare se pot da fie in aceeasi linie, separate prin virgule, fie pe linii separate.

Notatii folosite:

flib - este numele fisierului biblioteca creat (poate lipsi);
frel - este numele unui fisier de intrare de tip **.REL**;
opt - sint optiuni pentru bibliotecar.

Un fisier relocabil poate contine unul sau mai multe module obiect. Un modul obiect corespunde unei unitati de program Fortran (program principal, subrutina sau functie) sau unei secvente in limbaj de asamblare ce contine directive ENTRY si eventual alte directive pentru stabilire nume modul (NAME, TITLE).

Bibliotecarul LIB80 concateneaza module obiect si nu fisiere relocabile.

Pentru un fisier relocabil care contine mai multe module obiect se poate preciza numele modulului sau modulelor care se

extrag din fisierul respectiv. Daca nu se indica in mod explicit care module se extrag, atunci se considera implicit toate modulele din fisierul respectiv.

6.8.6.1 Specificarea modulelor fisierelor

Specificarea unui singur modul "MOD" din fisierul "FILE":

FILE<MOD> sau **FILE<MOD+n>** sau **FILE<MOD-n>**

modulul MOD sau modulul "n" dupa sau inainte de MOD.
Specificarea prin nume a fiecarui modul extras:

FILE<MOD1,MOD2,MOD3>

modulele MOD1, MOD2, MOD3.
Specificarea unei liste de module adiacente:

FILE<MOD1..MOD5>

toate modulele intre MOD1 si MOD5 inclusiv MOD1, MOD5;

FILE<..MOD5>

toate modulele dela inceputul lui FILE pina la MOD5, inclusiv;

FILE<MOD1..>

toate modulele care urmeaza lui MOD1 in FILE, inclusiv MOD1.
Daca nu se specifica explicit fisierul de iesire atunci se considera implicit ca fisier de iesire biblioteca FORLIB.REL.

6.8.6.2 Optiuni LIB80

/L Listarea modulelor dintr-un fisier sau dintr-o biblioteca cu punctele de intrare si referintele externe din fiecare modul.
/E Iesire din LIB80 in CP/M. Biblioteca creata primeste tipul .REL deoarece pe parcursul crearii are tipul .LIB.
/R Modifica tipul bibliotecii create din .LIB in .REL; nu se iese.
/U Listare referinte nerezolvate (referinte inapoi) dupa parcurgerea fisierului specificat.
/C Sterge biblioteca in curs de creare; reluare LIB80 de la inceput.

6.8.6.3 Exemple de utilizare LIB80

1)Crearea unei noi biblioteci GLIB.REL pe baza modulelor din fisierele MOVE.REL, DRAW.REL (provenite din Fortran) si DOT.REL (provenit din asamblare). Subrutina DOT este apelata din MOVE si din DRAW; DRAW apeleaza pe MOVE.

```
A>LIB80
*GLIB=DRAW,MOVE,DOT
*/E
```

2)Adaugarea modulelor din fisierul DASH.REL la biblioteca GLIB:

```
A>LIB80
*TEMP=GLIB,DASH
*TEMP/R
*GLIB=TEMP/E
```

3)Verificarea ordinii corecte a modulelor in biblioteca GLIB:

```
A>LIB80
*GLIB/U
*GLIB/L/E
```

4)Extragerea modulelor MOVA si DRAWA din biblioteca GLIB in fisierul GRAF.REL:

```
A>LIB80
*GRAF=GLIB<MOVEA,DRAWA>/E
```

Observatii:

Extragerea unor module obiect dintr-o biblioteca sau dintr-un fisier nu sterge aceste module din fisier; nu se poate face o stergere selectiva de module dintr-un fisier relocabil.
Daca exista anterior pe disc un fisier cu numele bibliotecii create, atunci se sterge acest fisier fara nici o avertizare!
Adaugarea de noi module se face intotdeauna la sfirsitul fisierului destinatie; nu se pot intercala noi module intre alte module dintr-o biblioteca.
Nu se poate da o comanda de forma:

```
GLIB=GLIB,DASH
```

cu intentia de adaugare a fisierului DASH.REL la GLIB.REL.
Nu se recomanda modificarea bibliotecii standard FORLIB.REL.

6.9 KERMIT

6.9.1 Generalitati

KERMIT este un utilitar care permite conectarea a doua calculatoare, in scopul transferului bidirectional de informatie, prin porturile lor de terminal (TTY), facind ca unul (sau amindoua) sa considere celalalt calculator ca fiind un terminal propriu. Suportul hardware al unei asemenea conectari il constituie interfata serie RS-232; software, coduri de caractere ASCII). O data cele doua calculatoare conectate in acest mod, pe fiecare se pot executa programe cooperante care sa asigure serviciile de comunicatie dorite, pe baza unui anumit protocol de comunicatie.

Dar, in fond, de ce este necesar un protocol? Pentru a raspunde la aceasta intrebare sa observam ca la conectarea a doua calculatoare prin linii TTY apar citeva probleme majore:

1.**Zgomot**; nu este recomandabil sa se presupuna ca nu exista interferente electrice pe o linie de comunicatie; orice linie de comunicatie inchiriata sau comutata va avea, aleator, interferente sau zgomote, care determina caractere eronate, lipsa sau false. Zgomotul poate corupe informatia din date in moduri foarte variate, de regula imprevizibile, fapt care nu poate fi depistat decit atunci cind este prea tirziu.

2.**Sincronizare**; datele nu trebuie sa apara la viteze mai mari decit cele la care receptorul poate sa raspunda corect. Desi viteza de comunicatie pe linie la cele doua capete ale conexiunii poate sa fie identica, totusi masina receptoare s-ar putea sa nu fie capabila sa prelucreze un flux de date pe intrare la acea viteza. Unitatea sa centrala poate fi prea lenta sau prea incarcata cu alte sarcini prioritare, sau buffer-ele acesteia pot fi ocupate sau prea mici. Simptomul tipic al unei astfel de probleme de sincronizare este pierderea datelor; majoritatea sistemelor de operare ignora datele de intrare pe care nu sint pregatite sa le primeasca.

3.**Caderi ale liniei**; o linie de comunicatie isi poate opri functionarea pentru perioade scurte datorita unui conector defectabil, caderii de tensiune sau altor motive similare. Pe conexiuni comutate sau cu apel automat astfel de caderi intermitente vor determina caderea purtatoarei si inchiderea deci a conexiunii. Oricum, pentru orice conexiuni in care semnalul de purtatoare nu este utilizat simptomul este acelasi: pierderea datelor.

Pentru a preintimpina corupera datelor si pentru sincronizarea comunicatiei, calculatoarele cooperante pot (de fapt, trebuie) sa schimbe intre ele informatii de control, pe parcursul transferului datelor. Schimbul intretesut de informatii de control si date dupa anumite reguli si actiunile asociate constituie un "protocol" de comunicatie.

KERMIT implementeaza un astfel de protocol. Este proiectat special pentru transferul de fisiere secventiale pe linii de telecomunicatii seriale normale. KERMIT nu este in mod necesar mai bun decit multe alte protocoale de transfer de fisiere orientate spre terminal, dar este implementat compatibil pe o varietate de mini si microcalculatoare, printre care si Tim-S Plus.

KERMIT transfera datele prin incapsularea lor in "pachete" de control a informatiei, cuprinzind: caractere de sincronizare, numar de pachet (care permite detectarea pachetelor pierdute), indicator de lungime si CRC (care permite verificarea integritatii datelor). Pierderea sau coruperea pachetelor este detectata

si se cere retransmiterea acestora. Pachetele duplicate sint ignorate. In plus, exista diferite alte pachete de control speciale care permit entitatilor KERMIT cooperante conectarea si deconectarea uneia fata de cealalta, precum si schimbul de informatii de control si stare. Deoarece in proiectarea protocolului s-au facut foarte putine presupuneri relativ la capabilitatile fiecarui calculator, KERMIT poate lucra pe o gama foarte larga de sisteme.

6.9.2 Cum se utilizeaza KERMIT

KERMIT reprezinta un protocol pentru transferul fiabil de fisiere, intre calculatoare pe linii de telecomunicatii seriale normale. Procedura de utilizare a KERMIT pentru a transfera un fisier presupune mai intii efectuarea anumitor operatii suplimentare pentru utilizare, operatii necesare deoarece KERMIT nu este integrat in nici un sistem particular, ci este "grefat deasupra" pe sistem.

6.9.2.1 Programul KERMIT

KERMIT inglobeaza un set de reguli pentru transferul fiabil de fisiere intre calculatoare. In general, un calculator este un sistem mare (un host, spre exemplu un sistem time-sharing cu multe terminale), iar altul un minicalculator sau un calculator personal (PC). Sint posibile si legaturi Host-la-Host sau PC-PC. In aceste conditii host-ul considera PC-ul un terminal normal. Pentru ca protocolul KERMIT sa lucreze este necesar ca la fiecare capat al liniei de comunicatie sa existe un program KERMIT (unul in host, altul in PC).

Cele doua programe KERMIT schimba mesaje intr-un limbaj special, si anume protocolul KERMIT. Dialogul decurge cam in modul urmator: "Hei! Iti voi trimite fisiere. Cind imi transmiti mesaje te rog sa nu fie mai lungi de 80 caractere, iar daca nu auzi nimic despre mine timp de 15 secunde, cheama-ma din nou, OK?"; "OK."; "Acum urmeaza fisierul numit FOO.TXT, OK?"; "OK."; "Iata prima bucata"; "Am primit-o"; "Bine, iata a doua bucata nou ..."; etc. Desigur, nimic din aceasta conversatie nu se vede la nivelul utilizatorului. Aceste mesaje, extinse literar, sint impachetate in cod concis pe care cele doua KERMIT il pot intelege; entitatile KERMIT se ocupa de transmisie, tratarea si recuperarea erorilor, translatarea setului de caractere si asa mai departe. Fiecare mesaj este numit pachet si fiecare pachet este intr-un format special pe care oricare KERMIT il poate intelege.

6.9.2.2 Conversind cu doua calculatoare deodata

Sarcina utilizatorului este de a lansa cele doua programe KERMIT, de la acelasi calculator. Confuzia apare deoarece trebuie utilizat un singur display si keyboard pentru a conversa cu cele doua calculatoare, mai precis cu cele doua programe KERMIT. Sa presupunem un caz concret: utilizatorul sta la un calculator personal (PC), care poseda un port serial de comunicatie (termenii PC, micro, microcalculator si statie de lucru vor fi utilizati pentru a desemna un sistem cu un singur utilizator). Portul serial este conectat la calculatorul host printr-un modem. Semnificatia reala a conexiunii nu este importanta in acest caz, ea poate fi o linie directa la host, o linie inchiriata etc.

In mod normal, cind utilizatorul foloseste PC-ul sau el "discuta" direct cu el, comenzile lui sint interpretate direct de sistemul de operare al PC-ului (CP/M, MS-DOS, UNIX, sau oricare altul), sau de anumite programe care sint executate de PC (un editor, un formater de text sau un joc ...). Versiunea de KERMIT din PC-ul acesta este un program ca oricare altul, dar care are o posibilitate speciala de functionare: fie de a interpreta comenzile utilizator direct, ca alte programe, fie sa transfere orice se introduce de la un terminal direct la host.

Cind se cere lui KERMIT sa execute comanda CONNECT, el trimite fiecare caracter pe care-l tasteaza utilizatorul la portul serial si va pune fiecare caracter care soseste de la portul serial pe ecran. Acesta se numeste **serviciu de terminal virtual:** un calculator se comporta, "virtual", ca si cum ar fi terminalul celuilalt. KERMIT, ca majoritatea programelor interactive are un prompt. Prompt-ul este un simbol afisat la marginea stinga a liniei de terminal, care indica faptul ca el este gata sa primeasca o noua comanda. Prompt-ul de KERMIT este de obicei "KERMIT-XX>". Simbolurile XX identifica implementarea curenta a KERMIT; KERMIT care lucreaza pe microcalculatoare cu microprocesor Z80 sau 8080 se numeste "KERMIT-80" si are prompt-ul "KERMIT-80>"; KERMIT de pe IBM-PC este "KERMIT-86" (desi procesorul in IBM PC este 8088, el este programat ca si cum ar fi 8086), si asa mai departe. Daca utilizatorului ii este la un moment dat neclar cu care din cele doua calculatoare discuta, prompt-ul ii ofera raspunsul (desigur daca sint calculatoare de tipuri diferite). In plus, majoritatea implementarilor KERMIT tiparesc mesaje informative ca:

[Connecting to remote host, type CTRL-\C to return]

in urma comenzii CONNECT, sau mesaje ca:

[Connection closed, back at PC]

cind se revine la calculatorul care a cerut conectarea.

O data "conectarea" la host facuta, trebuie sa existe o modalitate pentru utilizator de a obtine inapoi controlul direct asupra PC-ului. Aceasta se poate realiza printr-o secventa de **escape.** In timp ce KERMIT transfera caracterele utilizator catre host, el le verifica pe fiecare in parte pentru a vedea daca este un caracter definit ca **escape.** Cind il depisteaza, el opreste procesul de ignorare a utilizatorului si acesta este comutat pentru a "comunica" direct cu PC-ul si nu cu host-ul.

Caracterul de **escape** este, in mod normal, ales pentru a fi unul care nu se foloseste in conversatia cu host-ul si unul care este greu de tastat accidental; este de obicei un caracter de control ca Control-\, care se produce prin tastarea simultana a tasei marcata CTRL sau CONTROL si a caracterului indicat (in acest caz, slash invers "\"). Tasta CTRL lucreaza ca si cea SHIFT. Caracterele de control sint afisate ca Ctrl/A sau ^A, unde A este caracterul de tastat impreuna cu CTRL.

6.9.2.3 Transfer de fisiere

Pentru a transfera un fisier, trebuie mai intii sa fie anuntat sistemul de operare al PC-ului. Aceasta se obtine de obicei prin pornirea PC-ului si - eventual - introducerea mai intii a floppy disk-ului sistem. O data ajuns la nivel de comanda la PC, utilizatorul lanseaza KERMIT. Apoi se comanda CONNECT

(catre KERMIT) pentru conectarea la host. In acest moment utilizatorul comunica cu host-ul; desigur, acum trebuie "intrat" (log in) in sistemul host-ului si apoi lansat KERMIT pe host. Dupa aceasta, la fiecare capat al liniei exista activ cite un program KERMIT. Urmatorul pas este sa se anunte fiecare KERMIT de sarcina pe care o are de indeplinit. Sa presupunem ca se doreste sa se transfere un fisier de la host la PC; trebuie in primul rind sa se anunte KERMIT-ul host sa trimita (SEND) fisierul, apoi prin secventa "escape" sa se revina la KERMIT PC si sa se ceara acestuia sa receptioneze fisierul. Transferul incepe, iar utilizatorul poate sta sa urmareasca (dinamic) procesul, sau sa ia o pauza.

KERMIT de la PC va arata continuu pe ecran numarul de pachete transmise si numarul de retransmisii, daca este cazul, si va anunta utilizatorul cind transferul este incheiat.

Fisierul dorit este acum pe discul din PC. Protocolul KERMIT a asigurat ajungerea corecta si completa a fisierului. Acum utilizatorul trebuie sa "curete" contextul: se face CONNECT la loc, pe host, se termina KERMIT de pe host, se termina, de asemenea, sesiunea de lucru de pe host (log out), iar prin secventa "escape" se revine la KERMIT PC. Acum utilizatorul poate face ceea ce si-a propus cu fisierul sau: editare, afisare pe PC, transmitere la alt sau la acelasi host etc.

Protocolul KERMIT, si deci programele KERMIT, permit sa se transfere fisiere in mod fiabil de la host la PC, de la PC la host, de la host la host, sau de la PC la PC, de obicei fara vreo cerinta speciala asupra naturii sistemului de prelucrare implicat. Majoritatea implementarilor permit, de asemenea, transferul de fisiere in grup, cu o singura comanda, ca de exemplu "transfera toate fisierele FORTRAN". Scenariul de lucru pentru toate cazurile anterioare, este in mare acelasi, doar detaliile asupra modului de stabilire a conexiunii difera.

KERMIT lucreaza cel mai bine cu fisiere afisabile, fisiere compuse doar din litere, cifre, semne de punctuatie CR, TAB si altele, deoarece acestea sint interpretate la fel pe aproape oricare tip de calculatoare.

KERMIT poate sa transfere insa si fisiere "binare", ca de exemplu programe executabile, compuse din paternuri arbitrare de biti pe octet, dar aceste fisiere sint semnificative doar pe tipul de calculator pe care a fost generat. Totusi, KERMIT poate sa transfere astfel de fisiere de pe sistemul A pe sistemul B (unde ele nu sint utilizabile) si apoi sa le aduca inapoi pe sistemul A in conditii originale, desi in anumite cazuri trebuie luate anumite masuri de prevedere.

Acum, dupa ce am obtinut citeva cunostinte de baza despre ceea ce este KERMIT si cum functioneaza, sa urmarim niste exemple concrete. Mai intii, insa, utilizatorul trebuie sa stie care sint, in mare, comenzile KERMIT de baza.

6.9.2.4 Comenzi KERMIT de baza

In cele ce urmeaza se vá face o descriere generala a majoritatii comenzilor KERMIT de baza. Descrierile detaliate se vor face mai tirziu. In aceste descrieri termenul "local" se refera la sistemul pe care utilizatorul il foloseste direct, iar termenul "indepartat" se refera la sistemul la care s-a facut CONNECT-area prin KERMIT. Comenzile pot avea unul sau mai multi operanzi pe aceeasi linie si sint terminate prin CR. Vom prezenta urmatoarele comenzi:

SEND spec-fis

transfera un fisier sau un grup de fisiere specifiat prin **spec-fis** de la KERMIT-ul curent (de la care se emite comanda) la celalalt (cu care este conectat). Numele fiecarui fisier este transmis celuilalt KERMIT intr-un pachet de control special, astfel incit el poate fi creat acolo cu acelasi nume. Un grup de fisiere este de obicei specificat prin folosirea de caractere "wildcard", cum ar fi *, in cadrul **spec-fis.**
Exemple:

```
>send foo.txt
>send  *.for.
```

Totusi, anumite implementari KERMIT pot sa nu suporte transferuri de grupuri de fisiere; aceste versiuni necesita deci comenzi SEND separate pentru transferul fiecarui fisier.

RECEIVE

solicita receptia unui fisier sau a unui grup de fisiere de la KERMIT-ul cu care este CONNECT-at. Daca un nume de fisier sosit nu este legal, atunci se incearca sa se transforme intr-un nume legal similar, (ex: prin eliminarea caracterelor ilegale sau excesive). Numele astfel format nu poate fi garantat unic. In acest caz fisierele anterioare existente ar putea fi sterse (in cazul in care sistemul local de fisiere nu are facilitatea de versiune, ca spre exemplu in CP/M).

Anumite versiuni de KERMIT incearca sa preintimpine aceasta prin anuntarea cazurilor de coliziune de nume de fisier.

CONNECT

realizeaza o conexiune de tip "terminal virtual" cu sistemul de la distanta. Pentru un sistem PC sau micro, aceasta inseamna de obicei transmiterea tuturor caracterelor de la intrarea locala (tastatura) prin portul serial de iesire si afisarea pe display a tuturor caracterelor receptionate de la portul serial. Pentru a termina conexiunea de terminal virtual, se testeaza caracterul KERMIT de "escape" (ex. CTRL \) urmat de litera "C" (de la "Close connection": inchidere conexiune).

SET

stabileste diverse caracteristici nestandard ale sesiunii KERMIT, cum ar fi: caracter de escape pentru CONNECT, caracteristici ale fisierelor manipulate, numar de linii de comunicatie, paritate, sau caracteristici ale controlului de flux.

SHOW

Afiseaza valorile curente ale optiunilor SET.

HELP

Afiseaza un rezumat al comenzilor KERMIT.

EXIT

Termina sesiunea de lucru KERMIT si se revine la nivelul sistemului de operare gazda.

?

Acest caracter ("?") tastat oriunde in cadrul unei comenzi KERMIT determina listarea comenzilor, optiunilor sau operanzilor posibile in acel punct. In functie de sistemul de operare gazda, aceasta comanda necesita sau nu un CR.

6.9.2.5 Exemple concrete

Asa cum s-a aratat, KERMIT poate fi utilizat in mai multe feluri: de la un PC conectat la un calculator gazda mai mare; de la un calculator gazda (host) care este conectat la altul; de la un PC la altul.

6.9.2.5.1 De la Micro la Host

In acest exemplu, utilizatorul se afla in fata unui micro Tim-S Plus, care este conectat printr-un port serial la un minicalculator (CORAL sau INDEPENDENT), deci host. Tim-S Plus este local, minicalculatorul este indepartat. Acest exemplu este tipic pentru aproape orice alta implementare KERMIT pe microcalculatoare.

Cu micro pornit si cu programul KERMIT pe disc, utilizatorul lanseaza local KERMIT. Apoi utilizeaza comanda KERMIT de CONNECT pentru a deveni un terminal al minicalculatorului. De fapt, micro-ul emuleaza terminalul popular de tip VT52 (sau echivalent), astfel ca este recomandabil ca utilizatorul sa anunte sistemul gazda (host) ca "terminalul" este de acest tip. Inainte insa, sa va intra in sistem (prin procedura de "log in"), si apoi se va lansa KERMIT pe mini.

In cele ce urmeaza, aceasta procedura este exemplificata concret (comenzile intarite):

```
A>KERMIT                    ! Lanseaza KERMIT pe micro.
KERMIT-80 v4.05 configured for Tim-S/Plus vith VT52
KERMIT-80>                  ! Acesta este prompt-ul KERMIT-
                              ului pentru micro.
KERMIT-80>connect           ! Conecteaza la minicomputer.
[Connected to remote host, type control-\C to return]
                            ! Acum esti conectat la minicompu-
                              ter.
>HEL myuserid/mypassword    ! Login la minicomputer.
>SET  /VT52=TI:             ! Defineste tipul terminalului
                              (optional).
>KER                        ! Lanseaza programul KER la mini-
                              computer.
KER-11>SET LIN TT4:         ! Acesta este KERMIT-ul implicit
                              pe minicomputer.
```

Acum totul este pregatit pentru a se transfera fisiere intre cele doua masini.

Urmatorul exemplu ilustreaza modul de transmitere a fisierelor de la minicalculator la microcalculator. De observat utilizarea caracterului "wildcard" "*", care desemneaza un grup de fisiere.

```
KER-11>send *.for  ! Trimite toate fisierele mele de tip
                     FORTRAN.
^\C                ! Acum intoarce-te la micro prin tipari-
                     rea secventei escape, in acest caz ^\C
```

```
                              (tastele "CTRL" si "C" simultan).
[Back at micro]
 KERMIT-80>receive ! Spune-i lui micro ca fisierele sint pe
                   drum, urmind sa-i soseasca.
```

Textul de la caracterul "!" pina la sfirsitul liniei este comentariu si nu raspuns sistem sau parte de comanda.

Daca intre comanda **send** si cea de revenire la KERMIT-80 si de la lansarea a **receive** se scurg mai mult de aproximativ 5 secunde, primele pachete de la KER-11 pot sa ajunga prematur la KERMIT-80 si vor aparea pe ecranul micro-ului (afisarea acestor pachete in mod caracter pe ecran va fi desigur fara vreo semnificatie), dar din punct de vedere al transferului nu se va intimpla nimic, deoarece pachetul astfel pierdut va fi retransmis de KER-11 pina cind micro il va confirma.

Odata ce conexiunea este stabilita, micro-ul va afisa ceea ce se intimpla: va sterge mai intii ecranul si va astepta sosirea pachetelor; pe masura ce acestea sosesc micro-ul va afisa pe ecran, continuu, numele fisierului curent si numarul de pachete. Cind prompt-ul de micro "KERMIT-80>" reapare pe ecranul micro-ului, transferul este incheiat. In timpul transferului fisierului, ecranul microcalculatorului va arata cam asa:

```
      KERMIT-80 v4.05 [Tim-S/Plus]
Number of Packets:  294
Number of Retries:    2
File Name:   FOO.TXT
               Receiving...
```

Contoarele de numar de pachet curent si de retransmisii sint continuu actualizate, iar cuvintul din partea dreapta sus a ecranului anunta starea transferului: "receiving", "sending", "complete", "interrupted" sau "failed".

Cind transferul este complet (majoritatea versiunilor de KERMIT folosesc si o secventa de "bell", sonora), utilizatorul trebuie sa revina (prin **CONNECT**) la minicalculatorul host, sa iasa (**EXIT**) din KERMIT, sa termine sesiunea de lucru la mini (prin procedura de "log out") si apoi sa revina (prin secventa "escape") la PC, ca si in etapa anterioara (la transfer).

```
KERMIT-80>connect     ! Intoarce-te la mini.
[Connecting to host. Type CTRL-\C to return to micro.]
KER-11>               ! Esti aici.
KER-11>exit           ! Iesi din KER-11.
>BYE                  ! Logout din mini.
HAVE A GOOD MORNING
24-Jan-86 15:18:56
^\C                   ! Acum"escape" te intoarce la micro
[Back at micro]
 KERMIT-80>exit       ! si vei iesi din KERMIT-ul micro.
```

Fisierele transferate trebuie acum sa se gaseasca pe disk-ul micro-ului.

Pentru a transfera fisiere de la micro la mini, se urmeaza o procedura similara. Mai intii se urmeaza instructiunile din sectiunea anterioara, pentru a intra ("log in") in minisistem prin micro. Apoi, ca raspuns la prompt-ul host-ului "KER>", se comanda **receive** in loc de **send.** Apoi se revine la micro si se va da comanda SEND. Micro-ul va afisa evolutia transferului pe ecranul sau.

Cind prompt-ul "KERMIT-80>" indica terminarea transmisiei,

utilizatorul va urma procedura aratata mai sus pentru a iesi de pe host (eventual, inainte de aceasta, utilizatorul poate dori sa-si confirme ca fisierele au fost corect memorate in directorul respectiv de pe mini).

6.9.2.5.2 Micro la Micro

Atunci cind KERMIT functioneaza intre calculatoare personale (microcalculatoare, statii de lucru etc.), cererile de comenzi se dau de la ambele console (micro) si nu de la una singura. Acesta deoarece un calculator personal nu accepta, in mod normal, decit comenzi de la o consola. Daca un KERMIT de PC se CONNECT-eaza la altul, la celalalt capat nu exista nici un program de sistem care sa-l asculte (ca in cazul unui host).

Realizarea unei conexiuni fizice intre doua microcalculatoare (cum ar fi, de pilda, doua PC-uri) - care sint de modele diferite si nu sint compatibile la nivelul discurilor floppy - presupune asigurarea unor cerinte de conectare fizica diferite. De exemplu, unele necesita conector "tata" pe poarta lor seriala, altele de tip "mama"; unele cer ca semnalele RS-232 sa fie 0, altele 1. In anumite cazuri trebuie facute legaturi directe intre pini ai conectorului (spre exemplu: unele micro au nevoie ca DTR (pin 20, Data Terminal Ready) sa fie 1 logic, acesta poate fi realizat prin legarea lui la CTS (pin 5, Clear To Send). Pentru astfel de probleme trebuie cunoscut standardul EIA RS-232-C si oricum, trebuie studiat cu atentie manualul corespunzator al sistemului. In plus, si in primul rind, desigur, trebuie asigurata compatibilitatea de viteze la porturile de la ambele capete ale conexiunii fizice.

Conexiunile la distante mai mari se pot realiza prin apeluri automate (dial up), daca sint disponibile modemuri corespunzatoare (o parte trebuie sa aiba facilitatea de autoraspuns), sau printr-o linie inchiriata sau circuit comutat (sau orice altceva in care se poate intra cu un conector EIA).

In exemplul de fata un microcalculator A (Tim-S Plus sub CP/M) este conectat la un alt micro B (tot Tim-S Plus sub CP/M) utilizind un cablu de modem nul tata-mama. Realizarea acestei conexiuni fizice este partea cea mai dificila a problemei. Conexiunea poate fi testata prin rularea produsului KERMIT si lansarea comenzii CONNECT la fiecare capat: caracterele tastate de la fiecare micro trebuie sa apara pe ecranul celuilalt.

Sa presupunem ca dorim sa transmitem un fisier FOO.HEX de la sistemul A la sistemul B. Pentru aceasta procedam astfel:

1.Se lanseaza KERMIT pe sistemul B si se da comanda **receive:**

```
A>KERMIT
KERMIT-80 v4.05 configured for Tim-S/Plus vith VT52
KERMIT-80>receive _
```

2.Se lanseaza KERMIT pe sistemul A si se da comanda **send** pentru fisierul FOO.HEX:

```
A>KERMIT
KERMIT-80 v4.05 configured for Tim-S/Plus vith VT52
KERMIT-80>send foo.hex
```

In acest moment putem urmari evolutia transferului. Cind obtinem urmatorul prompt KERMIT-80> transferul este incheiat si putem iesi (EXIT) din ambele KERMIT.

Punctul esential consta in lansarea mai intii a capatului (i.e. micro) care receptioneza, deoarece majoritatea KERMIT-urilor pe micro nu includ facilitatea de time-out (astfel, daca receptorul nu este pregatit pentru receptie si emitatorul trimite primul pachet, se va produce un dead lock de protocol):

6.9.2.6 Alt mod de lucru: KERMIT ca server

Pina acum s-a descris versiunea initiala, simpla, a protocolului KERMIT (mai precis, a serviciilor oferite de acesta). O extensie (optionala) a protocolului include conceptul de server (KERMIT functionind in mod server). Un server KERMIT reprezinta un produs program KERMIT care nu interactioneaza direct cu utilizatorul, ci cu celalalt program KERMIT. Utilizatorul nu adreseaza comenzi unui KERMIT server, il lanseaza doar la un capat al conexiunii, dupa care lanseaza toate comenzile la celalalt capat.

Nu toate implementarile de KERMIT pot fi serveri, in schimb toate pot sa converseze cu un server KERMIT. Serverul este activat la calculatorul de la distanta, de obicei un minicalculator. Utilizatorul, desigur, trebuie sa se CONNECT-eze la calculatorul indepartat, sa "intre" (log in) in sistem si sa lanseze server-ul, dar nu trebuie nici sa declare un capat ca **send** si celalalt ca **receive**, nici sa se conecteze inapoi (cind transferul s-a terminat) la capatul de la distanta pentru a termina KERMIT-ul si pentru a termina sesiunea de lucru (logout). Utilizind serverul, se pot lansa oricite operatii **send** si **receive**, in orice secventa, fara sa fie nevoie sa se mai faca pe parcurs CONNECT-area la host-ul de la distanta. Anumiti serveri ofera, de asemenea, servicii suplimentare, ca afisare de directoare de fisiere, stergeri de fisiere sau alte solicitari de utilizari ale discului.

Un server KERMIT este un program KERMIT aflat intr-un mod de lucru (stare) special. El actioneaza ca si un KERMIT normal dupa ce i s-a dat comanda RECEIVE: asteapta un mesaj de la celalalt KERMIT, dar in acest caz mesajul reprezinta o comanda care solicita un serviciu (de obicei, SEND sau RECEIVE un fisier sau un grup de fisiere).

Dupa revenirea (prin secventa de "escape") la sistemul local utilizatorul poate da oricite comenzi (SEND si GET) doreste, iar la terminarea transferului fisierelor dorite se poate da comanda BYE, care transmite un mesaj la server-ul KERMIT pentru a-si termina singur functionarea. Nu mai este necesar sa se faca din nou conectarea la host-ul de la distanta si sa se ceara explicit terminarea lui. Totusi, daca se doreste conectarea din nou la host, trebuie folosita comanda FINISH in loc de BYE pentru a opri server-ul KERMIT din host fara a-l termina (permitind astfel comanda locala CONNECT).

In cele ce urmeaza se descrie un exemplu de utilizare a KERMIT in mod server. Utilizatorul sta la un micro Tim-S Plus, iar un minicalculator este host-ul de la distanta:

```
A>KERMIT                    ! Lanseaza KERMIT pe micro.
KERMIT-80 v4.05 configured for Tim-S/Plus vith VT52
>KERMIT-80>                 ! Acesta-i prompt-ul KERMIT-ului de
                              pe micro.
KERMIT-80>connect           ! Conecteaza la minicomputer.
[Connected to remote host. Type Ctrl-\C to return]
>HEL myuserid/mypassword    ! Login la minicomputer.
(the KER-11 prints various login messages here.)
>KER                        ! Lanseaza KER-11 normal.
KER-11 T2.17
```

```
KER-11>server            ! Spune-i sa fie server.
KERMIT Server running on PDP-11 host. Please type your
escape sequence to return to your local machine. Shut down
the server by typing the KERMIT command on your local
machine.
^\C                      ! Acum cu escape revii la micro.
[Connection closed, back at micro.]
KERMIT-80>get *.pas      ! Da-mi toate programele mele
                           Pascal
KERMIT-80>send foo.*     ! Trimite toate fisierele de tip
                           "foo" de la acest micro.
KERMIT-80>exit           ! Iesi din KERMIT si revii in CP/M.
A>
(Here you can do some work on the micro, edit files, whatever
you like.)
A>KERMIT                 ! Lanseaza KERMIT-80.
KERMIT-80>send file.pas  ! Trimite un alt fisier.
KERMIT-80>bye            ! Asta-i tot. Decupleaza KERMIT-ul
                           server.
A>                       ! Reintoarcere automata in CP/M.
```

Acest mod de operare este mult mai simplu. De observat,ca odata ce a fost pornit sreverul KERMIT la capatul indepartat, se poate lansa si opri KERMIT-ul local pe micro, ori de cite ori se doreste, fara a mai fi nevoie de conectare la revenirea de la host; totul este ca in final sa se lanseze comanda BYE.

Comenzile disponibile pentru "comunicarea" cu un server KERMIT sint:

SEND spec-fis

transfera un fisier sau un grup de fisiere de la host-ul local la cel de la distanta (ca si in modul normal: SEND-RECEIVE).

GET spec-fis

cere host-ului de la distanta sa trimita un fisier sau un grup de fisiere. Exemplu:

```
get *.C
```

Aceasta comanda este echivalenta cu "**send** *.C" la host-ul de la distanta, urmat de "**receive**" la cel local. De observat ca KERMIT-ul local nu incearca sa valideze **spec-fis**. Daca serverul nu il intelege, sau nu poate sa acceseze fisierele specificate, el va fi acela care va anunta eroarea (prin trimiterea unui mesaj corespunzator de eroare).

BYE

termina server-ul de la distanta si iese din KERMIT-ul local. Aceasta va determina, la fel, iesirea din KERMIT-ul de la distanta. Nu este nevoie sa se faca o conectare inapoi la server si sa se faca "curatenie", decit daca se obtine un mesaj de eroare ca raspuns la aceasta comanda (spre exemplu, daca operatia de terminare la distanta nu se poate face din motive de lipsa de resurse la host-ul de la distanta).

FINISH

termina modul de lucru server al KERMIT-ului de la distanta, fara sa iasa din KERMIT-ul local. O comanda CONNECT ulterioara va pune utilizatorul local inapoi la host-ul de la distanta, la nivelul

comenzilor sistem.

6.9.3 Cind apar probleme

Conectarea a doua calculatoare poate parea o sarcina relativ dificila, si intr-adevar unele lucruri pot fi solutionate mai greu. Inainte de orice tentativa de a tranfera fisiere, trebuie stabilita comunicatia la nivel de terminal. Dar, pe de alta parte, o conectare cu succes la nivel de terminal nu inseamna neaparat ca transferul de fisiere poate avea loc. Si chiar atunci cind transferul de fisiere pare sa "lucreze", pot aparea probleme care sa termine nefavorabil actiunea.

6.9.3.1 Probleme ale liniilor de comunicatie

Daca pe o versiune de micro de KERMIT, comanda CONNECT nu "pare" sa mearga, se recomanda urmatoarele:

- sa se asigure ca toate conexiunile fizice au fost facute corect. Daca se folosesc modemuri, trebuie verificat ca becul de purtatoare ("carier") este aprins;
- daca micro-ul dispune de mai multi conectori, sa se asigure ca este folosit cel corect;
- sa se asigure ca portul este setat la viteza de comunicatie corecta. Anumite versiuni de KERMIT au comenzi proprii de tip SET BAUD, altele necesita folosirea unor comenzi de sistem pentru modificarea vitezei (de obicei, inainte de a lansa programul KERMIT, daca nu exista facilitatea de lansare de comenzi sistem din KERMIT). Pentru a se afla viteza curenta a liniei, se poate folosi comanda SHOW.
- sa se asigure ca parametrii liniei de comunicatie (paritatea, biti pe caracter, control de flux etc.)sint corect setati.

Pentru probleme legate de echipamentele de comunicatie si sistem, trebuie desigur consultate manualele corespunzatoare.

Daca toate setarile de parametrii si conexiunile fizice par sa fie corecte, dar totusi comunicatia nu are loc, eroarea poate sa provina de la modem. Modemurile interne (i.e. cele incluse in carcasa microcalculatorului) nu sint recomandate a fi folosite cu KERMIT. Aceasta deoarece programele KERMIT de pe microcalculatoare controleaza hardware-ul de comunicatie explicit, iar modemurile interne pot interfera la nivel de comanda cu comenzile interne date de KERMIT.

In mod normal, KERMIT presupune ca are un control complet asupra portului de comunicatie. Totusi, uneori anumite echipamente de comunicatie controleaza linia dintre doua calculatoare, la fiecare capat. Echipamentele de acest tip pot fi: modemuri (in particular modemuri "inteligente"), unitati de selectie/control al porturilor, multiplexoare, retele locale si retele generale. Un astfel de echipament **poate** sa interfere cu protocolul de transfer de fisiere al KERMIT in diferite moduri:

-Poate sa impuna paritate pe linia de comunicatii. Aceasta inseamna ca al 8-lea bit al fiecarui caracter este utilizat de echipament pentru a verifica corectitudinea transmisiei. Utilizarea paritatii pune urmatoarele probleme:
  -va determina un calcul incorect al CRC-ului la receptor, si va bloca astfel transferul de fisiere. In majoritatea cazurilor, nici chiar primul pachet nu va fi declarat receptionat cu succes;

-nu va permite folosirea bitului 8 pentru fisiere cu date binare.

-Daca conexiunea de terminal functioneaza, dar transferul de fisier nu poate avea loc, paritatea este problema cea mai vizata a fi cauza. Pentru a remedia problema trebuie ca utilizatorul sa afle care este paritatea folosita si sa informeze KERMIT-urile la fiecare capat (utilizind comanda SET PARITY), astfel incit ambele programe KERMIT:

-sa compuna si respectiv sa interpreteze CRC-ul in mod corect si unitar;

-sa foloseasca aceeasi codificare, pentru a permite ca datele pe 8 biti sa poata fi transferate chiar daca canalul de comunicatie este pe 7 biti.

-Echipamentul de comunicatie respectiv poate, de asemenea, sa interpreteze anumite caractere din fluxul de date drept comenzi proprii, in loc sa le transfere mai departe. Spre exemplu, un modem "inteligent" poate la un moment dat sa deconecteze beneficiarul lui local si sa-l conecteze la o alta destinatie. Singurul mod de lucru este de a trece dispozitivul respectiv in modul "transparent" sau "binar" de functionare. In anumite cazuri, modul transparent de functionare va elimina si prelucrarea paritatii, astfel incit permite utilizarea completa a tuturor celor 8 biti pentru date.

6.9.3.2 Transferul este blocat

Datorita unor motive diverse, transferul de fisiere prin KERMIT se poate bloca. Insa, deoarece majoritatea host-urilor sint capabile sa genereze intreruperi de time-out daca evenimentele de intrare (receptii de mesaje) nu apar suficient de repede, acestea pot sa implementeze mecanisme de control al fluxului (prin retransmisii sau/si confirmari negative ale pachetelor pierdute). Totusi, daca transferul pare sa fie blocat, utilizatorii de KERMIT de micro pot sa tasteze RETURN, pentru a simula un time-out.

In urmatoarele sectiuni se prezinta diferite cazuri posibile de blocare a transferului. Intr-un caz concret, inainte de a examina aceste situatii, trebuie sa se asigure ca intr-adevar exista un KERMIT la fiecare capat al liniei si ca s-a lansat comanda corespunzatoare: SEND,RECEIVE sau SERVER. Daca KERMIT-ul de la distanta nu este server, reamintim ca trebuie facuta conectarea inapoi la host-ul de la distanta, intre fiecare transfer, si data o noua comanda SEND sau RECEIVE.

6.9.3.3 Microcalculatorul este "agatat"

Micro-ul insusi poate sa ramina "agatat" in anumite conditii, uneori fara vreo legatura cu KERMIT-ul (spre exemplu, datorita fluctuatiilor de tensiune). Daca ecranul micro-ului nu a fost actualizat o perioada mare de timp, atunci se poate ca micro-ul sa fie agatat. Pentru a depista aceste cazuri se propune, in ordine, urmarea acestor pasi:

-sa se verifice conexiunea. Sa se asigure ca nici un conector nu a iesit din mufa. Daca se foloseste un modem, sa se asigure ca semnalul de purtatoare exista in continuare. Daca este necesar, sa se restabileasca conexiunea fizica.

-sa se apese tasta RETURN pentru a "trezi" micro-ul. Aceasta

operatie ar trebui sa scoata protocolul dintr-un eventual dead lock. Uneori sint necesare mai multe RETURN-uri.

-daca problema nu consta intr-un dead lock, trebuie repornit micro-ul si apoi restartat in sistem si restartat transferul. Eventual va trebui sa se opreasca si apoi sa se reporneasca si KERMIT-ul de pe host-ul de la distanta.

6.9.3.4 Host-ul de la distanta creeaza problemele

Daca sistemul local este operational, atit prin el insusi cit si din punct de vedere al KERMIT-ului, dar transferul este blocat, probabil ca host-ul de la distanta sau programul KERMIT de acolo s-a blocat. In acest caz trebuie revenit la nivelul comanda pe KERMIT-ul local (in implementarile de pe micro, aceasta se poate obtine prin tastarea a aproximativ cinci RETURN-uri sau a unuia sau mai multor CONTROL/C-uri). Apoi se da comanda CONNECT astfel incit sa se vada ce s-a intimplat cu sistemul de la distanta. Daca acesta din urma s-a blocat, atunci trebuie asteptat pina va fi repornit si trebuie reluat transferul, cel putin de la fisierul care era in curs in momentul evenimentului.

6.9.3.5 Discul este plin

Daca pe floppy discul local sau pe discul de la host-ul indepartat nu mai exista spatiu liber suficient, KERMIT-ul de pe masina unde aceasta resursa a fost epuizata va informa utilizatorul, iar transferul va fi oprit. Aceasta poate continua transferul prin repetarea intregii proceduri dupa eliberarea corespunzatoare a resursei de memorie de disc. Anumite programe KERMIT (spre exemplu, de pe minicalculatoare de tip CORAL, sub RSX 11M), permit continuarea transferului din punctul in care a fost oprit, prin utilizarea comenzii SEND si includerea numelui fisierului la care s-a oprit transferul in cimpul "initial":

```
>KER
KER>send *.for (initial) foo.for
```

Informatii suplimentare privind **spec-fis** cu **initial** sint date in capitolul cu comenzi KERMIT pentru mini.

6.9.3.6 Interferenta mesajelor

Se poate intimpla ca transferul de fisiere sa se blocheze uneori neasteptat (fara a fi cauzat de vreo situatie descrisa anterior). O explicatie ar putea fi ca mesajele de terminal sint amestecate cu mesajele care transporta pachetele de date de fisier. Aceste mesaje pot fi mesaje broadcast de sistem (de exemplu, cele din procedura de SHUT UP: "System is going down in 20 minutes"), mesaje transmise de alti utilizatori ai sistemului host pe linia de terminal folosita de KERMIT ("Hei D, ce este cu programul acesta KERMIT pe care il tot rulezi?"), sau notificari pe care utilizatorul le-a cerut de la aplicatii de sistem (posta electronica etc.) sau personale ("Este 7:30, trebuie sa ...", sau "Ati primit prin posta electronica ...").

Majoritatea programelor KERMIT incearca sa dezactiveze automat aceste mesaje, dar nu poate fi garantat pentru orice versiune. Din acest motiv, este recomandat ca inainte de a se lansa KERMIT sa se opreasca, daca este posibil, orice sursa de mesaje

pe linia care va fi folosita de KERMIT.

6.9.3.7 Erori la host

In sistemul host indepartat pot apare situatii de exceptie foarte diverse care sa afecteze transferul de fisiere. Ori de cite ori apar astfel de erori, KERMIT-ul de la distanta incearca sa transmita mesaje de eroare informative la cel local si apoi opreste transmisia, dind controlul utilizatorului la nivel comanda de pe sistemul local.

6.9.3.8 Fisierul este stricat

Exista anumite situatii in care KERMIT-ul considera ca un fisier a fost transferat corect, dar in realitate el este corupt (mai precis, nu poate fi folosit ca si copia lui originala). Cauzele cele mai probabile pot fi legate de atribute de fisiere necorespunzatoare (ex. text fata de binar, 7 biti fata de 8 biti, mod blocat fata de mod flux, etc.).

Fiecare sistem are caracteristicile sale, si fiecare tip de KERMIT are comenzi speciale care sa-i permita sa se specifice modul in care un fisier trebuie sa fie transmis sau memorat. Totusi, probleme de acest gen apar doar la transferul de fisiere binare, fisierele text nu ar trebui sa prezinta probleme la transferul intre oricare doua programe KERMIT.

6.9.3.9 Erori la sfirsitul de fisier

Uneori, cind se transfera un fisier text, de la un microcalculator la un minicalculator, se pot gasi caractere "ciudate" la sfirsitul fisierului transferat. Aceasta se datoreaza faptului ca multe microcalculatoare nu au un mod consistent de a indica sfirsitul de fisier. CP/M-ul este un exemplu bun in acest sens. Unitatea minima de memorie pe un floppy disc CP/M este un bloc de 128 octeti. Fisierele binare constau intotdeauna dintr-un numar intreg de blocuri, dar fisierele text pot sa se termine oriunde in cadrul unui bloc. Deoarece CP/M nu pastreaza un contor de lungime de fisier (sau de baiti intr-un bloc), el utilizeaza o conventie de marcare a sfirsitului de fisier prin folosirea caracterului Control-Z. Daca versiunea de KERMIT de pe micro sub CP/M nu cauta acest caracter, ea va trimite ultimul bloc complet, care va contine desigur dupa sfirsitul real al fisierului caractere "ciudate". Pentru a rezolva aceasta problema, majoritatea programelor KERMIT pe micro au implementate comenzi ca SET FILE ASCII sau SET FILE TEXT sau pentru a forta KERMIT sa respecte conventia Ctrl/Z.

Anumite KERMIT-uri pe micro functioneaza implicit in mod "text", altele in mod "binar" sau "bloc", deci trebuie avuta in vedere trecerea in modul dorit de lucru.

6.9.4 Comenzi KERMIT

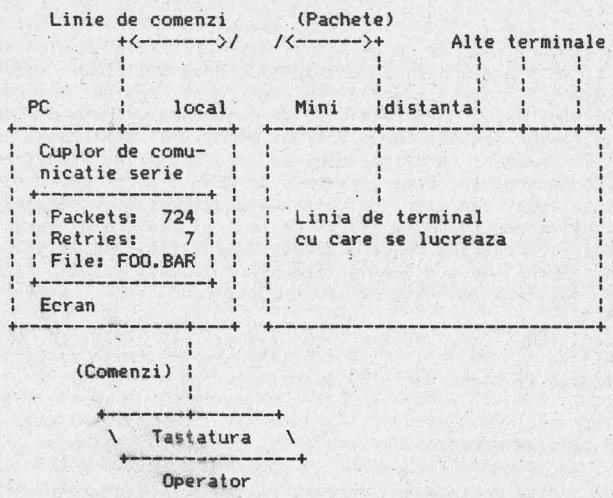
In capitolul de fata se vor descrie serviciile unui program KERMIT "ideal", care respecta (implementeaza) majoritatea facilitatilor oferite de protocolul KERMIT. In realitate, nici un program KERMIT nu descrie toate aceste comenzi sau nu suporta toate optiunile. In plus, forma exacta a unor comenzi poate sa

difere de la o versiune la alta. Anumite programe KERMIT pot, de asemenea, sa suporte optiuni dependente de sistem care nu sint descrise aici. Intentia acestei specificari este de a oferi o baza ulterioara de descriere a serviciilor specifice unor implementari reale de KERMIT.

6.9.4.1 Functionare locala si la distanta

Anumite programe KERMIT pot functiona in doua moduri, local si la distanta. Un KERMIT in mod "la distanta" ruleaza de obicei pe un mini sau sistem mare, la care s-a facut conectarea de la un PC sau alt calculator. Cind KERMIT-ul functioneaza la distanta, orice transfer de fisiere se desfasoara pe linia de terminal de intrare in sistemul de la distanta; aceeasi linie prin care s-a facut log-in si prin care s-au dat interactiv comenzi la sistem. Sistemul considera celalalt calculator pe care ruleaza o copie a KERMIT ca fiind un terminal propriu, pina cind se lanseaza si acolo KERMIT, cind controlul liniei respective este preluat, de obicei exclusiv de catre KERMIT.

Cind KERMIT-ul este in "mod local", transferul are loc printr-un dispozitiv extern - ca de exemplu portul de comunicatie serial al microcalculatorului - sau pe o linie de terminal asignata la un mini sau sistem mare. KERMIT-ul local este conectat intr-un mod specific (ca un mecanism de apel automat) la un alt calculator, pe care de asemenea ruleaza o copie a KERMIT. Un KERMIT local are controlul ecranului (display-ului), in timp ce KERMIT-ul de la distanta nu are acces direct la terminal. KERMIT-ul de pe microcalculator ruleaza implicit in "mod local", in timp ce KERMIT-ul de pe mini sau sistem mare de obicei necesita comenzi speciale pentru a functiona in mod local. Anumite comenzi au sens doar pentru KERMIT-uri la distanta, altele doar pentru local, iar altele pot fi folosite in ambele moduri. In figura de mai jos se prezinta schematic functionarea locala si la distanta (PC este local, iar mini este la distanta).



Programul KERMIT care ruleaza pe PC este KERMIT local. El poate controla ecranul, tastatura si cuplorul de comunicatie

separat. Astfel el poate: modifica ecranul cu informatii de stare, primi semnale de intrerupere de la claviatura, si transfera pachete pe linia de comunicatie, si toate acestea in acelasi timp.

Programul KERMIT care ruleaza pe minicalculator este KERMIT la distanta. Utilizatorul deschide o sesiune la minicalculator prin intermediul unui cuplor de terminal. Tastatura, ecranul si cuplorul de comunicatie sint toate combinate in linia de terminal a minicalculatorului.

Un KERMIT "server" este intotdeauna la distanta, si trebuie sa obtina comenzile sale de la KERMIT-ul local. Descrierea va indica cind o comanda trebuie sa fie la distanta sau locala.

6.9.4.2 Interfata de comanda

Ca raspuns la "KERMIT-xx>" se poate tasta un cuvint cheie (de exemplu: SEND, RECEIVE sau EXIT) urmat optional de un alt cuvint cheie sau operand.

6.9.4.3 Notatii

anything - un parametru;
[anything] - un cimp optional; daca se omite, se atribuie o valoare initiala;
numar - un numar intreg;
caracter - un singur caracter: introdus literal sau ca un numar reprezentind valoarea ASCII a caracterului;
filespec - o specificatie de fisier, adica numele unui fisier, incluzind eventual si dispozitivul si UIC-ul, si alte informatii; poate contine caractere "wildcard" pentru a desemna un grup de fisiere.
^X - un caracter de control poate fi scris folosind notatia cu "sageata in sus".

Comenzile sint aratate cu litere mari, dar pot fi introduse in orice combinatie de litere mari si mici.

6.9.4.4 Comenzile KERMIT

Lista comenzilor KERMIT pentru:

- transferul de fisiere: SEND, RECEIVE, GET;
- conectare la un calculator de la distanta: CONNECT, SET LINE, SET PARITY, SET DUPLEX, SET HANDSHAKE,SET ESCAPE, SET FLOW-CONTROL;
- a actiona ca un "server": SERVER;
- a discuta cu un "server": BYE, FINISH, GET,SEND, REMOTE;
- setarea parametrilor de transmisie si de fisier: SET BLOCK-CHECK, SET DEBUG, SET DELAY, SET FILE, SET INCOMPLETE, SET PARITY, SET RETRY, SET SEND (sau RECEIVE),END-OF-LINE,START-OF-PACKET,PACKET-LENGTH, PAUSE, TIMEOUT, PADDING;
- definirea de "macro" a comenzii SET: DEFINE;
- pentru intreruperea transmisiei: Control-X, Control-Z, Control-C;
- obtinerea de informatii: HELP, STATISTICS, SHOW;
- executarea de fisiere de comanda: TAKE;
- inregistrarea istoriei unei operatii de transfer de fi-

sier: LOG TRANSACTIONS;
- transmisie de fisier fara protocol: LOG SESSION, TRANSMIT;
- terminarea programului: EXIT, QUIT.

Daca exista un fisier KERMIT.INI, Kermit va executa automat o comanda TAKE la pornire. KERMIT.INI poate contine orice comenzi KERMIT, ca de exemplu comenzi SET, sau DEFINE pentru macro SET, pentru a configura KERMIT la diverse sisteme sau medii de comunicatie.

6.9.5 KERMIT sub RSX11M

6.9.5.1 Generalitati

Protocolul de transfer de fisiere KERMIT se recomanda a fi folosit intr-un loc unde exista o mare varietate de calculatoare - microcalculatoare, calculatoare personale, calculatoare de proces, sisteme timesharing - fabricate de diversi producatori. Toate aceste sisteme trebuie sa poata comunica in ASCII prin linii de comunicatie seriale obisnuite.

KERMIT a fost initial proiectat pentru a facilita transferul de fisiere intre calculatoare mari si microcalculatoare.

Protocolul KERMIT este proiectat pentru transmisii orientate pe caracter, pe linii de telecomunicatie seriale. Proiectarea a tinut seama de restrictiile mediului de comunicatie si de cerintele diverselor sisteme de operare - buffer-are, transmisie duplex, paritate, set de caractere, organizare de fisiere etc. Protocolul este stabilit intre programe KERMIT care se afla la fiecare capat al conexiunii prin trimitere de pachete; transmitatorul trimite numele fisierului, continutul fisierului si informatii de control; receptorul confirma (pozitiv sau negativ) fiecare pachet.

Pachetele au o proiectare structurata pe nivele, in conformitate cu recomandarile ANSI si ISO, avind cimpuri pentru asigurarea integritatii datelor, asigurarea continuitatii si datele propriu-zise.

Conexiunile intre sisteme sint stabilite de catre utilizator. Intr-un caz tipic utilizatorul ruleaza programul KERMIT pe un microcalculator, se conecteaza prin emulare de terminal la un calculator de la distanta, deschide o sesiune, lanseaza KERMIT pe calculatorul de la distanta, ii da comanda pentru a starta un transfer de fisiere, revine la micro si da comanda de startare a transferului de fisiere din acest capat. Fisierele pot fi transferate singure sau in grup.

KERMIT permite rularea in mod "server" pe calculatorul de la distanta; in acest fel se pot transfera fisiere in orice directie prin comenzi date da catre utilizatorul de la KERMIT local. Server-ul poate executa comenzi suplimentare pentru gestiunea fisierelor, transmitere de mesaje etc.

Exista optiuni pentru diverse tipuri de verificare a transmisiei, un mecanism de a trimite date de 8 biti prin linii de comunicatie de 7 biti, un mijloc de a comprima secvente de caractere care se repeta, si altele.

KERMIT permite utilizatirilor obisnuiti sa stabileasca conexiuni sigure si fara erori intre doua calculatoare.

In continuare se descriu comenzile si setarile de parametri pentru KERMIT care ruleaza pe un minicalculator.

6.9.5.2 @

Aceasta comanda deschide un fisier de comenzi indirecte. Este identica cu comanda TAKE. Formatul este:

```
KER-11>@ file-specification
```

unde "file-specification" este orice specificare de fisier de pe sistemul pe care ruleaza KERMIT.

6.9.5.3 Fisiere binare

Fisierele binare sint acele fisiere care folosesc toti cei 8 biti dintr-un caracter. Fisierele de text (ca programele sursa, listing-uri de compilare si fisiere RUNOFF) folosesc numai 7 biti in fiecare caracter. Fisierele binare cuprind imagini de task-uri, fisiere RMS relative si indexate si alte tipuri de fisiere speciale.

Pentru a transfera fisiere binare KER-11 necesita o linie de comunicatie pe 8 biti. In plus, KERMIT trebuie pus in mod binar cu comanda "SET FILETYPE BINARY" sau "SET FIL BIN" prescurtat. Aceasta comanda trebuie data in ambele capete ale conexiunii pentru a se putea transfera fisiere binare.

6.9.5.4 BYE

Comanda BYE data la KERMIT local este transmisa la celalalt KERMIT (care se afla in mod server) pentru a termina executia. La primirea confirmarii comenzii, KERMIT-ul local va accepta o noua comanda care ar trebui sa fie cea de DISCONNECT. Formatul este:

```
KER-11>BYE
```

6.9.5.5 CONNECT

Comanda CONNECT va permite conectarea ca un terminal virtual pe linia care a fost specificata la comanda SET LINE. Formatul este:

```
KER-11>CONNECT
```

Inainte de executia acestei comenzi trebuie data comanda:

```
MCR:SET /FDX=TI:.
```

De asemenea, trebuie setata viteza corespunzatoare a liniei de comunicatie, fie cu comanda SET SPEED sau cu comanda MCR: SET /SPEED=TTnn:xxxx:xxxx. (Numai liniile de pe multiplexor pot fi setate la o anumita viteza cu aceste comenzi).

Dupa conectare exista o "secventa de escape" (control\C) care permite revenirea la KERMIT-ul local. **Control** informeaza emulatorul de terminal ca urmatorul caracter este o comanda. **Control\?** va tipari un mesaj de help.

Urmeaza un exemplu de conectare de la distanta a unui sistem CORAL:

```
SET /FDX=TI:
>KER
KER-11 T2.17
KER-11>set logfile f.log
```

```
Created debug file f.log
KER-11>set deb console
KER-11>set lin TT4:
Link device set to TT4:
KER-11>set spe 1200
KER-11>con
Connecting to line TT4: at 1200 baud
Type CTRL \ and then C to return to command level.
Note: This is not a remote line.
HEL
Account or name 7,2
Password

RSX-11M BL32 [1,54] System RSX 11M
22-MAY-87 08:55 Logged on Terminal TT11:

Good Morning
          !
>PIP  /BR !  ! Se poate lucra in continuare pe calculato-
          !    rul de la distanta.

>BYE

Have a good Morning
22-MAY-87 09:10 ! Acum trebuie tastat CTRL\C pentru revenire
                  la sistemul local.

KER-11>disc
KERMIT link TT4: disconnected
KER-11>exit
Debug file closed
```

In acest moment fisierul **f.log** contine dialogul sustinut.

6.9.5.6 COPY

Comanda COPY creaza o copie a fisierului de intrare. Nu suporta "wildcards" in numele fisierului. Acelasi lucru se poate executa prin utilitarul PIP. Formatul este:

```
KER-11> COPY
From: ABCDEF.DAT
To  : FUBAR.LST
```

sau

```
KER-11> COPY ABCDEF.DAT FUBAR.LST
```

6.9.5.7 DELETE

Comanda DELETE va sterge fisierul specificat sau un grup de fisiere din sistem. Formatul este:

```
KER-11>DELETE FUBAR.MAC
KER-11>DELETE FUBAR.*
```

6.9.5.8 DIRECT

Comanda DIRECT va afisa lista fisierelor din director-ul curent cind se foloseste formatul:

```
KER-11>DIRECT
```

Daca se doreste listarea din alt director se va folosi formatul:

```
KER-11>DIRECT file-specification
```

unde "file-specification" este orice combinatie valida de nume de periferic, UIC sau nume de fisier care poate cuprinde si caractere "wildcard".

6.9.5.9 DISCONNECT

Comanda DISCONNECT are ca efect deconectare liniei de comunicatii care a fost specificata prin comanda SET LINE si la care s-a facut conectarea prin comanda CONNECT. Formatul este:

```
KER-11>DISCONNECT
```

6.9.5.10 DISPLAY

Comanda DISPLAY va afisa datele globale definite in programul KERMIT, ca de exemplu:

```
KER-11>DISP PROMPT
```

6.9.5.11 ERASE

Comanda ERASE va sterge fisierul specificat sau un grup de fisiere din sistem. Caracterele "wildcard" sint permise.

```
KER-11>ERASE FUBAR.MAC
KER-11>ERASE FUBAR.*
```

Comanda ERASE este identica cu comanda DELETE.

6.9.5.12 EXIT

Terminarea executiei programului KERMIT se face cu comanda EXIT (la fel ca si cu comanda QUIT).
Formatul este:

```
KER-11>EXIT
```

6.9.5.13 FINISH

Comanda FINISH indica programului KER-11, care este in mod local, sa semnaleze celuilalt KERMIT sa termine modul server. Cind KER-11 va receptiona confirmarea ca aceasta s-a facut, se poate da o noua comanda. Formatul este:

```
KER-11>FINISH
```

6.9.5.14 GET

Comanda GET indica programului KERMIT care este in mod server sa trimita un fisier sau un grup de fisiere. Formatul este:

KER-11>GET file-specification

unde "file-specification" poate fi orice combinatie valida de nume de periferic, UIC sau nume de fisier care poate cuprinde si caractere "wildcard".

6.9.5.15 HANGUP

Comanda HANDGUP va intrerupe comunicarea pe linia de comunicatie specificata prin comanda SET LINE. Aceeasi functie poate fi executata si cu comanda DISCONNECT. Formatul este:

KER-11>HANGUP

6.9.5.16 HOST

Comanda HOST (identica cu comanda SYSTEM) permite executarea unei comenzi pe sistemul gazda.

6.9.5.17 LOCAL

Comanda LOCAL poate fi folosita ca prefix pentru orice comanda KERMIT pentru a fi siguri ca aceasta comanda se executa de catre KERMIT-ul local. Toate comenzile sint locale cu exceptia comenzilor BYE, FINISH si GET. Formatul este:

KER-11>LOCAL KERMIT comanda

6.9.5.18 QUIT

Comanda QUIT (identica cu comanda EXIT) termina executia programului KERMIT. Formatul este:

KER-11>QUIT

6.9.5.19 PRINT

Comanda PRINT va tipari fisierele specificate pe imprimanta sistemului.

6.9.5.20 RECEIVE

Comanda RECEIVE va pune KER-11 in modul de asteptare a unei singure tranzactii de transfer de fisier. KER-11 va astepta pentru o initializare de transfer de fisier de la alt KERMIT. Aceasta comanda este folositoare daca celalalt KERMIT nu suporta modul server. Formatul este:

KER-11>RECEIVE

6.9.5.21 REMOTE

Comanda REMOTE este folosita ca un prefix pentru a indica programului KER-11 ca urmeaza sa se execute o comanda de catre KERMIT-ul de la distanta, aflat in mod server. Formatul este:

```
KER-11>REMOTE
```

apoi va cere numele comenzii:

```
Remote KERMIT cmd ?
```

care poate fi dat ca argument:

```
KER-11>REMOTE  numele comenzii
```

In continuare se vor prezenta formele sub care se poate utiliza comanda REMOTE.

6.9.5.21.1 BYE

Comanda REMOTE BYE va termina KERMIT-ul de la distanta aflat in mod server. Comanda DISCONNECT ar trebui sa urmeze intotdeauna. Aceasta comanda (REMOTE BYE) este aceeasi cu comanda BYE. Formatul este:

```
KER-11>REMOTE BYE
```

sau

```
KER-11>REMOTE
Remote KERMIT cmd ?BYE
```

6.9.5.21.2 COPY

Comanda "REMOTE COPY" creeaza (ca si comanda de copiere locala) o copie a fisierului de intrare. Nu suporta "wildcard" in numele de fisier. Formatul este:

```
KER-11>REMOTE COPY
From: ABCDEF.DAT
To  : FUBAR.LST
```

sau

```
KER-11>REMOTE COPY ABCDEF.DAT FUBAR.LST
```

6.9.5.21.3 DIRECT

Comanda DIRECT va afisa o lista a fisierelor din directorul sub care lucreaza TRAVEl-ul de la distanta, aflat in mod server.

```
KER-11>REM DIR *.MAC
```

6.9.5.21.4 ERASE

Comanda ERASE va sterge un fisier sau un grup de fisiere la distanta. De exemplu:

```
KER-11>REM ERASE *.MAC

Files deleted:
SY: [2,2]FUBAR.MAC
```

```
SY: [2,2]HLPSUB.MAC

KER-11>
```

KERMIT-ul de la distanta va indica fisierele sterse.

6.9.5.21.5 FINISH

Prin comanda FINISH se semnaleaza KERMIT-ului da la distanta sa termine modul server.
Comanda REMOTE FINISH are acelasi efect ca si comanda FINISH. Formatul comenzii este:

```
KER-11>REMOTE FINISH
```

sau

```
KER-11>REMOTE
Remote KERMIT cmd ?FINISH
```

6.9.5.21.6 GET

Comanda REMOTE GET cere programului KERMIT de la distanta (care este in mod server) sa execute trimiterea unui fisier sau a unui grup de fisiere. Are acelasi efect ca si comanda GET. Formatul este:

```
KER-11>REMOTE GET
```

sau

```
KER-11>REMOTE
Remote KERMIT cmd ?GET
```

6.9.5.21.7 HELP

Comanda REMOTE HELP cere programului KERMIT de la distanta (care este in mod server) sa trimita o lista a comenzilor pe care le poate executa.

6.9.5.21.8 RENAME

Comanda REMOTE RENAME este folosita pentru a renumi un fisier sau un set de fisiere. Formatul este:

```
KER-11>REMOTE RENAME nume vechi  nume nou
```

Argumentele pot fi cerute:

```
KER-11>REMOTE RENAME
From: nume vechi
To  : nume nou
```

KERMIT-ul de la distanta va indica o lista de fisiere nedenumite sau un numar de fisiere.

```
KER-11>REMOTE RENAME t.* junk
Remote ack: 3 files renamed
```

6.9.5.21.9 TYPE

Comanda REMOTE TYPE la fel ca si comanda TYPE semnaleaza programului KERMIT de la distanta (care este in mod server) sa trimita un fisier pentru a fi tiparit la terminal.

```
KER-11>REMOTE TYPE FUBAR.MAC
```

6.9.5.22 RENAME

Comanda RENAME este folosita pentru a renumi fisiere locale. Formatul este:

```
KER-11>RENAME nume vechi  nume nou
```

Argumentele pot fi cerute:

```
KER-11>RENAME
From: nume vechi
To  : nume nou
```

Comanda RENAME afiseaza o lista a fisierelor redenumite:

```
KER-11>RENAME t.* junk
File DP0:[1,8]T.T1 renamed to DP0:[1,8]JUNK.T1
File DP0:[1,8].T renamed to DP0:[1,8]JUNK.T
```

6.9.5.23 RSX11M

Programul KER-11 trebuie instalat checkpoint-abil.
Comanda CONNECT se va executa corect daca exista optiunea "typeahead" pentru driver-ul de terminal.

6.9.5.24 SEND

Comanda SEND permite utilizatorului sa trimita fisiere celuilalt KERMIT. Prin comanda SET DELAY se poate specifica numarul de secunde care vor trece pina cind va incepe transmiterea fisierelor. Aceasta da timp utilizatorului sa revina in KERMIT-ul local si sa execute comanda RECEIVE.
Formatul este:

```
KER-11>SEND file-specification
```

unde "file-specification" poate include nume de periferice, UIC, nume de fisiere, caractere "wildcard". Folosirea caracterelor "wildcard" este cea mai raspindita metoda de a indica un grup de fisiere intr-o singura specificare de fisier. De exemplu, FOO.FOR este un singur fisier, un program FORTRAN denumit FOO, iar *.FOR poate fi un grup de programe FORTRAN.

6.9.5.25 SERVER

Comanda SERVER va pune programul KER-11 in mod server. Cind KER-11 este in mod server, celalalt KERMIT poate trimite sa ii semnaleze trimiterea sau receptia de fisiere fara sa se dea comenzile SEND sau RECEIVE. Pentru a se transmite in mod corect fisiere binare, trebuie sa fie data comanda SET FILETYPE BINARY.

Formatul este:

```
KER-11>SERVER
```

In mod server se pot executa comenzile:

```
BYE            terminarea programului KER-11;
REMOTE COPY    copierea unui fisier;
REMOTE DIR     tiparirea unei liste de fisiere;
REMOTE ERASE   stergerea unui fisier;
FINISH         terminarea modului server;
GET            trimiterea de fisiere;
REMOTE HELP    tiparirea unei liste de comenzi;
REMOTE RENAME  redenumirea unui fisier;
REMOTE TYPE    tiparirea unui fisier.
```

6.9.5.26 SET

Comanda SET este folosita pentru a seta diversi parametri in KERMIT. Formatul este:

```
KER-11>SET parametru  cuvint cheie
```

6.9.5.26.1 ATTRIBUTES

Prin protocolul KERMIT se pot indica atributele de fisier: informatii asupra marimii fisierului, datei de creere, tipului de fisier. Datorita unor incompatibilitati existente intre sistemele interconectate, aceasta caracteristica poate fi dezactivata.

```
KER-11>SET ATTRIBUTES OFF
KER-11>SET ATTRIBUTES ON
```

6.9.5.26.2 BAUD

Are acelasi efect ca si comanda SET SPEED.

6.9.5.26.3 DEBUG

Comanda SET DEBUG este folosita pentru a specifica tipul si nivelul de depanare pe un fisier disk. Fisierul disc trebuie sa fie creat prin comanda SET LOGFILE. Formatul este:

```
KER-11>SET DEBUG tip
```

6.9.5.26.3.1 ALL

Comanda SET DEBUG ALL va activa toate tipurile: CONSOLE, CONECT, FILE, PACKET si STATE. Are acelasi efect ca si comanda SET DEBUG ON. Formatul este:

```
KER-11>SET DEBUG ALL
```

6.9.5.26.3.2 CONSOLE

Comanda SET DEBUG CONSOLE va permite inregistrarea tuturor intrarilor/iesirilor in timpul unei conexiuni la distanta, pe un fisier disk specificat prin SET LOGFILE. Are acelasi efect ca si comanda SET DEBUG CONNECT. Formatul este:

```
KER-11>SET DEBUG CONSOLE
```

6.9.5.26.3.3 CONNECT

```
KER-11>SET DEBUG CONNECT
```

6.9.5.26.3.4 FILE

Comanda SET DEBUG FILE va activa inregistrrea fisierelor deschise sau create. Formatul este:

```
KER-11>SET DEBUG FILE
```

6.9.5.26.3.5 HELP

Comanda SET DEBUG HELP da o lista a tipurilor de DEBUG care pot fi folosite. Formatul este:

```
KER-11>SET DEBUG HELP
```

6.9.5.26.3.6 NONE

Comanda SET DEBUG NONE dezactiveaza toate tipurile de depanare, la fel ca si comanda SET DEBUG OFF. Formatul este:

```
KER-11>SET DEBUG NONE
```

6.9.5.26.3.7 OFF

```
KER-11>SET DEBUG OFF
```

6.9.5.26.3.8 ON

```
KER-11>SET DEBUG ON
```

6.9.5.26.3.9 PACKET

Comanda SET DEBUG PACKET va actiona inregistrarea tuturor pachetelor receptionate si transmise pe un fisier disc. Formatul este:

```
KER-11>SET DEBUG PACKET
```

6.9.5.26.4 DELAY

Parametrul de DELAY este numarul de secunde care trebuie asteptat inainte de trimiterea unui fisier dupa ce s-a dat comanda SEND. Formatul este:

KER-11>SET DELAY numar de secunde (in zecimal)

Acesta este folosit pentru a permite utilizatorului sa revina in KERMIT-ul local si sa execute comanda RECEIVE.

6.9.5.26.5 DEFAULT

Parametrul DEFAULT permite sa se specifice perifericul si UIC-ul pentru urmatoarele fisiere deschise (pentru SEND) sau create (pentru RECEIVE). Este dezactivat prin SET HOME. Formatul este:

```
KER-11>SET DEFAULT device
KER-11>SET DEFAULT DP2:[200,201]
```

6.9.5.26.6 DUPLEX

La comanda SET DUPLEX HALF se va face un ecou local al caracterelor tratate dupa comanda CONNECT. Formatul este:

```
KER-11>SET DUPLEX HALF
KER-11>SET DUPLEX FULL
```

6.9.5.26.7 END-OF-LINE

Parametrul END-OF-LINE seteaza caracterul ASCII care va fi folosit ca terminator de linie pentru orice pachet trimis de KERMIT.

KER-11>SET END-OF-LINE valoare octala a caracterului

6.9.5.26.8 ESCAPE

Aceasta comanda va seta caracterul de escape pentru iesirea din mod CONNECT. Valoarea initiala folosita este Control\(octal 34).

KER-11>SET ESCAPE valoarea caracterului in octal

6.9.5.26.9 FILETYPE

Aceasta comanda va seta tipul de fisiere pe care KERMIT le receptioneaza.

KER-11>SET FILE TYPE tip

6.9.5.26.9.1 ASCII

Tipul de fisier ASCII este folosit pentru fisiere test.

6.9.5.26.9.2 BINARY

Tipul de fisier BINARY este pentru fisiere care se doreste

sa fie transmise fara modificari de continut. Se pot insa pierde atribute de fisier. Parametrul PARITY trebuie setat pentru a se putea transmite fisiere binare.

6.9.5.26.10 HOME

Comanda SET HOME reseteaza perifericul si UIC la cel initial, pentru fisierele deschise sau create.

```
KER-11>SET HOME
```

6.9.5.26.11 LINE

Comanda SET LINE seteaza linia de terminal ca va fi folosita pentru conectare. Pentru aceasta trebuie sa existe dreptul de acces la acea linie. Formatul este:

```
KER-11>SET LINE TT55:        (pentru RSX)
```

6.9.5.26.12 LOGFILE

Comanda SET LOGFILE creeaza un fisier de depanare. Trebuie folosita inainte de orice comanda SET DEBUG.

```
KER-11>SET LOGFILE MYLOG.TXT
Created debug file MYLOG.TXT
KER-11>
```

6.9.5.26.13 PACKET-LENGTH

Se poate modifica lungimea pachetelor prin comanda SET PACKET-LENGTH. Aceasta comanda este utila daca pe linia de comunicatie sint multe erori.

```
KER-11>SET PACKET 60
```

6.9.5.26.14 PARITY

Prin aceasta se specifica tipul de paritate folosit. Initial este NONE.

```
KER-11>SET PARITY NONE
KER-11>SET PARITY ODD
KER-11>SET PARITY EVEN
KER-11>SET PARITY MARK
KER-11>SET PARITY SPACE
```

Generarea paritatii se face prin software si numai pe lungime de 8 biti.

6.9.5.26.15 PAUSE

Parametrul PAUSE specifica numarul de secunde care trebuie asteptat pina la trimiterea urmatorului pachet. Aceasta poate fi folositor la sisteme foarte incarcate. Acest parametru este cal-

culat automat in functie de yiteza liniei.

```
KER-11>SET PAUSE 1
```

6.9.5.26.16 PROMPT

Prin folosirea comenzii SET PROMPT, se poate schimba promptul "KER-11>" in altceva care sa indice sistemul cu care este conectat.

```
KER-11>SET PROMPT CORAL-87>
KER-11>SET PROMPT FUBAR>
```

6.9.5.26.17 RANDOM

Acesta comanda permite programului KER-11 sa genereze erori de checksum pentru testarea recuperarilor la erori. Este o comanda rar folosita.

```
KER-11>SET RANDOM ON
KER-11>SET RANDOM OFF
```

6.9.5.26.18 RECORD-FORMAT

Programul KERMIT creeaza fisiere RMS 11 cu inregistrari de lungime variabila si cu CR implicit pentru fisiere text. Acesta se poate modifica:

```
KER-11>SET RECORD-FORMAT STREAM
KER-11>SET RECORD-FORMAT VARIABLE
```

Aceasta comanda este bine sa fie data in fisierul KERMIT.INI, care se executa cind programul KERMIT porneste.

6.9.5.26.19 RETRY

Comanda SET RETRY seteaza numarul de reluari pina cind se renunta la transmiterea unui pachet neconfirmat. Acesta se foloseste daca exista erori pe linia de comunicatie sau sistemul este incarcat.

```
KER-11>SET RETRY 10
```

6.9.5.26.20 SPEED

Comanda SET SPEED seteaza viteza liniei de terminal specificata prin comanda SET LINE. Schimbarea vitezei se poate face numai de catre utilizatorii privilegiati. Are efect numai pentru linii de multiplexor.

```
KER-11>SET SPEED 1200
```

6.9.5.26.21 TIMEOUT

Valoarea de timeout indica cit timp sa se astepte pina la

obtinerea unui pachet de la celalalt KERMIT. Daca incarcarea sistemului este mare se poate mari aceasta valoare.

6.9.5.27 SHOW

Comanda SHOW va afisa setarile facute cu comanda SET si anumite statistici.

KER-11>SHOW parametru

6.9.5.27.1 ALL

Se afiseaza toate caracteristicile.

6.9.5.27.2 DEFAULT

Afiseaza perifericul si UIC folosit pentru operatii cu fisiere.

6.9.5.27.3 ESCAPE

Afiseaza caracterul de "escape" folosit pentru revenire la KER-11 local.

6.9.5.27.4 FILE-TYPE

Afiseaza tipul de fisier curent (BINARY sau ASCII).

6.9.5.27.5 LINE

Afiseaza parametrii asociati cu linia de conectare folosita.

6.9.5.27.6 PACKET

Afiseaza statistici legate de numarul de pachete manipulate.

6.9.5.27.7 PARAMETERS

Afiseaza diversi parametrii de send.

6.9.5.27.8 RECORD-FORMAT

Afiseaza ce fel de fisier va fi creat de catre KERMIT.

6.9.5.27.9 TIME

Afiseaza timpul curent.

6.9.5.27.10 Version

Afiseaza versiunea curenta.

6.9.5.28 STARTUP

Se poate pune un fisier denumit KERMIT.INI in contul curent de unde KER-11 va citi comenzile la pornire, inainte de a accepta comenzile de la terminal.

6.9.5.29 SYSTEM

Comanda SYSTEM trimite o linie de comanda MCR/DCL pentru a fi executata. Aceasta se face prin directiva SPWN$S.
Exemple:

```
KER-11>SYS PIP DP1:=FUBAR.DAT
KER-11>SYS
Command: SUB MYJOB
KER-11>
```

6.9.5.30 TAKE

Comanda TAKE este la fel ca si comanda @. Se deschide un fisier disc pentru citirea de comenzi care vor fi executate.

```
KER-11>TAKE MYKERM.CMD
```

6.9.5.31 TYPE

Comanda TYPE tipareste un fisier la terminal ca in:

```
KER-11>TYPE KERMIT.INI
```

6.9.6 Utilizare

Se descrie un exemplu de dialog intre un micro M118 conectat la un CORAL prin programul KERMIT.

```
A>b:
B>a:kermit
KERMIT-80 v4.05 configured for Tim-S/Plus vith VT52

KERMIT-80 B:>connect
[Connected to remote host, type Control-\C to return]

hel
Account or name 7,2
Password:

RSX-11M BL 32 [1,54] System RSX 11M

22-MAY-87 08:16 Logged on Terminal TT0:
Good Morning

>KER
Kermit2.17
KER-11>server
KERMIT Server running on PDP-11 host.  Please type
```

your escape sequence to return to your local machine. Shut down the server by typing the KERMIT BYE command on your local machine.

(the user now types Ctrl\C)

[Connection closed, back to micro]

KERMIT-80 B:>get *.odl

. Toate fisierele de pe mini de tip ODL sint trimise

. Urmatoarele informatii sint afisate pe ecran
. Numar de pachete: n
. Numar de reluari: m
. Nume de fisier transmis:

KERMIT-80 B:>bye
B>

6.9.7 Utilizare KERMIT sub CP/M

6.9.7.1 Descriere KERMIT-80

Deoarece KERMIT-80 ruleaza pe un micro single-user, are in permanenta controlul ecranului. Astfel, in timpul emisiei sau receptiei sint afisate permanent numarul total de reluari efectuate.

KERMIT-80 utilizeaza un time-out simultan pe cererile de intrare si poate intrerupe automat blocajele. De cele mai multe ori acest aspect nu este important deoarece host-ul este capabil sa gestioneze time-out-urile.

Daca, in pofida facilitatii de time-out, transmisia pare blocata (ceea ce se poate observa din faptul ca ecranul nu se modifica pentru un timp) tastarea unui RETURN (CR, ENTER) determina microcalculatorul sa faca ce ar fi facut in cazul unui time-out, adica sa transmita host-ului un NAK, pentru ca acesta sa retransmita ultimul pachet (sau, daca microcalculatorul e in emisie, sa retransmita el ultimul pachet).

E posibil ca transferurile micro/micro sau micro/mini sa necesite acest tip de interventie manuala.

Transferurile de fisiere pot fi intrerupte in citeva moduri:

-Control-C: intrerupe imediat transferul si se revine la nivelul de comenzi KERMIT-80, astfel ca utilizatorul se poate conecta la post sau poate lua orice alta decizie.

-Control-X: in timpul transmiterii unui fisier, Ctrl/X va termina transmiterea fisierului curent si va semnala KERMIT-ului receptor sa renunte la ce a primit. Daca exista mai multe fisiere de transmis, KERMIT-80 va trece la urmatorul. Daca e in receptie, KERMIT-80 va transmite un semnal KERMIT-ului de la distanta pentru a opri transmisia acestui fisier. Daca KERMIT-ul de pe host intelege acest semnal (nu toate implementarile pot acest lucru) se va conforma, astfel fisierul va continua sa soseasca. In orice caz, KERMIT-ul de la distanta va continua transmisia urmatorului fisier din grup (daca exista).

-Control-Z: la fel ca ctrl-X, cu exceptia faptului ca daca se transmite un grup de fisiere, ctrl-Z va termina transmisia intregului grup. Daca se transmite un singur fisier, are acelasi

efect ca si ctrl-X.

-**Carriage;** daca se tasteaza de mai multe ori carriage return **Return**, KERMIT-80 va retransmite pachetul curent de un numar de ori cel mult egal cu numarul maxim de transmisii (circa 16) si daca nu se primeste un raspuns valid, se revine la nivelul de comenzi KERMIT-80.

6.9.7.2 Comenzi KERMIT-80

KERMIT-80 utilizeaza un limbaj de comanda cu cuvinte cheie. Fiecare cuvint poate fi abreviat la lungimea minima unica. In orice punct se poate tasta "?" pentru a obtine un meniu al optiunilor disponibile pentru cimpul curent. In orice punct al unei comenzi se poate tasta ESC pentru a completa cuvintul cheie curent sau numele fisierului; daca nu au fost tastate suficiente caractere pentru a identifica in mod unic cimpul curent, KERMIT-80 va emite un semnal sonor si va permite continuarea din acel punct.

CONNECT

stabileste o conexiune de tip "terminal virtual" la orice host care poate fi conectat la un post serial, adica transmite toate caracterele introduse de la tastatura postului serial, si afiseaza pe ecran toate caracterele receptionate de la postul serial. De asemenea, anuleaza terminalul DEC VT 52 pentru a permite controlul cursorului, stergerea ecranului etc., daca switch-ul VT52-EMULATION este setat (vezi mai jos). La emiterea comenzii CONNECT, micro va tiparii un mesaj care indica modul de revenire in local. Secventa de "escape" consta dintr-un caracter (de regula mai putin utilizat) cum ar fi ctrl-\ sau ctrl-], urmat de o comanda dintr-o singura litera:

C	- inchide conexiunea, revine la nivelul de comenzi KERMIT-80;
S	- afiseaza starea conexiunii, dar mentine conexiunea;
?	- afiseaza comenzile de un caracter disponibile;
0 (zero)	- transmite un caracter null(0);
^ (sau **^]**)	- transmite caracterul de escape la host.

SEND specfis

transmite fisierul (fisierele) specificat (e) la KERMIT-ul de la distanta. Cuvintul **specfis** poate indica un grup de fisiere.

RECEIVE

receptioneaza fisierul (fisierele) de la KERMIT-ul de la distanta. Se inregistreza sub numele furnizate in antetele de fisier provenite de la host. Daca numele nu sint legale, utilizeaza atitea caractere legale din nume cit e posibil (vezi si FILE-WARNING mai jos). Daca exista conflicte si switch-ul FILE-WARNING este setat se avertizeaza utilizatorul si se incearca construirea unui nume unic.

GET specfis

cind KERMIT-80 converseaza cu un server din host se poate include

o specificare de fisier pentru a solicita serverului sa transmita fisiere, de exemplu:

```
GET DK2:[7,2] KER*.HLP
```

LOG specfis

pe durata unei conexiuni inregistreaza toate caracterele primite de la host intr-un fisier specificat. Buna functionare a acestei optiuni depinde de capacitatea host-ului de a exercita un control XON/XOFF, prin urmare nu este garantata o transcriere completa. Fisierul de "log" este inchis la inchiderea conexiunii prin tastarea secventei de "escape".

TRANSMIT specfis

transmite fisierul specificat sistemului aflat la celalalt capat al conexiunii, ca si cum ar fi tastat de la terminal, cite o linie la un moment dat. Nu se utilizeaza nici un fel de protocol KERMIT. Fiecare linie trebuie confirmata manual. Aceasta functie este utila pentru a transmite fisiere la sisteme care nu au un program KERMIT. In timpul transmisiei se poate tasta caracterul "escape" urmat de una din urmatoarele comenzi de o singura litera:

C - opreste transmisia;
R - retransmite linia precedenta.

BYE

daca e in curs o legatura cu un server al unui KERMIT de la distanta, comanda termina serverul, face "logout" si iese din KERMIT-80, revenind la nivelul de comenzi CP/M.

LOGOUT

acelasi efect ca BYE, dar ramine la nivel de comenzi KERMIT-80.

FINISH

acelasi efect ca LOGOUT dar termina server-ul fara a face "logout". Utilizatorul ramine la nivel comenzi KERMIT-80; urmatoarele comenzi CONNECT vor duce utilizatorul la nivel de comenzi sistem host.

SET parametru [valoare]

pozitioneaza parametrul specificat la valoarea specificata.

Pozitionari posibile:

WARNING ON (sau OFF)

avertizeaza utilizatorul asupra conflictelor de nume in timpul receptiei fisierelor de la host si incearca sa genereze un nume unic, prin modificarea numelui dat. Valoarea implicita este ON.

VT52-EMULATION ON (sau OFF)

in timpul conectarii ca terminal la un host controleaza daca micro emuleaza un VT52 sau ruleaza in mod "nativ". Valoarea implicita este ON.

LOCAL ECHO ON (sau OFF)

pentru conectarea la un host trebuie setat LOCAL-ECHO ON daca host-ul este half-duplex si OFF daca host-ul este full duplex. Valoarea implicita este OFF.

ESCAPE

modifica valoarea caracterului ESCAPE pentru conectarea terminalelor virtuale. KERMIT-80 emite un prompt si asteapta ca utilizatorul sa introduca noua valoare a caracterului ESCAPE.

BAUD

modifica valoarea ratei de transmisie a postului de comunicatie. Comanda este implementata numai pentru anumite sisteme (CUB, TPD, M118 cu SAI, Tim-S Plus). Dupa SET BAUD se afiseaza o lista de corespondenta intre literele alfabetului in ratele de transmisie selectabile. Utilizatorul va tasta litera corespunzatoare vitezei dorite.

PARITY

seteaza paritatea pentru caracterele de iesire la una din valorile: NOVE, MARK, EVEN sau ODD. In intrare, daca paritatea este NOVE, atunci bitul 8 este pastrat (bit de date), altfel este eliminat si ignorat. Paritatea se aplica atit la conectarea la terminal cit si la transferul de fisiere.

IBM ON (sau OFF)

permite transferul fisierelor la si de la un calculator IBM. Setarea parametrului determina KERMIT-80 sa astepte de la IBM caracterul "turnaround" (XON), sa ignore paritatea in intrare, sa adauge paritatea corespunzatoare in iesire si sa utilizeze ecou local in CONNECT. Valoare implicita este OFF.

BLOCK-CHECK-TYPE

Optiunile sint:

1-CHARACTER-CHECK SUM
Valoarea imlicita, suma de control pe 6 biti.

2-CHARACTER-CHECK SUM
Suma de control codificata ca doua caractere.

3-CHARACTER-CRC-CCITT
CRC (Cyclical Redundancy Check), CCITT pe 16 biti, codificat ca 3 caractere.

Optiunile pe 2 si 3 caractere trebuie sa fie folosite numai in conditii de zgomot foarte mare pe linie.

FILE-MODE

indica lui KERMIT-80 ce tip de fisier se transmite, astfel ca KERMIT sa poata determina corect sfirsitul de fisier. SET FILE BINARY inseamna transmiterea ultimului bloc in intregime: SET FILE ASCII se foloseste pentru transmiterea de fisiere text, iar transmisia se opreste cind se intilneste primul Control-Z, oriunde in fisier. Daca se transmite un fisier text in mod binar, e posibil ca la sfirsitul fisierului pe sistemul destinatie sa apara unele caractere suplimentare (cel mult 127). Daca se trans-

mite un fisier binar in mod ASCII, se poate intimpla sa nu se transmita intregul fisier daca se intilnesc bytes al caror cod corespunde lui Control-Z.

DEFAULT-DISK

permite specificarea discului implicit ca sursa si destinatie ale transferurilor de fisiere. In plus, emiterea acestei comenzi implica comutarea utilizatorului pe discul specificat. Discul selectat apare in promtul KERMIT, de exemplu:

KERMIT-80 A:>

PORT

permite comutarea intre diferite porturi de comunicatie. Comanda nu este disponibila pe toate sistemele (doar pe cele dotate cu SAI).

PRINTER ON (sau OFF)

daca parametrul este ON, in timpul unei sesiuni CONNECT se face o copie la imprimanta a caracterelor receptionate. Nu se face buffer-are sau control de flux, ci se presupune ca imprimanta poate tipari in ritmul transmisiei.

DIR

listeaza numele fisierelor solicitate (implicit toate). Sint incluse si dimensiunile fisiserelor in Kiloocteti. Tastarea oricarui caracter are ca efect intreruperea listarii. Afisarea (chiar daca este intrerupta) se termina cu indicarea spatiului liber pe disc.

ERA

sterge fisierul (fisierele) specificate. Numele fisierelor sterse nu sint afisate.

```
;7 Integrala programelor sursa ale principalelor componente ale
;  nucleului sistemului de operare CP/M
;  7.1 Componenta CCP
;  7.2 Componenta BDOS
;  7.3 Componenta BIOS
;       7.3.1 MainBios
;       7.3.2 ShadowBios
```

```
                        ;    7 Integrala programelor sursa ale principalelor componente
                        ;      ale nucleului sistemului de operare CP/M

                        ;  ====================
                        ;  ! 7.1 Componenta CCP !
                        ;  ====================
                        ;

0005                    BDOS    EQU     5

                                ORG     0DE00H

DE00'                   CCPBASE:
DE00'  C3 E15C'                 JP      CSTART
DE03'  C3 E158'                 JP      REDOST
                        ;Zona tampon de intrare conform cu cerintele BDOS
                        ;
DE06'  7F               BUFMAX: DB      127
                        ;Numar de octeti dat de functia BDOS READ INPUT BUFFER
                        ;
DE07'  00               BUFLEN: DB      0
                        ;Aici incepe de fapt
                        ;
DE08'  20 20 20 20      CINPBUF:DB      '                COPYRIGHT (C) 1979, DIGITAL'
DE0C'  20 20 20 20
DE10'  20 20 20 20
DE14'  20 20 20 20
DE18'  43 4F 50 59
DE1C'  52 49 47 48
DE20'  54 20 28 43
DE24'  29 20 31 39
DE28'  37 39 2C 20
DE2C'  44 49 47 49
DE30'  54 41 4C
DE33'  20 52 45 53              DB      ' RESEARCH '
DE37'  45 41 52 43
DE3B'  48 20 20
DE3E'                           DS      74              ;
                        ;Indicator pentru zona tampon de intrare
                        ;
DE88'                   BUFPOINT:
DE88'  DE08'                    DW      CINPBUF
DE8A'  0000             POINSKP:DW      0               ;indicator in zona tampon
                                                        ;de intrare. Sare spatii
                        ;       *****   ******
                        ;Rutina trimite catre consola caracterul din registrul A
                        ;
DE8C'                   CCONOUT:
DE8C'  5F                       LD      E,A             ;Caracter in registrul E
DE8D'  0E 02                    LD      C,2             ;Functia in registrul C
DE8F'  C3 0005                  JP      BDOS            ;Functie de intrare in BDOS
                        ;       *****   ******
DE92'                   COSVB:
DE92'  C5                       PUSH    BC              ;Salveaza registrul B,C in stiva
DE93'  CD DE8C'                 CALL    CCONOUT         ;Afiseaza registrul A
DE96'                   RESTB:
DE96'  C1                       POP     BC              ;Reface registrul B,C
DE97'  C9                       RET
                        ;       *****   ******
DE98'                   CCRLF:                          ;Afiseaza "CARRIAGE RETURN" si
DE98'  3E 0D                    LD      A,13            ;"LINE FEED"
```

```
DE9A'  CD DE92'          CALL   COSVB
DE9D'  3E 0A             LD     A,10
DE9F'  C3 DE92'          JP     COSVB
                         ;      *****  ******
DEA2'                    BLANCK:                  ;Afiseaza spatiu
DEA2'  3E 20             LD     A,' '
DEA4'  C3 DE92'          JP     COSVB
                         ;      *****  ******
DEA7'                    NLMESS:                  ;Afiseaza "CRLF"si apoi mesaj di
DEA7'  C5                PUSH   BC                ;B,C
DEA8'  CD DE98'          CALL   CCRLF
DEAB'  E1                POP    HL
                         ;      *****  ******
DEAC'                    MESSAGE:                 ;Afiseaza mesaj din H,L
DEAC'  7E                LD     A,(HL)
DEAD'  B7                OR     A
DEAE'  C8                RET    Z
DEAF'  23                INC    HL
DEB0'  E5                PUSH   HL
DEB1'  CD DE8C'          CALL   CCONOUT
DEB4'  E1                POP    HL
DEB5'  C3 DEAC'          JP     MESSAGE
                         ;      *****  ******
DEB8'                    DOSINIT:                 ;Functia BDOS "INIT"
DEB8'  0E 0D             LD     C,13
DEBA'  C3 0005           JP     BDOS
                         ;      *****  ******
DEBD'                    CSELDSK:
DEBD'  5F                LD     E,A
DEBE'  0E 0E             LD     C,14
DEC0'  C3 0005           JP     BDOS
DEC3'                    BDRES:                   ;Apeleaza BDOS si memoreaza
DEC3'  CD 0005           CALL   BDOS              ;rezultat
DEC6'  32 E5EE'          LD     (DIRENTRY),A
DEC9'  3C                INC    A
DECA'  C9                RET
                         ;      *****  ******
DECB'                    COPEN:                   ;Deschide fisier. D,E
DECB'  0E 0F             LD     C,15
DECD'  C3 DEC3'          JP     BDRES
                         ;      *****  ******
DED0'                    COMOPEN:                 ;Deschide fisier din zona de
DED0'  AF                XOR    A                 ;comanda
DED1'  32 E5ED'          LD     (COMCR),A
DED4'  11 E5CD'          LD     DE,COMFCB
DED7'  C3 DECB'          JP     COPEN
                         ;      *****  ******
DEDA'                    CCLOSE:                  ;Inchide fisier
DEDA'  0E 10             LD     C,16
DEDC'  C3 DEC3'          JP     BDRES
DEDF'                    SRCHFIRST:               ;Cauta primul punct de intrare
DEDF'  0E 11             LD     C,17              ;lista "DIRECTORY"
DEE1'  C3 DEC3'          JP     BDRES
                         ;      *****  ******
DEE4'                    SRCHNEXT:                ;Cauta urmatorul punct de intra
DEE4'  0E 12             LD     C,18              ;in lista "DIRECTORY"
DEE6'  C3 DEC3'          JP     BDRES
                         ;      *****  ******
DEE9'                    COMSRCH:                 ;Cauta fisier din zona de coma
DEE9'  11 E5CD'          LD     DE,COMFCB
DEEC'  C3 DEDF'                 SRCHFIRST
```

```
                        ;       *****   ******
DEEF'                   ERAFILE:                        ;Sterge fisier
DEEF'   0E 13                   LD      C,19
DEF1'   C3 0005                 JP      BDOS
DEF4'                   BDFLAG:                         ;Apeleaza BDOS si pune IND
DEF4'   CD 0005                 CALL    BDOS
DEF7'   B7                      OR      A
DEF8'   C9                      RET
                        ;       *****   ******
DEF9'                   SEQREAD:                        ;Citeste inregistrare din fisie-
DEF9'   0E 14                   LD      C,20            ;rul deschis
DEFB'   C3 DEF4'                JP      BDFLAG
                        ;       *****   ******
DEFE'                   COMREAD:                        ;Citeste inregistrare din coman-
DEFE'   11 E5CD'                LD      DE,COMFCB       ;da FCB
DF01'   C3 DEF9'                JP      SEQREAD
                        ;       *****   ******
DF04'                   WRITE:                          ;Scrie inregistrare in fisier
DF04'   0E 15                   LD      C,21
DF06'   C3 DEF4'                JP      BDFLAG
                        ;       *****   ******
DF09'                   CREATE:                         ;Creaza fisier D,E
DF09'   0E 16                   LD      C,22
DF0B'   C3 DEC3'                JP      BDRES
                        ;       *****   ******
DF0E'                   CRENAME:                        ;Redenumeste fisier
DF0E'   0E 17                   LD      C,23
DF10'   C3 0005                 JP      BDOS
                        ;       *****   ******
DF13'                   GETUSER:                        ;Ia numar utilizator
DF13'   1E FF                   LD      E,255
                        ;       *****   ******
DF15'                   PUTUSER:                        ;Pune numar utilizator
DF15'   0E 20                   LD      C,' '
DF17'   C3 0005                 JP      BDOS
                        ;       *****   ******
DF1A'                   USSAV:                          ;Ia utilizator si salveaza cu
DF1A'   CD DF13'                CALL    GETUSER         ;drive conectat la locatia 4
DF1D'                   SAVUSER: FC09'                      BCONIN  EQU     $+9
FC0C'                   BCONOUT EQU     $+0CH
FC0F'                   LISTOUT EQU     $+0FH
FC12'                   PUNCH   EQU     $+12H
FC15'                   BRDER   EQU     $+15H
FC18'                   BHOME   EQU     $+18H
FC1B'                   BSELDSK EQU     $+1BH
FC1E'                   BSETTRK EQU     $+1EH
FC21'                   BSETSEC EQU     $+21H
```

```
FC24'                   LD      A,(CLOGDSK)     ;locatia 4
DF2C'  32 0004                  LD      (4),A
DF2F'  C9                       RET
                        ;       *****   ******
DF30'                   CAPSTR:                 ;Transforma in litere mari
DF30'  FE 61                    CP      97
DF32'  D8                       RET     C
DF33'  FE 7B                    CP      123
DF35'  D0                       RET     NC
DF36'  E6 5F                    AND     95
DF38'  C9                       RET
                        ;       *****   ******
DF39'                   LINEINP:                ;Ia linia de comanda de la consola
                                                ;sau din fisierul de "SUBMIT"-are
DF39'  3A E5AB'                 LD      A,(SUBSWITCH);Incarca si testeaza indicatorii de
DF3C'  B7                       OR      A       ;SUBMIT. Daca este <>0 incearca
DF3D'  CA DF96'                 JP      Z,NOSUB ;citire linie din fisierul $$$.SUB,
DF40'  3A E5EF'                 LD      A,(CLOGDSK) ;altfel de la consola. Testeaza
DF43'  B7                       OR      A       ;daca discul conectat este"A".
DF44'  3E 00                    LD      A,0     ;Fisierul $$$.SUB trebuie sa fie pe
DF46'  C4 DEBD'                 CALL    NZ,CSELDSK ;discul "A". Daca nu, atunci
DF49'  11 E5AC'                 LD      DE,SUBFCB ;selecteaza discul A
DF4C'  CD DECB'                 CALL    COPEN   ;Deschide fisierul de"SUBMIT-are
DF4F'  CA DF96'                 JP      Z,NOSUB ;Daca nu exista fisier, ia linia
                                                ;de la consola
DF52'  3A E5BB'                 LD      A,(SUBRC) ;Incarca numar de inregistrare din
DF55'  3D                       DEC     A       ;FCb si decrementeaza; fisierul
DF56'  32 E5CC'                 LD      (SUBCR),A ;SUBMIT se va goli, inregistrare cu
DF59'  11 E5AC'                 LD      DE,SUBFCB ;inregistrare
DF5C'  CD DEF9'                 CALL    SEQREAD ;Ia o linie de comanda
DF5F'  C2 DF96'                 JP      NZ,NOSUB ;Fisier epuizat
DF62'  11 DE07'                 LD      DE,BUFLEN ;Spre zona tampon
DF65'  21 0080                  LD      HL,128  ;Din zona DMA
DF68'  06 80                    LD      B,128   ;Numar de octeti
DF6A'  CD E242'                 CALL    MOVDHB  ;Transfera linia de comanda
DF6D'  21 E5BA'                 LD      HL,FCBS2 ;Incarca S2 din FCB
DF70'  36 00                    LD      (HL),0  ;Si pune pe zero
DF72'  23                       INC     HL      ;Selecteaza CR in fisierul SUBMIT
DF73'  35                       DEC     (HL)    ;FCB. Decrementeaza CR
DF74'  11 E5AC'                 LD      DE,SUBFCB ;Dupa ce s-a luat o linie, lungimea
DF77'  CD DEDA'                 CALL    CCLOSE  ;fiserului este decrementata. Fisier
DF7A'  CA DF96'                 JP      Z,NOSUB ;inchis. Incercare de inchidere
DF7D'  3A E5EF'                 LD      A,(CLOGDSK) ;nereusita. Reselecteaza discul
DF80'  B7                       OR      A       ;conectat schimbat in timpul
DF81'  C4 DEBD'                 CALL    NZ,CSELDSK ;secventei de SUBMIT-are. Daca nu,
DF84'  21 DE08'                 LD      HL,CINPBUF ;disc "A "deja selectat. Afiseaza
DF87'  CD DEAC'                 CALL    MESSAGE ;linia "SUBMIT"-ata. La consola
DF8A'  CD DFC2'                 CALL    CNSTCI  ;sistem. Testeaza daca este caracter
DF8D'  CA DFA7'                 JP      Z,STTRAN ;Daca nu, inspecteaza linia
DF90'  CD DFDD'                 CALL    ERASUB  ;Altfel sterge secventa de SUBMIT
DF93'  C3 E182'                 JP      RESUME  ;si reintra in comanda
DF96'                   NOSUB:                  ;Nu este fisier SUBMIT sau epuizat
DF96'  CD DFDD'                 CALL    ERASUB  ;Sterge fisier SUBMIT $$$.SUB
DF99'  CD DF1A'                 CALL    USSAV   ;Ia si salveaza nume utilizator
DF9C'  0E 0A                    LD      C,10    ;Codul functiei "READ BUFFER"
DF9E'  11 DE06'                 LD      DE,BUFMAX ;Adresa zona tampon
DFA1'  CD 0005                  CALL    BDOS    ;Citeste zona tampon de intrare
DFA4'  CD DF29'                 CALL    STLOGG  ;Memoreaza disc conectat(nume
```

```
                                        ;utilizator inlaturat)
DFA7'                   STTRAN:         ;Inspecteaza linia si schimba in litere mari
DFA7'  21 DE07'                 LD      HL,BUFLEN       ;Incarca numarator octeti
DFAA'  46                       LD      B,(HL)          ;in registrul B
DFAB'                   LOOPCAPS:
DFAB'  23                       INC     HL              ;Indicator caracter urmator
DFAC'  78                       LD      A,B             ;Testeaza daca mai sint caractere
DFAD'  B7                       OR      A
DFAE'  CA DFBA'                 JP      Z,CAPSEND       ;Daca nu, termina schimbare
DFB1'  7E                       LD      A,(HL)          ;Incarca caracter
DFB2'  CD DF30'                 CALL    CAPSTR          ;Schimba
DFB5'  77                       LD      (HL),A          ;Pune inapoi in zona tampon de
DFB6'  05                       DEC     B               ;intrare si decrementeaza numar
DFB7'  C3 DFAB'                 JP      LOOPCAPS        ;din nou
DFBA'                   CAPSEND:
DFBA'  77                       LD      (HL),A          ;Memoreaza un zero urmator
DFBB'  21 DE08'                 LD      HL,CINPBUF      ;Incarca si memoreaza
DFBE'  22 DE88'                 LD      (BUFPOINT),HL   ;adresa zona tampon
DFC1'  C9                       RET
                        ;       *****   ******
DFC2'                   CNSTCI:                 ;Testeaza daca este caracter la consola
DFC2'  0E 0B                    LD      C,11    ;Rezultatul este zero daca nu este
DFC4'  CD 0005                  CALL    BDOS    ;caracter. Altfel caracterul introdus
DFC7'  B7                       OR      A       ;pentru stare consola
DFC8'  C8                       RET     Z       ;Return daca nu este caracter
DFC9'  0E 01                    LD      C,1     ;Citeste caracter
DFCB'  CD 0005                  CALL    BDOS    ;de la consola si pune
DFCE'  B7                       OR      A       ;indicatorii de conditie
DFCF'  C9                       RET
                        ;       *****   ******
DFD0'                   LOGGIN:                         ;returneaza discul logic
DFD0'  0E 19                    LD      C,25
DFD2'  C3 0005                  JP      BDOS
                        ;       *****   ****X*
DFD5'                   SDMAD:                          ;Pune adresa DMA la 80H
DFD5'  11 0080                  LD      DE,128
                        ;       *****   ******
DFD8'                   SETDMA:                         ;Pune adresa DMA specificata
DFD8'  0E 1A                    LD      C,26
DFDA'  C3 0005                  JP      BDOS
                        ;       *****   ******
DFDD'                   ERASUB:                 ;Sterge fisier SUBMIT, daca trebuie
DFDD'  21 E5AB'                 LD      HL,SUBSWITCH    ;Testeaza activitate de SUBMIT-are
DFE0'  7E                       LD      A,(HL)          ;Daca nu, Return
DFE1'  B7                       OR      A               ;Nu e necesara stergerea
DFE2'  C8                       RET     Z
DFE3'  36 00                    LD      (HL),0          ;Pune comutator de SUBMIT inactiv
DFE5'  AF                       XOR     A
DFE6'  CD DEBD'                 CALL    CSELDSK         ;Selecteaza drive "A"si sterge
DFE9'  11 E5AC'                 LD      DE,SUBFCB       ;fisierul de SUBMIT-are
DFEC'  CD DEEF'                 CALL    ERAFILE
DFEF'  3A E5EF'                 LD      A,(CLOGDSK)     ;Reselecteaza discul conectat,
DFF2'  C3 DEBD'                 JP      CSELDSK         ;schimbat la stergere
                        ;       *****   ******
DFF5'                   VRSVER:                 ;Testeaza daca versiunea CCP
DFF5'  11 E128'                 LD      DE,CCPVER       ;este compatibila cu
DFF8'  21 E600'                 LD      HL,DOSVER       ;versiunea BDOS
DFFB'  06 06                    LD      B,6             ;Lungime descriptor
DFFD'                   VERLOOP:
DFFD'  1A                       LD      A,(DE)          ;Incarcare octet
DFFE'  BE                       CP      (HL)            ;compara
```

```
DFFF'  C2 E1CF'          JP    NZ,FATAL     ;Daca nu, eroare fatala
E002'  13                INC   DE           ;Urmator CCP
E003'  23                INC   HL           ;Urmator BDOS
E004'  05                DEC   B            ;Decrementeaza numar
E005'  C2 DFFD'          JP    NZ,VERLOOP   ;Din nou
E008'  C9                RET
                ;        ***** ******
E009'                    ERROR:             ;Rutina eroare
E009'  CD DE98'          CALL  CCRLF        ;Primul Return, Line Feed
E00C'  2A DE8A'          LD    HL,(POINSKP) ;Incarca indicator zona tampon
E00F'                    LNERR:             ;Afiseaza secventa de caractere
E00F'  7E                LD    A,(HL)       ;eronata
E010'  FE 20             CP    ' '          ;Testeaza sfirsit
E012'  CA E022'          JP    Z,QMARK
E015'  B7                OR    A
E016'  CA E022'          JP    Z,QMARK      ;Termina cu semnul intrebarii
E019'  E5                PUSH  HL
E01A'  CD DE8C'          CALL  CCONOUT      ;Daca nu e gata, afiseaza
E01D'  E1                POP   HL           ;caracter
E01E'  23                INC   HL           ;Caracteul urmator
E01F'  C3 E00F'          JP    LNERR        ;Din nou
E022'                    QMARK:             ;
E022'  3E 3F             LD    A,'?'        ;Incarca semnul intrebarii
E024'  CD DE8C'          CALL  CCONOUT      ;si afiseaza
E027'  CD DE98'          CALL  CCRLF
E02A'  CD DFDD'          CALL  ERASUB       ;Daca este eroare in SUBMIT
E02D'  C3 E182'          JP    RESUME       ;sterge $$$.SUB si reia comanda
                ;        ***** ******
E030'                    DELIM:             ;Test daca e delimitator valid
E030'  1A                LD    A,(DE)       ;Incarca caracter
E031'  B7                OR    A            ;Nu mai sint
E032'  C8                RET   Z
E033'  FE 20             CP    ' '          ;Test daca se poate afisa
E035'  DA E009'          JP    C,ERROR      ;Eroare daca nu
E038'  C8                RET   Z            ;Delimitatorii valizi sint:
E039'  FE 3D             CP    '='          ;spatiu, egal, sageata inapoi
E03B'  C8                RET   Z            ;punct, punct si virgula
E03C'  FE 5F             CP    95           ;"L" si ">"
E03E'  C8                RET   Z            ;La rezultat subrutina
E03F'  FE 2E             CP    '.'          ;intoarce indicatorul de zero
E041'  C8                RET   Z            ;pe 1 pentru orice delimitator
E042'  FE 3A             CP    ':'          ;valid
E044'  C8                RET   Z
E045'  FE 3B             CP    ';'
E047'  C8                RET   Z
E048'  FE 3C             CP    '<'
E04A'  C8                RET   Z
E04B'  FE 3E             CP    '>'
E04D'  C8                RET   Z
E04E'  C9                RET
                ;        ***** ******
E04F'                    SKPBLK:            ;Sare spatiile premergatoare
E04F'  1A                LD    A,(DE)
E050'  B7                OR    A,           ;Mai sint caractere ?
E051'  C8                RET   Z
E052'  FE 20             CP    ' '          ;Test daca e spatiu
E054'  C0                RET   NZ
E055'  13                INC   DE           ;Caracterul urmator
E056'  C3 E04F'          JP    SKPBLK
                ;        ***** ******
E059'                    ADDHL:             ;Aduna registru A la H,L
```

```
E059'  85                      ADD     A,L
E05A'  6F                      LD      L,A     ;Muta octetul CMPS (Cel Mai Putin
E05B'  D0                      RET     NC      ;Semnificativ) in registrul L. CARRY ?
E05C'  24                      INC     H       ;Daca este CARRY incrementeaza
E05D'  C9                      RET             ;registrul H
                       ;       *****   ******
                       ;Inspecteaza zona tampon de intrare si muta in zona de comanda
                       ;Incepe cu prima pozitie din zona de comanda
E05E'                  TRABUF:
E05E'  3E 00                   LD      A,0
                       ;       *****   ******
                       ;Incepe de la pozitia specificata din zona de comanda
E060'                  TRB01:
E060'  21 E5CD'                LD      HL,COMFCB       ;Zona de comanda
E063'  CD E059'                CALL    ADDHL           ;Aduna deplasament
E066'  E5                      PUSH    HL
E067'  E5                      PUSH    HL              ;Salveaza in stiva
E068'  AF                      XOR     A
E069'  32 E5F0'                LD      (SPECDRIVE),A   ;Selecteaza drive-ul implicit
E06C'  2A DE88'                LD      HL,(BUFPOINT)   ;Incarca adresa zona tampon
E06F'  EB                      EX      DE,HL
E070'  CD E04F'                CALL    SKPBLK          ;Sare spatii
E073'  EB                      EX      DE,HL
E074'  32 DE8A'                LD      (POINSKP),A     ;Salveaza adresa
E077'  EB                      EX      DE,HL           ;D,E primul indicator diferit de
E078'  E1                      POP     HL              ;spatiu. Zona de comanda
E079'  1A                      LD      A,(DE)          ;Incarca caracter din zona tampon
E07A'  B7                      OR      A
E07B'  CA E089'                 JP      Z,NOSPEC        ;ZERO, nu este drive specificat
E07E'  DE 40                   SBC     A,'@'           ;Poate fi un drive; scade litera
E080'  47                      LD      B,A             ;de start si salveaza nr.in reg B
E081'  13                      INC     DE              ;Drive specificat daca urmatorul
E082'  1A                      LD      A,(DE)          ;caracter este":". Incarca si
E083'  FE 3A                   CP      ':'             ;testeaza
E085'  CA E090'                 JP      Z,SPCDRV        ;Este OK. Daca nu
E088'  1B                      DEC     DE              ;inapoi la primul caracter
E089'                  NOSPEC:                         ;Nu a fost specificat vreun drive
E089'  3A E5EF'                LD      A,(CLOGDSK)     ;Incarca disc conectat
E08C'  77                      LD      (HL),A          ;Salveaza zona de comanda
E08D'  C3 E096'                JP      FNSTART         ;Salt peste specificatia de drive
E090'                  SPCDRV:
E090'  78                      LD      A,B             ;Drive-ul specificat este memorat
E091'  32 E5F0'                LD      (SPECDRIVE),A   ;atit in zona tampon
E094'  70                      LD      (HL),B          ;cit si in zona de comanda
E095'  13                      INC     DE      ;Primul caracter dupa specificatia de drive
E096'                  FNSTART:                        ;Numele fisierului incepe aici
E096'  06 08                   LD      B,8             ;Nume lungime
E098'                  FN1:
E098'  CD E030'                CALL    DELIM           ;Test daca e delimitator
E09B'  CA E0B9'                JP      Z,FN5           ;Daca e nume gata
E09E'  23                      INC     HL              ;Urmatorul caracter
E09F'  FE 2A                   CP      '*'             ;Test daca e "*"
E0A1'  C2 E0A9'                JP      NZ,FN2          ;Daca e "*" umple nume cu ???????
E0A4'  36 3F                   LD      (HL),'?'        ;Zona de comanda=???...
E0A6'  C3 E0AB'                JP      FN3
E0A9'                  FN2:
E0A9'  77                      LD      (HL),A          ;Muta caracter in zona de comanda
E0AA'  13                      INC     DE              ;Urmatorul in zona tampon-intrare
E0AB'                  FN3:
E0AB'  05                      DEC     B               ;Mai sint caractere in numele de
E0AC'  C2 E098'                JP      NZ,FN1          ;fisier. Din nou
```

```
E0AF'                   FN4:
E0AF'  CD E030'                 CALL    DELIM           ;Test daca e delimitator
E0B2'  CA E0C0'                 JP      Z,FN6           ;Salt la extensie
E0B5'  13                       INC     DE              ;Urmatorul caracter de intrare
E0B6'  C3 E0AF'                 JP      FN4             ;Cauta primul delimitator
E0B9'                   FN5:
E0B9'  23                       INC     HL  ;Delimitator intilnit inainte de 8 caractere
E0BA'  36 20                    LD      (HL),' '        ;Umple zona ramasa din numele de
E0BC'  05                       DEC     B               ;fisier cu spatii
E0BD'  C2 E0B9'                 JP      NZ,FN5
E0C0'                   FN6:
E0C0'  06 03                    LD      B,3             ;Aici incepe extensia ?
E0C2'  FE 2E                    CP      '.'             ;E intr-adevar inceput  extensie
E0C4'  C2 E0E9'                 JP      NZ,FN11
E0C7'  13                       INC     DE
E0C8'                   FN7:
E0C8'  CD E030'                 CALL    DELIM           ;Test daca e delimitator
E0CB'  CA E0E9'                 JP      Z,FN11          ;Sfirsit extensie
E0CE'  23                       INC     HL
E0CF'  FE 2A                    CP      '*'             ;Semnul "*"
E0D1'  C2 E0D9'                 JP      NZ,FN8          ;Umple extensia cu ???
E0D4'  36 3F                    LD      (HL),'?'
E0D6'  C3 E0DB'                 JP      FN9
E0D9'                   FN8:
E0D9'  77                       LD      (HL),A
E0DA'  13                       INC     DE              ;Urmatorul
E0DB'                   FN9:
E0DB'  05                       DEC     B               ;Mai sint caractere
E0DC'  C2 E0C8'                 JP      NZ,FN7
E0DF'                   FN10:
E0DF'  CD E030'                 CALL    DELIM           ;Inainte pina la primul
E0E2'  CA E0F0'                 JP      Z,FN12          ;delimitator
E0E5'  13                       INC     DE
E0E6'  C3 E0DF'                 JP      FN10
E0E9'                   FN11:                           ;Extensie mai scurta de 3
E0E9'  23                       INC     HL              ;caractere
E0EA'  36 20                    LD      (HL),' '        ;Umple cu spatii
E0EC'  05                       DEC     B
E0ED'  C2 E0E9'                 JP      NZ,FN11
E0F0'                   FN12:
E0F0'  06 03                    LD      B,3
E0F2'                   FN13:
E0F2'  23                       INC     HL              ;Umple zona de comanda cu
E0F3'  36 00                    LD      (HL),0          ;ultimele 3 zerouri (0) pentru
E0F5'  05                       DEC     B               ;EX, S1, S2
E0F6'  C2 E0F2'                 JP      NZ,FN13
E0F9'  EB                       EX      DE,HL
E0FA'  22 DE88'                 LD      (BUFPOINT),HL   ;Salveaza noul indicator in zona
E0FD'  E1                       POP     HL              ;tampon de intrare. Zona de
E0FE'  01 000B                  LD      BC,11           ;comanda. Reg.B=0,
                                                        ;reg.C=nume+lung.extensie
E101'                   FN14:
E101'  23                       INC     HL              ;Test daca este ?
E102'  7E                       LD      A,(HL)          ;in zona de comanda
E103'  FE 3F                    CP      '?'
E105'  C2 E109'                 JP      NZ,FN15
E108'  04                       INC     B               ;Incrementeaza numarul de ?
E109'                   FN15:
E109'  0D                       DEC     C
E10A'  C2 E101'                 JP      NZ,FN14
E10D'  78                       LD      A,B
```

```
E10E'  B7                      OR      A               ;Reg A numar de ? intilnite
E10F'  C9                      RET
                          ;Numele de comenzi sint dupa cum urmeaza
E110'                     COMTAB:
E110'  44 49 52 20             DB      'DIR ERA TYPESAVEREN USER'
E114'  45 52 41 20
E118'  54 59 50 45
E11C'  53 41 56 45
E120'  52 45 4E 20
E124'  55 53 45 52

                          ;Descriptorii de versiune
E128'                     CCPVER:
E128'  F9 16 00 00             DB      249,22,0,0;0,26
E12C'  00 1A

                          ;       *****   ******
                          ;Afla daca zona de comanda specifica o functie rezidenta
E12E'                     PICKCOM:
E12E'  21 E110'                LD      HL,COMTAB       ;Zona de comanda
E131'  0E 00                   LD      C,0             ;Numar comanda
E133'                     PICK1:
E133'  79                      LD      A,C             ;Maximum 6 comenzi
E134'  FE 06                   CP      6
E136'  D0                      RET     NC              ;Nu mai sint intrari in COMTAB
E137'  11 E5CE'                LD      DE,FILNAME      ;Nume fisier in zona de comanda
E13A'  06 04                   LD      B,4             ;Lungime comanda
E13C'                     PICK2:
E13C'  1A                      LD      A,(DE)          ;incarca caracter din nume fisier
E13D'  BE                      CP      (HL)            ;Compara cu COMTAB
E13E'  C2 E14F'                JP      NZ,PICK3        ;Nu coincide
E141'  13                      INC     DE
E142'  23                      INC     HL              ;Urmatorul
E143'  05                      DEC     B               ;Mai ?
E144'  C2 E13C'                JP      NZ,PICK2
E147'  1A                      LD      A,(DE)          ;Test de coicidenta cind numele
E148'  FE 20                   CP      ' '             ;fisier e intr-adevar gata
E14A'  C2 E154'                JP      NZ,PICK4
E14D'  79                      LD      A,C             ;Intoarce identificator comanda
E14E'  C9                      RET
E14F'                     PICK3:
E14F'  23                      INC     HL              ;Necoincidenta. Pregateste
E150'  05                      DEC     B               ;urmatoarea intrare
E151'  C2 E14F'                JP      NZ,PICK3
E154'                     PICK4:                       ;Urmatoarea intrare
E154'  0C                      INC     C
E155'  C3 E133'                JP      PICK1
E158'                     REDOST:
E158'  AF                      XOR     A               ;Numar caractere in zona tampon
E159'  32 DE07'                LD      (BUFLEN),A      ;pus pe zero
                          ;Aici se intra dupa orice start "CALD" sau "RECE"
                          ;
E15C'                     CSTART:
E15C'  31 E5AB'                LD      SP,CLOCSTACK    ;Pune indicatorul de stiva local
E15F'  C5                      PUSH    BC              ;Registrul C=
E160'  79                      LD      A,C             ;disc conectat+utilizator * 16
E161'  1F                      RRA
E162'  1F                      RRA
E163'  1F                      RRA
E164'  1F                      RRA
E165'  E6 0F                   AND     15              ;Remuta discul si salveaza codul
E167'  5F                      LD      E,A             ;Utilizator in registrul C
E168'  CD DF15'                CALL    PUTUSER         ;Pune cod utilizator
```

```
E16B'  CD DEB8'          CALL   DOSINIT     ;Initializeaza DOS selecteaza disc si
                                            ;pune tabela alocare si verifica
                                            ;vector pentru disc A
E16E'  32 E5AB'          LD     (SUBSWITCH),A  ;BDOS intoarce un rezultat
                                            ;diferit de 0 daca e un fisier
                                            ;care incepe cu $. Acest fisier
                                            ;poate fi SUBMIT-at
E171'  C1                POP    BC          ;Restaureaza disc si cod utilizator
E172'  79                LD     A,C
E173'  E6 0F             AND    15          ;Indeparteaza cod utilizator
E175'  32 E5EF'          LD     (CLOGDSK),A ;si memoreaza pe discul conectat
E178'  CD DEBD'          CALL   CSELDSK     ;Selecteaza discul specificat
E17B'  3A DE07'          LD     A,(BUFLEN)  ;Incarca si testeaza un caracter
E17E'  B7                OR     A           ;in zona tampon
E17F'  C2 E198'          JP     NZ,GETCOM   ;Daca e, fugi la linia de comanda
E182'                RESUME:
E182'  31 E5AB'          LD     SP,CLOCSTACK    ;Indicator de stiva
E185'  CD DE98'          CALL   CCRLF           ;Afiseaza CRLF
E188'  CD DFD0'          CALL   LOGGIN          ;Adreseaza discul conectat...
E18B'  C6 41             ADD    A,'A'
E18D'  CD DE8C'          CALL   CCONOUT
E190'  3E 3E             LD     A,'>'
E192'  CD DE8C'          CALL   CCONOUT         ;si afiseaza mesaj prompt
E195'  CD DF39'          CALL   LINEINP         ;Citeste zona tampon de intrare
                                                ;de la consola
E198'                GETCOM:
E198'  11 0080           LD     DE,128
E19B'  CD DFD8'          CALL   SETDMA          ;Pune adresa DMA la 80H
E19E'  CD DFD0'          CALL   LOGGIN          ;Adreseaza discul conectat
E1A1'  32 E5EF'          LD     (CLOGDSK),A     ;si salveaza
E1A4'  CD E05E'          CALL   TRABUF          ;Inspecteaza si translateaza
E1A7'  C4 E009'          CALL   NZ,ERROR        ;E semnul ? Daca da, eroare
E1AA'  3A E5F0'          LD     A,(SPECDRIVE)   ;Incarca drive specificat
E1AD'  B7                OR     A               ;Test daca nu e specificat
E1AE'  C2 E4A5'          JP     NZ,TRANCOM      ;Pentru diferit de 0 poate fi
                                                ;numai comanda tranzitorie
E1B1'  CD E12E'          CALL   PICKCOM         ;Cauta comanda rezidenta
E1B4'  21 E1C1'          LD     HL,COMADRS      ;Tabela cu adrese rutine
E1B7'  5F                LD     E,A
E1B8'  16 00             LD     D,0
E1BA'  19                ADD    HL,DE
E1BB'  19                ADD    HL,DE
E1BC'  7E                LD     A,(HL)
E1BD'  23                INC    HL
E1BE'  66                LD     H,(HL)
E1BF'  6F                LD     L,A             ;H,L specifica adresa functiei
E1C0'  E9                JP     (HL)
E1C1'                COMADRS:
E1C1'  E277' E31F'       DW     DIRCOM,ERACOM,TYPECOM,SAVECOM
E1C5'  E35D' E3AD'
E1C9'  E410' E48E'       DW     RENCOM,USERCOM,TRANCOM
E1CD'  E4A5'
E1CF'                FATAL:
E1CF'  21 76F3           LD     HL,076F3H       ;eroare de compatibilitate
E1D2'  22 DE00'          LD     (CCPBASE),HL    ;distruge START CCP inserind:
E1D5'  21 DE00'          LD     HL,CCPBASE      ;       DI
E1D8'  E9                JP     (HL)            ;       HLT
                     ;   *****  ******          ;la CCPBASE (acolo-i START CCP)
E1D9'                RDERR:
E1D9'  01 E1DF'          LD     BC,RDMESS       ;afiseaza mesaj de eroare la
E1DC'  C3 DEA7'          JP     NLMESS          ;citire
```

```
E1DF'                   RDMESS:
E1DF'  52 45 41 44              DB      'READ ERROR'
E1E3'  20 45 52 52
E1E7'  4F 52
E1E9'  00                       DB      0
                        ;       *****   ******
E1EA'                   NOFERR:
E1EA'  01 E1F0'                 LD      BC,MESNOF       ;Afiseaza mesaj de eroare
E1ED'  C3 DEA7'                 JP      NLMESS          ;"NO FILE"
E1F0'  4E 4F 20 46      MESNOF: DB      'NO FILE'
E1F4'  49 4C 45
E1F7'  00                       DB      0
                        ;       *****   ******
E1F8'                   DECBIN:         ;Converteste numere zecimale la forma binara
E1F8'  CD E05E'                 CALL    TRABUF          ;Transfer in zona de comanda
E1FB'  3A E5F0'                 LD      A,(SPECDRIVE)   ;Incarca drive-ul specificat
                                                        ;de TRABUF
E1FE'  B7                       OR      A               ;Test daca este specificatie
E1FF'  C2 E009'                 JP      NZ,ERROR    ;Un numar zecimal nu incepe cu "DR:"
E202'  21 E5CE'                 LD      HL,FILNAME      ;Indicator pentru numar zecimal
E205'  01 000B                  LD      BC,11           ;Lungime maxima in registrul C
E208'                   DEC1:                           ;registrul B=0
E208'  7E                       LD      A,(HL)          ;Preia caracter
E209'  FE 20                    CP      ' '             ;Test pentru sfirsit numar
E20B'  CA E233'                 JP      Z,DEC2          ;Daca-i spatiu e gata
E20E'  23                       INC     HL              ;Locatia urmatoare
E20F'  D6 30                    SUB     '0'             ;Scade baza cifrelor ASCII
E211'  FE 0A                    CP      10              ;Teste daca e intre 0 si 9
E213'  D2 E009'                 JP      NC,ERROR        ;Daca nu, eroare
E216'  57                       LD      D,A             ;Salveaza cifra
E217'  78                       LD      A,B             ;Valoare calculata anterior
E218'  E6 E0                    AND     0E0H            ;Masca pentru inmultire zecimala
E21A'  C2 E009'                 JP      NZ,ERROR        ;Numar prea mare
E21D'  78                       LD      A,B             ;Incarca din nou
E21E'  07                       RLCA                    ; * 2
E21F'  07                       RLCA                    ; * 4
E220'  07                       RLCA                    ; * 8
E221'  80                       ADD     A,B             ;Aduna valoarea
E222'  DA E009'                 JP      C,ERROR         ;Teste daca e depasire
E225'  80                       ADD     A,B             ;Aduna valoarea.  Acum A=B*10
E226'  DA E009'                 JP      C,ERROR         ;Test daca e depasire
E229'  82                       ADD     A,D             ;Aduna noua cifra CMPS
E22A'  DA E009'                 JP      C,ERROR         ;Test daca e depasire
E22D'  47                       LD      B,A             ;Salveaza noua valoare
E22E'  0D                       DEC     C               ;Decrementeaza numar
E22F'  C2 E208'                 JP      NZ,DEC1         ;Mai sint caractere? Da, salt
E232'  C9                       RET
E233'                   DEC2:
E233'  7E                       LD      A,(HL)          ;Numarul e gata
E234'  FE 20                    CP      ' '             ;Salt peste spatiile urmatoare
E236'  C2 E009'                 JP      NZ,ERROR        ;Permise numai spatii
E239'  23                       INC     HL              ;Urmatorul caracter
E23A'  0D                       DEC     C               ;Decrementeaza numar
E23B'  C2 E233'                 JP      NZ,DEC2         ;Mai ?
E23E'  78                       LD      A,B             ;Reface rezultat
E23F'  C9                       RET
                        ;       *****   ******
E240'                   MOV3DH:                         ;Muta 3 caractere sir
E240'  06 03                    LD      B,3
                        ;       *****   ******
E242'                   MOVDHB:                         ;Muta sir de la locatia H,L
```

```
E242'  7E              LD      A,(HL)          ;la locatia D,E. Registrul B
E243'  12              LD      (DE),A          ;contine numar octeti sir
E244'  23              INC     HL              ;Urmatoarea sursa
E245'  13              INC     DE              ;Urmatoarea destinatie
E246'  05              DEC     B               ;Decrementeaza numar
E247'  C2 E242'        JP      NZ,MOVDHB       ;Mai ?
E24A'  C9              RET
                ;      *****   ******
E24B'                  ADDCH:          ;Intoarce continutul de la (80H+reg.A+reg.
E24B'  21 0080         LD      HL,128          ;Incarca in H,L implicitul (80H)
E24E'  81              ADD     A,C             ;Aduna registrul C la registrul A
E24F'  CD E059'        CALL    ADDHL           ;Aduna registrul A la H,L
E252'  7E              LD      A,(HL)          ;Incarca din locatia de memorie
E253'  C9              RET
                ;      *****   ******
E254'                  SELD1:
E254'  AF              XOR     A               ;Pune 0 in acumulator
E255'  32 E5CD'        LD      (COMFCB),A ;Memoreaza in FCB locatia pentru drive
E258'  3A E5F0'        LD      A,(SPECDRIVE)   ;Incarca drive-ul specificat
E25B'  B7              OR      A               ;Test daca
E25C'  C8              RET     Z               ;e implicit. Nici o actiune
E25D'  3D              DEC     A          ;Daca nu, converteste la conventia FCB
E25E'  21 E5EF'        LD      HL,CLOGDSK      ;Test daca e acelasi
E261'  BE              CP      (HL)            ;cu discul conectat
E262'  C8              RET     Z               ;La coincidenta nici o actiune
E263'  C3 DEBD'        JP      CSELDSK         ;altfel selecteaza disc
                ;      *****   ******
E266'                  SELD2:
E266'  3A E5F0'        LD      A,(SPECDRIVE)   ;Incarca discul specificat
E269'  B7              OR      A               ;Test daca
E26A'  C8              RET     Z               ;e implicit. Nici o actiune
E26B'  3D              DEC     A               ;Converteste la conventia FCB
E26C'  21 E5EF'        LD      HL,CLOGDSK      ;Incarca discul conectat
E26F'  BE              CP      (HL)            ;La coincidenta
E270'  C8              RET     Z               ;nici o actiune
E271'  3A E5EF'        LD      A,(CLOGDSK)     ;altfel incarca discul conectat
E274'  C3 DEBD'        JP      CSELDSK         ;si il selecteaza
                ;Aici incepe comanda DIRectory
                ;Forma este :DIR [DR:]NAME.EXT
E277'                  DIRCOM:
E277'  CD E05E'        CALL    TRABUF          ;Pregateste zona de comanda
E27A'  CD E254'        CALL    SELD1           ;Selecteaza discul daca e necesar
E27D'  21 E5CE'        LD      HL,FILNAME      ;Nume de fisier
E280'  7E              LD      A,(HL)          ;Preia caracter
E281'  FE 20           CP      ' '             ;Test daca e specificatie de fisier
E283'  C2 E28F'        JP      NZ,DIR2         ;Afiseaza toate punctele de intrare
                                               ;in lista DIRECTORY, altfel
E286'  06 0B           LD      B,11            ;numai cele care se potrivesc
E288'                  DIR1:                   ;reg.B = lungime nume fisier
E288'  36 3F           LD      (HL),'?'        ;Umple nume fisier cu
E28A'  23              INC     HL              ;semne de intrebare
E28B'  05              DEC     B               ;Mai este de umplut
E28C'  C2 E288'        JP      NZ,DIR1         ;Urmatorul
E28F'                  DIR2:                   ;Numele de fisier pregatit
E28F'  1E 00           LD      E,0             ;Numarator de linii
E291'  D5              PUSH    DE              ;Salvare in stiva
E292'  CD DEE9'        CALL    COMSRCH    ;Cauta puncte de intrare in DIRECTORY
E295'  CC E1EA'        CALL    Z,NOFERR   ;Daca nu sint afiseaza mesaj de eroare
E298'                  DIR3:
E298'  CA E31B'        JP      Z,DIR11         ;Nu mai sint puncte de intrare
E29B'  3A E5EE'        LD      A,(DIRENTRY)    ;Incarca cod de intrare in
```

```
                                        ;DIRECTORY 0,1,2,3
E29E'  0F                       RRCA    ;Fiecare punct de intrare are
                                        ;lungimea de 32 octeti
E29F'  0F                       RRCA    ;Deci,pentru a gasi informatia din DIR
E2A0'  0F                       RRCA    ;se inmulteste codul "DIR"cu 32
E2A1'  E6 60                    AND     60H     ;se mascheaza bitii nenecesari
E2A3'  4F                       LD      C,A     ;reg.C-deplasamentul pentru DIRECTORY
E2A4'  3E 0A                    LD      A,10    ;reg.A-deplasament atribut sistem
E2A6'  CD E24B'                 CALL    ADDCH   ;Calculeaza locatia pentru atribut sistem
E2A9'  17                       RLA             ;Test daca fisierul
E2AA'  DA E30F'                 JP      C,DIR10 ;este invizibil
E2AD'  D1                       POP     DE      ;Restaurarea numaratorului de linii
E2AE'  7B                       LD      A,E     ;Salvare in acumuator
E2AF'  1C                       INC     E       ;Incrementeaza numaratorul de linii
E2B0'  D5                       PUSH    DE      ;Salveaza noua valoare
E2B1'  E6 03                    AND     3       ;Test daca e necesara noua linie
E2B3'  F5                       PUSH    AF      ;Salveaza indicatorii
E2B4'  C2 E2CC'                 JP      NZ,DIR4 ;Nu e linie de start
E2B7'  CD DE98'                 CALL    CCRLF   ;Afiseaza CRLF
E2BA'  C5                       PUSH    BC      ;Salveaza deplasament pentru DIR
E2BB'  CD DFD0'                 CALL    LOGGIN  ;Incarca discul conectat
E2BE'  C1                       POP     BC      ;Restaureaza deplasament pentru DIR
E2BF'  C6 41                    ADD     A,'A'   ;Pregateste nume 'drive' adaugind
                                        ;litera de baza
E2C1'  CD DE92'                 CALL    COSVB   ;Afiseaza drive
E2C4'  3E 3A                    LD      A,':'   ;si ':'
E2C6'  CD DE92'                 CALL    COSVB   ;este de asemenea afisat
E2C9'  C3 E2D4'                 JP      DIR5
E2CC'                   DIR4:                   ;Nu e prima linie
E2CC'  CD DEA2'                 CALL    BLANCK  ;Afiseaza spatiu
E2CF'  3E 3A                    LD      A,':'   ;si delimitatorul
E2D1'  CD DE92'                 CALL    COSVB   ; ':'
E2D4'                   DIR5:
E2D4'  CD DEA2'                 CALL    BLANCK  ;Urmat de un spatiu
E2D7'  06 01                    LD      B,1     ;sare cod utilizator din
E2D9'                   DIR6:                   ;"DIRECTORY"
E2D9'  78                       LD      A,B     ;Nume deplasament
E2DA'  CD E24B'                 CALL    ADDCH   ;Calculeaza adresa noului carac-
                                        ;ter si incarca
E2DD'  E6 7F                    AND     127     ;Inlatura bitul de paritate
E2DF'  FE 20                    CP      ' '     ;Este spatiu
E2E1'  C2 E2F9'                 JP      NZ,DIR8 ;Nu
E2E4'  F1                       POP     AF      ;Restaureaza numar linie
E2E5'  F5                       PUSH    AF      ;Salveaza din nou
E2E6'  FE 03                    CP      3       ;Test daca e ultimul in linie
E2E8'  C2 E2F7'                 JP      NZ,DIR7 ;daca nu, umple cu spatii
E2EB'  3E 09                    LD      A,9     ;altfel, test daca e extensie
E2ED'  CD E24B'                 CALL    ADDCH   ;Acum se calculeaza
E2F0'  E6 7F                    AND     127     ;fara bitul de paritate
E2F2'  FE 20                    CP      ' '     ;Incepe cu spatiu
E2F4'  CA E30E'                 JP      Z,DIR9  ;Daca da merge la urmatorul, alt-
E2F7'                   DIR7:                   ;fel umple cu spatii
E2F7'  3E 20                    LD      A,' '   ;Incarca spatiu
E2F9'                   DIR8:
E2F9'  CD DE92'                 CALL    COSVB   ;Afiseaza caracter
E2FC'  04                       INC     B       ;Urmatorul
E2FD'  78                       LD      A,B     ;Test daca mai sint caractere in
E2FE'  FE 0C                    CP      12      ;numele de fisier si extensie
E300'  D2 E30E'                 JP      NC,DIR9 ;Nu mai sint, mergi la urmatorul
                                        ;punct de intrare
E303'  FE 09                    CP      9       ;Test daca a inceput extensia
```

```
E305'  C2 E2D9'          JP    NZ,DIR6          ;Inca nume, altfel
E308'  CD DEA2'          CALL  BLANCK           ;afiseaza spatiu
E30B'  C3 E2D9'          JP    DIR6    ;Mergi la urmatorul caracter din extensie
E30E'                    ;Punctul de intrare curent, epuizat
E30E'  F1                POP   AF               ;Stiva goala
E30F'                    ;
E30F'  CD DFC2'          CALL  CNSTCI           ;Test daca
E312'  C2 E31B'          JP    NZ,DIR11         ;mai e un caracter
E315'  CD DEE4'          CALL  SRCHNEXT         ;Daca nu, continua
E318'  C3 E298'          JP    DIR3             ;comanda
E31B'                    ;altfel abandoneaza
E31B'  D1                POP   DE               ;Goleste stiva
E31C'  C3 E586'          JP    COMEXIT          ;si reintra in modul comanda
                         ;Comanda 'ERASE FILE' incepe aici
                         ;Forma comenzii este : ERA [DR:] NAME.EXT
E31F'                    ;
E31F'  CD E05E'          CALL  TRABUF           ;Pregateste zona comanda
E322'  FE 0B             CP    11               ;Test daca este "*" (tot)
E324'  C2 E342'          JP    NZ,ERA1          ;Nu e stergere totala
E327'  01 E352'          LD    BC,ERAMES        ;Afiseaza mesaj "ALL?" (toate?)
E32A'  CD DEA7'          CALL  NLMESS           ;ca avertizare
E32D'  CD DF39'          CALL  LINEINP          ;asteapta confirmare utilizator
E330'  21 DE07'          LD    HL,BUFLEN        ;Numar de caractere introduse
E333'  35                DEC   (HL)             ;Test daca e unul
E334'  C2 E182'          JP    NZ,RESUME        ;Daca sint mai mult de unu,
E337'  23                INC   HL               ;abandoneaza. Altfel preia
E338'  7E                LD    A,(HL)           ;caracter
E339'  FE 59             CP    'Y'              ;Test daca e validata stergerea
E33B'  C2 E182'          JP    NZ,RESUME        ;Daca nu-i "Y" (DA) abandoneaza
E33E'  23                INC   HL               ;Urmatorul caracter
E33F'  22 DE88'          LD    (BUFPOINT),HL    ;Salveaza indicator
E342'                    ;
E342'  CD E254'          CALL  SELD1            ;Selecteaza disc daca e necesar
E345'  11 E5CD'          LD    DE,COMFCB        ;Pregateste
E348'  CD DEEF'          CALL  ERAFILE          ;apel la stergere BDOS
E34B'  3C                INC   A                ;Test daca stergerea
E34C'  CC E1EA'          CALL  Z,NOFERR         ;s-a incheiat cu succes
E34F'  C3 E586'          JP    COMEXIT          ;Iesire din comanda
E352'                    ;
E352'  41 4C 4C 20       DB    'ALL (Y/N)?'
E356'  28 59 2F 4E
E35A'  29 3F
E35C'  00                DB    0
                         ;Comanda 'TYPE' incepe aici
                         ;Forma comenzii TYPE [DR:] NAME.EXT
E35D'                    ;
E35D'  CD E05E'          CALL  TRABUF           ;Pregateste zona
E360'  C2 E009'          JP    NZ,ERROR         ;Nu sint permise "*"
E363'  CD E254'          CALL  SELD1            ;Selecteaza daca e necesar
E366'  CD DED0'          CALL  COMOPEN          ;Deschide fisier
E369'  CA E3A7'          JP    Z,TYPE4          ;Nu e gasit fisierul
E36C'  CD DE98'          CALL  CCRLF            ;altfel afiseaza CRLF
E36F'  21 E5F1'          LD    HL,BCOUNT        ;si pune numarul de
E372'  36 FF             LD    (HL),255         ;octeti inregistrare
E374'                    ;
E374'  21 E5F1'          LD    HL,BCOUNT        ;Test daca mai sint
E377'  7E                LD    A,(HL)           ;octeti in inregistrare
E378'  FE 80             CP    128              ;de exemplu contor octeti <128
E37A'  DA E387'          JP    C,TYPE2          ;Ia urmatorul octet
E37D'  E5                PUSH  HL               ;Salveaza indicator
E37E'  CD DEFE'          CALL  COMREAD          ;Citeste urmatoarea inregistrare
```

```
E381'  E1                        POP     HL              ;Restaureaza indicator
E382'  C2 E3A0'                  JP      NZ,TYPE3        ;Nu mai sint inregistrari sau
E385'  AF                        XOR     A               ;eroare de citire. Anuleaza numar
E386'  77                        LD      (HL),A          ;de octeti
E387'                    TYPE2:
E387'  34                        INC     (HL)            ;Incrementeaza numar de octeti
E388'  21 0080                   LD      HL,128          ;Zona tampon inregistrare
E38B'  CD E059'                  CALL    ADDHL           ;Calculeaza adresa curenta a oc-
E38E'  7E                        LD      A,(HL)          ;tetului. Incarca caracter
E38F'  FE 1A                     CP      26              ;Test daca e CTRL si Z
E391'  CA E586'                  JP      Z,COMEXIT       ;Daca da, fisierul e gata
E394'  CD DE8C'                  CALL    CCONOUT         ;altfel afiseaza caracter
E397'  CD DFC2'                  CALL    CNSTCI          ;Test daca se abandoneaza comanda
E39A'  C2 E586'                  JP      NZ,COMEXIT      ;Iesire
E39D'  C3 E374'                  JP      TYPE1           ;Urmatorul caracter
E3A0'                    TYPE3:
E3A0'  3D                        DEC     A               ;Test daca este
E3A1'  CA E586'                  JP      Z,COMEXIT       ;EOF sau
E3A4'  CD E1D9'                  CALL    RDERR           ;cod de eroare la citire
E3A7'                    TYPE4:
E3A7'  CD E266'                  CALL    SELD2           ;Restaureaza discul conectat
E3AA'  C3 E009'                  JP      ERROR           ;Procedura de tratare eroare
                         ;Comanda SAVE incepe aici
                         ;Forma comenzii este SAVE numar zecimal [DR:]FILE.EXT
E3AD'
E3AD'                    SAVECOM:
E3AD'  CD E1F8'                  CALL    DECBIN          ;Incarca numar de pagini de sal-
E3B0'  F5                        PUSH    AF              ;vat. Salveaza in stiva
E3B1'  CD E05E'                  CALL    TRABUF          ;Pregateste nume fisier
E3B4'  C2 E009'                  JP      NZ,ERROR        ;Eroare daca e "*"
E3B7'  CD E254'                  CALL    SELD1           ;Selecteaza disc daca e necesar
E3BA'  11 E5CD'                  LD      DE,COMFCB       ;Indicator FCB
E3BD'  D5                        PUSH    DE              ;Salveaza in stiva
E3BE'  CD DEEF'                  CALL    ERAFILE         ;Sterge fisier mai intii
E3C1'  D1                        POP     DE              ;Restaureaza indicator FCB
E3C2'  CD DF09'                  CALL    CREATE          ;Acum creaza fisier
E3C5'  CA E3FB'                  JP      Z,SAVE3         ;Test daca s-a incheiat cu
E3C8'  AF                        XOR     A               ;succes, fara eroare.
E3C9'  32 E5ED'                  LD      (COMCR),A
E3CC'  F1                        POP     AF              ;Restaureaza numar de pagini
E3CD'  6F                        LD      L,A             ;Muta in registrele
E3CE'  26 00                     LD      H,0             ;L si H
E3D0'  29                        ADD     HL,HL           ;Acum *2 (128*2 = pagina )
E3D1'  11 0100                   LD      DE,100H         ;Zona TPA
E3D4'                    SAVE1:
E3D4'  7C                        LD      A,H             ;Test daca
E3D5'  B5                        OR      L               ;mai sint pagini
E3D6'  CA E3F1'                  JP      Z,SAVE2         ;Nu mai sint
E3D9'  2B                        DEC     HL              ;Decrementeaza numar pagini
E3DA'  E5                        PUSH    HL              ;Salveaza numar pagini
E3DB'  21 0080                   LD      HL,128          ;Lungime inregistrare
E3DE'  19                        ADD     HL,DE   ;Pregateste adresa pentru urmatorul DMA
E3DF'  E5                        PUSH    HL              ;Salveaza urmatorul DMA in stiva
E3E0'  CD DFD8'                  CALL    SETDMA          ;Pune DMA la adresa curenta
E3E3'  11 E5CD'                  LD      DE,COMFCB       ;Pregateste si
E3E6'  CD DF04'                  CALL    WRITE           ;apeleaza BDOS
E3E9'  D1                        POP     DE              ;Restaureaza adresa DMA
E3EA'  E1                        POP     HL              ;Restaureaza numar pagini
E3EB'  C2 E3FB'                  JP      NZ,SAVE3        ;Test daca scrierea s-a incheiat
E3EE'  C3 E3D4'                  JP      SAVE1           ;cu succes. Urmatoarea inregis-
E3F1'                    SAVE2:                          ;trare.
E3F1'  11 E5CD'                  LD      DE,COMFCB       ;Toate inregistrarile
```

```
E3F4'  CD DEDA'          CALL   CCLOSE          ;Acum inchide fisier
E3F7'  3C                INC    A               ;Test daca s-a incheiat cu succes
E3F8'  C2 E401'          JP     NZ,SAVE4        ;Este OK
E3FB'                 SAVE3:
E3FB'  01 E407'          LD     BC,SAVE5        ;Afiseaza mesajul
E3FE'  CD DEA7'          CALL   NUMESS          ;"NO SPACE" (nu-i loc)
E401'                 SAVE4:
E401'  CD DFD5'          CALL   SDMAD           ;Pune DMA implicit la 80H
E404'  C3 E586'          JP     COMEXIT         ;Iesire din comanda SAVE
E407'                 SAVE5:
E407'  4E 4F 20 53       DB     'NO SPACE'
E40B'  50 41 43 45
E40F'  00                DB     0
                      ;Comanda RENAME incepe aici
                      ;Forma comenzii este RENAME [DR:]Nume nou.EXT=[DR:]Nume vechi.EXT
E410'                 RENCOM:
E410'  CD E05E'          CALL   TRABUF          ;Pregateste nume nou
E413'  C2 E009'          JP     NZ,ERROR        ;Nu e permis "*"
E416'  3A E5F0'          LD     A,(SPECDRIVE)   ;Incarca si salveaza
E419'  F5                PUSH   AF              ;discul specificat
E41A'  CD E254'          CALL   SELD1           ;Selecteaza daca e necesar
E41D'  CD DEE9'          CALL   COMSRCH         ;Cauta daca exista deja
E420'  C2 E479'          JP     NZ,REN5         ;Exista eroare fisier
E423'  21 E5CD'          LD     HL,COMFCB       ;Pregateste al doilea
E426'  11 E500'          LD     DE,SECFN        ;nume din zona FCB
E429'  06 10             LD     B,16            ;mutind 16 caractere
E42B'  CD E242'          CALL   MOVDHB          ;din FCB in SECFN
E42E'  2A DE88'          LD     HL,(BUFPOINT)   ;Incarca indicator pe zona tampon
E431'  EB                EX     DE,HL           ;de intrare
E432'  CD E04F'          CALL   SKPBLK          ;Sare spatii
E435'  FE 3D             CP     '='             ;Test daca e delimitator valid
E437'  CA E43F'          JP     Z,REN1          ;egal (adica '=')sau
E43A'  FE 5F             CP     95              ;sageata inapoi
E43C'  C2 E473'          JP     NZ,REN4         ;Nu, eroare
E43F'                 REN1:
E43F'  EB                EX     DE,HL
E440'  23                INC    HL              ;Salveaza indicator
E441'  22 DE88'          LD     (BUFPOINT),HL   ;pe al doilea nume
E444'  CD E05E'          CALL   TRABUF          ;Pregateste nume
E447'  C2 E473'          JP     NZ,REN4         ;Nu "*"
E44A'  F1                POP    AF              ;Restaureaza primul disc specifi-
E44B'  47                LD     B,A             ;cat si salveaza in registrul B
E44C'  21 E5F0'          LD     HL,SPECDRIVE    ;Test daca noul
E44F'  7E                LD     A,(HL)          ;disc specificat
E450'  B7                OR     A               ;este implicit...
E451'  CA E459'          JP     Z,REN2
E454'  B8                CP     B               ;sau coincide cu cel vechi
E455'  70                LD     (HL),B          ;Restaureaza discul specificat
E456'  C2 E473'          JP     NZ,REN4         ;Eroare
E459'                 REN2:
E459'  70                LD     (HL),B          ;Disc specificat implicit prin
E45A'  AF                XOR    A               ;fisier sursa. Initializeaza
E45B'  32 E5CD'          LD     (COMFCB),A      ;discul FCB
E45E'  CD DEE9'          CALL   COMSRCH         ;Cauta "DIRECTORY"
E461'  CA E46D'          JP     Z,REN3          ;Test daca e gasita
E464'  11 E5CD'          LD     DE,COMFCB       ;Pregateste si
E467'  CD DF0E'          CALL   CRENAME         ;apeleaza BDOS
E46A'  C3 E586'          JP     COMEXIT         ;Iesire din comanda
E46D'                 REN3:
E46D'  CD E1EA'          CALL   NOFERR          ;Nu s-a gasit eroare
E470'  C3 E586'          JP     COMEXIT         ;Iesire din comanda
```

```
E473'                   REN4:
E473'  CD E266'                 CALL    SELD2           ;Selecteaza disc
E476'  C3 E009'                 JP      ERROR           ;Eroare, nu coincide discul
E479'                   REN5:
E479'  01 E482'                 LD      BC,REN6         ;Afiseaza
E47C'  CD DEA7'                 CALL    NLMESS          ;mesaj de eroare
E47F'  C3 E586'                 JP      COMEXIT         ;Iesire din comanda
E482'                   REN6:
E482'  46 49 4C 45              DB      'FILE EXISTS'
E486'  20 45 58 49
E48A'  53 54 53
E48D'  00                       DB      0
                        ;Comanda USER incepe aici
                        ;Forma comenzii este USER numar zecimal
E48E'                   USERCOM:
E48E'  CD E1F8'                 CALL    DECBIN          ;Incarca cod utilizator
E491'  FE 10                    CP      16              ;Test daca este
E493'  D2 E009'                 JP      NC,ERROR        ;mai mare decit 15
E496'  5F                       LD      E,A             ;Salveaza in registrul E
E497'  3A E5CE'                 LD      A,(FILNAME)     ;Test daca sint caractere in plus
E49A'  FE 20                    CP      ' '             ;diferite de 0
E49C'  CA E009'                 JP      Z,ERROR         ;Procedura de eroare
E49F'  CD DF15'                 CALL    PUTUSER         ;Selecteaza cod utilizator
E4A2'  C3 E589'                 JP      EXIT1           ;Iesire din comanda
                        ;Comenzile tranzitorii incep aici
                        ;Forma comenzii [DR:] nume fisier 1.EXT fisier 2.EXT
E4A5'                   TRANCOM:
E4A5'  CD DFF5'                 CALL    VRSVER          ;Verifica versiunea BDOS si CCP
E4A8'  3A E5CE'                 LD      A,(FILNAME)     ;Incarca primul caracter din nume
E4AB'  FE 20                    CP      ' '             ;Test daca e spatiu
E4AD'  C2 E4C4'                 JP      NZ,TRN1         ;Nume fisier specificat
                                                        ;Selecteaza comanda conectare disc
E4B0'  3A E5F0'                 LD      A,(SPECDRIVE)   ;Incarca drive-ul specificat
E4B3'  B7                       OR      A               ;Test daca nu e specificatie
E4B4'  CA E589'                 JP      Z,EXIT1         ;Iesire din comanda
E4B7'  3D                       DEC     A               ;altfel incarca
E4B8'  32 E5EF'                 LD      (CLOGDSK),A     ;memoreaza
E4BB'  CD DF29'                 CALL    STLOGG          ;si pune conectare
E4BE'  CD DEBD'                 CALL    CSELDSK         ;si selectie disc
E4C1'  C3 E589'                 JP      EXIT1           ;Iesire comanda
E4C4'                   TRN1:
E4C4'  11 E5D6'                 LD      DE,EXTEN        ;Incarca
E4C7'  1A                       LD      A,(DE)          ;extensia si test
E4C8'  FE 20                    CP      ' '             ;daca exista extensie
E4CA'  C2 E009'                 JP      NZ,ERROR        ;Nu e permisa extensie
E4CD'  D5                       PUSH    DE              ;Salveaza indicator
E4CE'  CD E254'                 CALL    SELD1           ;Selecteaza disc daca e necesar
E4D1'  D1                       POP     DE              ;Restaureaza indicator
E4D2'  21 E583'                 LD      HL,COMEXT       ;Se adauga extensia
E4D5'  CD E240'                 CALL    MOV3DH          ;COM la numele fisierului
E4D8'  CD DED0'                 CALL    COMOPEN         ;Deschide fisier
E4DB'  CA E56B'                 JP      Z,CERR01        ;Test daca e eroare
E4DE'  21 0100                  LD      HL,100H         ;Incarca start TPA
E4E1'                   TRN2:
E4E1'  E5                       PUSH    HL              ;Salveaza indicator memorie
E4E2'  EB                       EX      DE,HL           ;Incarca indicator memorie in D,E
E4E3'  CD DFD8'                 CALL    SETDMA          ;Pune DMA
E4E6'  11 E5CD'                 LD      DE,COMFCB       ;Citeste
E4E9'  CD DEF9'                 CALL    SEQREAD         ;sector in memorie
E4EC'  C2 E501'                 JP      NZ,LDEND        ;Test daca mai sint inregistrari
E4EF'  E1                       POP     HL              ;Restaureaza lungimea
```

```
E4F0'  11 0080          LD    DE,128      ;inregistrarii din memorie
E4F3'  19               ADD   HL,DE       ;Pregateste noua adresa de incar
E4F4'  11 DE00'         LD    DE,CCPBASE  ;care. Test daca
E4F7'  7D               LD    A,L         ;noua adresa este inca mai
E4F8'  93               SUB   E           ;mica decit
E4F9'  7C               LD    A,H         ;baza CCP, deci
E4FA'  9A               SBC   A,D         ;nu poate apare segment
E4FB'  D2 E571'         JP    NC,LDERR    ;Adresa DMA prea mare
E4FE'  C3 E4E1'         JP    TRN2        ;Incarca urmatoarea inregistrare
E501'                LDEND:               ;Incarcarea programului este gata
E501'  E1               POP   HL          ;Golire stiva
E502'  3D               DEC   A           ;Test daca-i EOF sau
E503'  C2 E571'         JP    NZ,LDERR    ;alta eroare
E506'  CD E266'         CALL  SELD2       ;Reselecteaza discul conectat
E509'  CD E05E'         CALL  TRABUF      ;Inspecteaza ce a ramas din zona
                                          ;tampon de intrare
E50C'  21 E5F0'         LD    HL,SPECDRIVE;Incarca discul specificat
E50F'  E5               PUSH  HL          ;Salveaza in stiva
E510'  7E               LD    A,(HL)      ;si
E511'  32 E5CD'         LD    (COMFCB),A  ;salveaza in zona FCB
E514'  3E 10            LD    A,16        ;Deplasament zona comanda
E516'  CD E060'         CALL  TRB01       ;Transfera al doilea nume in zona
                                          ;de comanda +16
E519'  E1               POP   HL          ;Restaureaza adresa discului
E51A'  7E               LD    A,(HL)      ;Muta drive-ul in
E51B'  32 E5DD'         LD    (SECFN),A   ;a doua zona FCB
E51E'  AF               XOR   A
E51F'  32 E5ED'         LD    (COMCR),A   ;Initializeaza zona CR
E522'  11 005C          LD    DE,5CH      ;Primul FCB returnat
E525'  21 E5CD'         LD    HL,COMFCB   ;de CCP
E528'  06 21            LD    B,21H       ;lungimea
E52A'  CD E242'         CALL  MOVDHB      ;Transfera zona din comanda la
E52D'  21 DE08'         LD    HL,CINPBUF  ;5CH. Zona tampon intrare
E530'                MOVLINE:
E530'  7E               LD    A,(HL)      ;Incarca din zona tampon
E531'  B7               OR    A           ;Test daca e sfirsitul comenzii
                                          ;tranzitorii
E532'  CA E53E'         JP    Z,MVL0      ;Aici incepe restul liniei
E535'  FE 20            CP    ' '         ;ca mai sus
E537'  CA E53E'         JP    Z,MVL0
E53A'  23               INC   HL          ;Urmatoarea pozitie
E53B'  C3 E530'         JP    MOVLINE     ;Din nou
E53E'                MVL0:
E53E'  06 00            LD    B,0         ;Numar octeti
E540'  11 0081          LD    DE,129      ;zona de comunicatie
E543'                MVL1:
E543'  7E               LD    A,(HL)      ;Muta linia de intrare
E544'  12               LD    (DE),A      ;in zona de comunicatie
E545'  B7               OR    A           ;pina la primul 0
E546'  CA E54F'         JP    Z,LEAVE     ;E gata
E549'  04               INC   B           ;Incrementeaza numar octeti
E54A'  23               INC   HL          ;Sursa
E54B'  13               INC   DE          ;si destinatia
E54C'  C3 E543'         JP    MVL1        ;Din nou
E54F'                LEAVE:               ;
E54F'  78               LD    A,B         ;Salveaza
E550'  32 0080          LD    (128),A     ;numar de octeti
E553'  CD DE98'         CALL  CCRLF       ;Afiseaza CRLF
E556'  CD DFD5'         CALL  SDMAD       ;Restaureaza adresa DMA
E559'  CD DF1A'         CALL  USSAV       ;Salveaza cod utilizator
E55C'  CD 0100          CALL  100H        ;Transfera control la programu
```

```
E55F' 31 E5AB'              LD      SP,CLOCSTACK    ;tranzitoriu. Pune SP
E562' CD DF29'              CALL    STLOGG          ;Pune disc conectat
E565' CD DEBD'              CALL    CSELDSK         ;Selecteaza disc
E568' C3 E182'              JP      RESUME          ;Reintra comanda
E56B'                   CERR01:
E56B' CD E266'              CALL    SELD2           ;Selecteaza disc
E56E' C3 E009'              JP      ERROR           ;Procedura de eroare
E571'                   LDERR:
E571' 01 E57A'              LD      BC,LDMESS       ;Incarcare eronata
E574' CD DEA7'              CALL    NLMESS          ;Mesaj eroare
E577' C3 E586'              JP      COMEXIT         ;Iesire din comanda
E57A'                   LDMESS:
E57A' 42 41 44 20           DB      'BAD LOAD'
E57E' 4C 4F 41 44
E582' 00                    DB      0
E583'                   COMEXT:
E583' 43 4F 4D              DB      'COM'           ;Extensie program tranzitoriu
E586'                   COMEXIT:
E586' CD E266'              CALL    SELD2           ;Reselecteaza
E589'                   EXIT1:
E589' CD E05E'              CALL    TRABUF          ;Test daca este
E58C' 3A E5CE'              LD      A,(FILNAME)     ;nume de fisier
E58F' D6 20                 SUB     ' '             ;in plus
E591' 21 E5F0'              LD      HL,SPECDRIVE
E594' B6                    OR      (HL)
E595' C2 E009'              JP      NZ,ERROR
E598' C3 E182'              JP      RESUME
                        ;Rezerva 16 octeti pentru stiva
E59B'                       DS      16
E5AB'                   CLOCSTACK:
E5AB'                   SUBSWITCH:
E5AB' 00                    DB      0
E5AC' 00                SUBFCB: DB      0
E5AD' 24 24 24 20           DB      '$$$     SUB'
E5B1' 20 20 20 20
E5B5' 53 55 42
E5B8' 00 00                 DB      0,0
E5BA' 00                FCBS2:  DB      0
E5BB' 00 00 00 00       SUBRC:  DB      0,0,0,0,0,0,0,0
E5BF' 00 00 00 00
E5C3' 00 00 00 00           DB      0,0,0,0,0,0,0,0
E5C7' 00 00 00 00
E5CB' 00                    DB      0
E5CC' 00                SUBCR:  DB      0
E5CD' 00                COMFCB: DB      0
E5CE' 00 00 00 00       FILNAME:DB      0,0,0,0,0,0,0,0
E5D2' 00 00 00 00
E5D6' 00 00 00 00       EXTEN:  DB      0,0,0,0,0,0,0
E5DA' 00 00 00
E5DD' 00 00 00 00       SECFN:  DB      0,0,0,0,0,0,0,0
E5E1' 00 00 00 00
E5E5' 00 00 00 00           DB      0,0,0,0,0,0,0,0
E5E9' 00 00 00 00
E5ED' 00                COMCR:  DB      0
E5EE'                   DIRENTRY:
E5EE' 00                    DB      0
E5EF' 00                CLOGDSK:DB      0
E5F0'                   SPECDRIVE:
E5F0' 00                    DB      0
E5F1' 00                BCOUNT: DB      0
E5F2' 00 00 00 00       DUMMY:  DB      0,0,0,0,0,0,0
```

```
E5F6'  00 00 00
E5F9'  00 00 00 00          DB      0,0,0,0,0,0,0
E5FD'  00 00 00
E600'  F9 16 00 00    DOSVER: DB    249,22,0,0,0,26
E604'  00 1A

000D                  CR      EQU   13
000A                  LF      EQU   10
0009                  TAB     EQU   9
0008                  BKSP    EQU   8
0020                  SPACE   EQU   20H

                      ;  ====================
                      ;  ! 7.2 Componenta BDOS !
                      ;  ====================

                              ORG   0E606H

E606'                 BBDOS:
E606'  C3 E611'               JP    BSTART          ;Punct de intrare actual in BDOS
                      ;patru erori fatale de procedura
E609'                 ERR01:                        ;BAD SECTOR (sector eronat)
E609'  E699'                  DW    EPRC1
E60B'                 ERR02:                        ;SELECT ERROR (eroare selectie
E60B'  E6A5'                  DW    EPRC2
E60D'                 ERR03:                        ;DSK R/O (protectie la scriere)
E60D'  E6AB'                  DW    EPRC3
E60F'                 ERR04:                        ;FILE R/O (fisier nemodificabil)
E60F'  E6B1'                  DW    EPRC4
E611'                 BSTART:
E611'  EB                     EX    DE,HL
E612'  22 E943'               LD    (FCBSAVE),HL    ;Salveaza adresa FCB (sau DMA)
E615'  EB                     EX    DE,HL           ;Restaureaza in D,E adresa FCB
E616'  7B                     LD    A,E             ;Preia octetul
E617'  32 F3D6'               LD    (DRVFLG),A      ;si-l salveaza
E61A'  21 0000                LD    HL,0            ;Anuleaza
E61D'  22 E945'               LD    (RESULT),HL     ;rezultatul
E620'  39                     ADD   HL,SP           ;si-l salveaza
E621'  22 E90F'               LD    (SPSAVE),HL     ;drept indicator stiva utilizator
E624'  31 E941'               LD    SP,LOCSTACK     ;Preia indicatorul de stiva local
E627'  AF                     XOR   A               ;anuleaza
E628'  32 F3E0'               LD    (TSPCDSK),A     ;discul specificat
E62B'  32 F3DE'               LD    (DSKINV),A      ;si indicatorul fisier implicat
E62E'  21 F374'               LD    HL,EXISEQ       ;Pregateste adresa de revenire
E631'  E5                     PUSH  HL              ;si-o depune in stiva
E632'  79                     LD    A,C             ;Incarca codul functiei si
E633'  FE 29                  CP    41              ;testeaza daca este valid
E635'  D0                     RET   NC              ;Intoarce secventa daca codul = 41
E636'  4B                     LD    C,E             ;Salveaza octetul
E637'  21 E647'               LD    HL,ADDTAB       ;Incarca tabela de acces indirect
E63A'  5F                     LD    E,A             ;Cod functie
E63B'  16 00                  LD    D,0
E63D'  19                     ADD   HL,DE
E63E'  19                     ADD   HL,DE           ;Calculeaza
E63F'  5E                     LD    E,(HL)          ;preia adresa CMPS
E640'  23                     INC   HL              ;si
E641'  56                     LD    D,(HL)          ;adresa MS (mai semnificativa)
E642'  2A E943'               LD    HL,(FCBSAVE)    ;Reface adresa FCB in H,L
E645'  EB                     EX    DE,HL           ;interschimba D,E cu adresa FCB
E646'  E9                     JP    (HL)            ;salt la adresa functiei
                      ;Aceasta tabela are toate adresele functiilor BDOS
```

```
E647'                   ADDTAB:
                        ;
                        ;       Nume functie            Cod     Descriere
                        ;       ************            ***     *********
                        ;
E647'  F403'                    DW      WBOOT           ;0      restaurare sistem
E649'  E8C8'                    DW      CONIN           ;1      intrare consola
E64B'  E790'                    DW      CONOUT          ;2      iesire consola
E64D'  E8CE'                    DW      READINP         ;3      intrare cititor
E64F'  F412'                    DW      PUNCH           ;4      iesire perforator
E651'  F40F'                    DW      LISTOUT         ;5      iesire listare
E653'  E8D4'                    DW      DIRCON          ;6      consola (direct)
E655'  E8ED'                    DW      GETIOB          ;7      incarca IOBYTE
E657'  E8F3'                    DW      SETIOB          ;8      modifica IOBYTE
E659'  E8F8'                    DW      PRNBUF          ;9      afiseaza zona tampon
E65B'  E7E1'                    DW      INPBUF          ;10     incarca zona tampon
E65D'  E8FE'                    DW      CNSTAT          ;11     stare consola
E65F'  F27E'                    DW      GETVERS         ;12     intoarce versiunea
E661'  F283'                    DW      DOSINT          ;13     initializare DOS
E663'  F245'                    DW      SELDSK          ;14     selectie disc
E665'  F29C'                    DW      OPEN            ;15     deschide fisier
E667'  F2A5'                    DW      CLOSE           ;16     inchide fisier
E669'  F2AB'                    DW      SRCFIRST        ;17     cauta prima intrare
E66B'  F2C8'                    DW      SRCNEXT         ;18     cauta urmatoarea intrare
E66D'  F2D7'                    DW      ERASE           ;19     sterge fisier
E66F'  F2E0'                    DW      READREC         ;20     citeste sector
E671'  F2E6'                    DW      WRITEREC        ;21     scrie sector
E673'  F2EC'                    DW      MAKEFILE        ;22     creaza fisier
E675'  F2F5'                    DW      RENAME          ;23     renumeste fisier
E677'  F2FE'                    DW      LOGVECT         ;24     intoarce vector conectare
E679'  F304'                    DW      LOGDSK          ;25     intoarce disc conectat
E67B'  F30A'                    DW      STDMA           ;26     pune adresa DMA
E67D'  F311'                    DW      ALLVEC          ;27     intoarce adresa
                                                        ;       vector alocare
E67F'  EB2C'                    DW      PROTECT         ;28     protejeaza disc
E681'  F317'                    DW      ROVECT          ;29     intoarce vector numai
                                                        ;       citire
E683'  F31D'                    DW      ATTRIB          ;30     pune atribute fisiere
E685'  F326'                    DW      GETDPB          ;31     intoarce adresa DPB
E687'  F32D'                    DW      GPUSER          ;32     ia/pune cod utilizator
E689'  F341'                    DW      RANREAD         ;33     citire aleatoare
E68B'  F347'                    DW      RANWRITE        ;34     scriere aleatoare
E68D'  F34D'                    DW      GETSIZE         ;35     intorc dimensiune fisier
E68F'  F20E'                    DW      SELRND          ;36     selecteaza adresa
                                                        ;       aleatoare
E691'  F353'                    DW      LOGOFF          ;37     deselecteaza disc
E693'  E904'                    DW      NULLSUB         ;38     nu-i tratat
E695'  E904'                    DW      NULLSUB         ;39     nu-i tratat
E697'  F39B'                    DW      RNDERA          ;40     sterge arbore (umple cu
                                                        ;       zerouri)
                        ;
E699'                   EPRC1:
E699'  21 E6CA'                 LD      HL,MESS2        ;indicator sector eronat
E69C'  CD E6E5'                 CALL    DISPERR         ;tiparesc mesaj de eroare si astept
E69F'  FE 03                    CP      3               ;caracter. Test, daca-i ^C,
E6A1'  CA 0000                  JP      Z,0             ;restartare sistem
E6A4'  C9                       RET
E6A5'                   EPRC2:
E6A5'  21 E6D5'                 LD      HL,MESS3        ;indicator selectie eroare
E6A8'  C3 E6B4'                 JP      PRN$ERR$ABO     ;tipareste eroare, asteapta consola
                                                        ;si restartare
```

```
E6AB'                   EPRC3:
E6AB'  21 E6E1'                 LD      HL,MESS5    ;indicator eroare numai citire
E6AE'  C3 E6B4'                 JP      PRN$ERR$ABO ;tipareste eroare, asteapta consola
                                                    ;si restartare
E6B1'                   EPRC4:
E6B1'  21 E6DC'                 LD      HL,MESS4
E6B4'                   PRN$ERR$ABO:
E6B4'  CD E6E5'                 CALL    DISPERR     ;tipareste eroare, asteapta consola
                                                    ;si restartare
E6B7'  C3 0000                  JP      0           ;restartare

E6BA'  42 44 4F 53      MESS0:  DB      'BDOS ERR ON '
E6BE'  20 45 52 52
E6C2'  20 4F 4E 20
E6C6'  20 3A 20 24      MESS1:  DB      ' : $'
E6CA'  42 41 44 20      MESS2:  DB      'BAD SECTOR$'
E6CE'  53 45 43 54
E6D2'  4F 52 24
E6D5'  53 45 4C 45      MESS3:  DB      'SELECT$'
E6D9'  43 54 24
E6DC'  46 49 4C 45      MESS4:  DB      'FILE '
E6E0'  20
E6E1'  52 2F 4F 24      MESS5:  DB      'R/O$'

                        ;       *****   ******
E6E5'                   DISPERR:
E6E5'  E5                       PUSH    HL              ;Salveaza adresa mesaj
E6E6'  CD E7C9'                 CALL    CRLF            ;trimite CR, LF
E6E9'  3A E942'                 LD      A,(DSKSPEC)     ;incarca discul implicat
E6EC'  C6 41                    ADD     A,'A'           ;il converteste in ASCII
E6EE'  32 E6C6'                 LD      (MESS1),A       ;si-l memoreaza la adresa mesaj
E6F1'  01 E6BA'                 LD      BC,MESS0        ;incarca adresa primului bloc de
E6F4'  CD E7D3'                 CALL    BUFPRN          ;mesaje si-l tipareste
E6F7'  C1                       POP     BC              ;reface adresa mesaj
E6F8'  CD E7D3'                 CALL    BUFPRN          ;si iarasi tipareste
                        ;       *****   ******
E6FB'                   LOCCI:                          ;intrare consola locala
E6FB'  21 E90E'                 LD      HL,CHRBUF       ;zona tampon un caracter
E6FE'  7E                       LD      A,(HL)          ;test daca este vreunul
E6FF'  36 00                    LD      (HL),0          ;anuleaza zona tampon
E701'  B7                       OR      A               ;zero daca nu-i caracter
E702'  C0                       RET     NZ              ;revenire daca nu, altfel
E703'  C3 F409'                 JP      BCONIN          ;salt in BIOS pentru preluare
                                                        ;caracter
                        ;       *****   ******
E706'                   LCICO:                  ;consola locala pentru intrare/iesire
E706'  CD E6FB'                 CALL    LOCCI   ;intoarce caracter din zona tampon sau
E709'  CD E714'                 CALL    TSTPRN  ;CONIN si se uita daca-i tiparibil
E70C'  D8                       RET     C       ;revenire daca nu-i, altfel
E70D'  F5                       PUSH    AF      ;salveaza caracter
E70E'  4F                       LD      C,A     ;mutindu-l in registrul C
E70F'  CD E790'                 CALL    CONOUT  ;si tiparindu-l
E712'  F1                       POP     AF      ;reface caracter
E713'  C9                       RET

                        ;       *****   ******
E714'                   TSTPRN:         ;test pentru caracter afisabil sau functie speciala
E714'  FE 0D                    CP      CR              ;este CARRIAGE RETURN?
E716'  C8                       RET     Z               ;DA, revenire
E717'  FE 0A                    CP      LF              ;sau poate-i LINE FEED
E719'  C8                       RET     Z
E71A'  FE 09                    CP      TAB             ;sau TAB?
```

```
E71C  C8                      RET   Z
E71D  FE 08                   CP    BKSP          ;BACK SPACE (un spatiu inapoi)?
E71F  C8                      RET   Z
E720  FE 20                   CP    SPACE         ;sau chiar SPACE (spatiu)?
E722  C9                      RET                 ;CARRY=1 <=> caracter netiparibil
                        ;     *****  ******
E723                    LCSTCI:                   ;stare consola locala
E723  3A E90E                 LD    A,(CHRBUF)    ;preia caracterul din zona tampon
E726  B7                      OR    A             ;si testeaza daca-i
E727  C2 E745                 JP    NZ,LCST2      ;caracter valid. Da, salt
E72A  CD F406                 CALL  BCNSST        ;altfel apeleaza stare consola
E72D  E6,01                   AND   1             ;din BIOS si se uita daca avem
E72F  C8                      RET   Z             ;caracter. Daca nu, revenire
E730  CD F409                 CALL  BCONIN        ;altfel il preia
E733  FE 13                   CP    19            ;este de tip stop tiparire (^S)?
E735  C2 E742                 JP    NZ,LCST1      ;Nu, salt
E738  CD F409                 CALL  BCONIN        ;asteapta un caracter oarecare
E73B  FE 03                   CP    3             ;este de tip restartare (^C)?
E73D  CA 0000                 JP    Z,0           ;Da, reinitializare sistem
E740  AF                      XOR   A             ;altfel anuleaza A
E741  C9                      RET                 ;si revine
E742                    LCST1:
E742  32 E90E                 LD    (CHRBUF),A    ;salveaza caracter in zona tampon
E745                    LCST2:
E745  3E 01                   LD    A,1           ;intoarce indicator
E747  C9                      RET
                        ;     *****  ******
E748                    LOCCON:                   ;iesire consola locala
E748  3A E90A                 LD    A,(DELCOSW)   ;preia comutator stergere
E74B  B7                      OR    A             ;daca-i zero, atunci caracter
E74C  C2 E762                 JP    NZ,NOTPR      ;afisabil, altfel se va sterge
E74F  C5                      PUSH  BC            ;salveaza caracter (in reg.C)
E750  CD E723                 CALL  LCSTCI        ;apeleaza stare consola
E753  C1                      POP   BC
E754  C5                      PUSH  BC            ;si salveaza din nou
E755  CD F40C                 CALL  BCONOUT       ;afiseaza caracter via BIOS
E758  C1                      POP   BC            ;restaureaza si
E759  C5                      PUSH  BC            ;salveaza din nou
E75A  3A E90D                 LD    A,(HRDCOPY)   ;test daca-i in curs hardcopy
E75D  B7                      OR    A
E75E  C4 F40F                 CALL  NZ,LISTOUT    ;listare iesire
E761  C1                      POP   BC
E762                    NOTPR:
E762  79                      LD    A,C           ;preia caracter
E763  21 E90C                 LD    HL,LINCNT     ;indicator pentru numar linie
E766  FE 7F                   CP    127           ;este RUB OUT (stergere)?
E768  C8                      RET   Z             ;Da, revenire
E769  34                      INC   (HL)          ;altfel incrementeaza numar linie
E76A  FE 20                   CP    SPACE         ;este tiparibil?
E76C  D0                      RET   NC            ;revenire daca da
E76D  35                      DEC   (HL)          ;altfel decrementeaza numar linie
E76E  7E                      LD    A,(HL)        ;incarca numar linie
E76F  B7                      OR    A             ;este zero?
E770  C8                      RET   Z             ;revenire pentru linie goala
E771  79                      LD    A,C           ;este BACK SPACE?
E772  FE 08                   CP    BKSP          ;salt daca-i alt caracter
E774  C2 E779                 JP    NZ,NOTAB      ;neafisabil
E777  35                      DEC   (HL)          ;decrementeaza numar linie
E778  C9                      RET
E779                    NOTAB:
E779  FE 0A                   CP    LF            ;este linie noua?
```

```
E77B' C0          RET   NZ          ;Nu, revenire
E77C' 36 00       LD    (HL),0      ;Da, anuleaza contor caractere pe
E77E' C9          RET               ;linie
                  ;     *****  ******
E77F'             CRTLCH:           ;Afiseaza caracter si daca nu-i
                                    ;afisabil ii aduna 'A
E77F' 79          LD    A,C         ;preia caracter
E780' CD E714'    CALL  TSTPRN      ;este tiparibil?
E783' D2 E790'    JP    NC,CONOUT   ;Daca da, tipareste-l
E786' F5          PUSH  AF          ;altfel salveaza-l
E787' 0E 5E       LD    C,94        ;incarca "^"
E789' CD E748'    CALL  LOCCON      ;si tipareste
E78C' F1          POP   AF          ;reface caracter
E78D' F6 40       OR    40H         ;converteste in litere mari
E78F' 4F          LD    C,A         ;si-l duce in registrul C
                  ;     *****  ******
E790'             CONOUT:           ;tipareste caracterul din reg.C
E790' 79          LD    A,C         ;ducindu-l in reg.A
E791' FE 09       CP    TAB         ;este TAB?
E793' C2 E748'    JP    NZ,LOCCON   ;Nu, salt ->consola locala iesire
E796'             TABEXP:           ;extinde TAB
E796' 0E 20       LD    C,SPACE     ;incarca si
E798' CD E748'    CALL  LOCCON      ;afiseaza spatii
E79B' 3A E90C'    LD    A,(LINCNT)  ;pina cind numarul de linie
E79E' E6 07       AND   7           ;modulo 8 este zero
E7A0' C2 E796'    JP    NZ,TABEXP   ;Nu-i inca zero, mai tiparesc un
E7A3' C9          RET               ;spatiu
                  ;     *****  ******
E7A4'             BSSPBS:           ;sterge ultimul caracter de pe
E7A4' CD E7AC'    CALL  BSPRN       ;ecran prin tiparire BACK SPACE
E7A7' 0E 20       LD    C,SPACE     ;caracterul SPACE
E7A9' CD F40C'    CALL  BCONOUT     ;acum...
                  ;     *****  ******  si inca o data
E7AC'             BSPRN:
E7AC' 0E 08       LD    C,BKSP      ;tipareste
E7AE' C3 F40C'    JP    BCONOUT     ;BACK SPACE
                  ;     *****  ******
E7B1'             RLPOZ:            ;tipareste # la sfirsitul liniei
E7B1' 0E 23       LD    C,'#'       ;curente si restaureaza cursorul
E7B3' CD E748'    CALL  LOCCON      ;pe linia urmatoare
E7B6' CD E7C9'    CALL  CRLF
E7B9'             SETPOZ:
E7B9' 3A E90C'    LD    A,(LINCNT)  ;incarca numar linie si test cu
E7BC' 21 E90B'    LD    HL,IBLCNT   ;pozitia liniei de pe ecran
E7BF' BE          CP    (HL)        ;introdusa
E7C0' D0          RET   NC          ;Daca sint egale sau numarul
E7C1' 0E 20       LD    C,SPACE     ;liniei este mai mare incarca
E7C3' CD E748'    CALL  LOCCON      ;si tipareste spatii pina se
E7C6' C3 E7B9'    JP    SETPOZ      ;atinge limita specificata
                  ;     *****  ******
E7C9'             CRLF:
E7C9' 0E 0D       LD    C,CR        ;incarca si
E7CB' CD E748'    CALL  LOCCON      ;tipareste CARRIAGE RETURN
E7CE' 0E 0A       LD    C,LF        ;incarca si
E7D0' C3 E748'    JP    LOCCON      ;tipareste LINE FEED
                  ;     *****  ******
E7D3'             BUFPRN:           ;tipareste tamponul adresat de registrele B,C
E7D3' 0A          LD    A,(BC)      ;preia caracter
E7D4' FE 24       CP    '$'         ;este marca sfirsit de
E7D6' C8          RET   Z           ;tampon ($)? Da, revenire
E7D7' 03          INC   BC          ;altfel trecem la urmatorul
```

```
E7D8' C5                PUSH    BC              ;caracter, salvam noua adresa
E7D9' 4F                LD      C,A
E7DA' CD E790'          CALL    CONOUT          ;tiparim caracter
E7DD' C1                POP     BC              ;refacem adresa
E7DE' C3 E7D3'          JP      BUFFPRN         ;si reluam bucla
                    ;incarca tamponul de date, de la consola
E7E1'                   INPBUF:
E7E1' 3A E90C'          LD      A,(LINCNT)      ;pozitia de linie curenta
E7E4' 32 E90B'          LD      (IBLCNT),A      ;pozitia de start a liniei din
E7E7' 2A E943'          LD      HL,(FCBSAVE)    ;zona tampon. Incarca adresa de
E7EA' 4E                LD      C,(HL)          ;inceput. Incarca in reg.C
E7EB' 23                INC     HL              ;numarul maxim de caractere, iar
E7EC' E5                PUSH    HL              ;in reg.H,L adresa de debut a
E7ED' 06 00             LD      B,0             ;tamponului de caractere
E7EF'                   FRSTPOZ:
E7EF' C5                PUSH    BC              ;salveaza contor caractere
E7F0' E5                PUSH    HL              ;si adresa curenta a tamponului
E7F1'                   BIREDO:
E7F1' CD E6FB'          CALL    LOCCI           ;preia un caracter de la consola
E7F4' E6 7F             AND     127             ;anuleaza bit paritate
E7F6' E1                POP     HL              ;reface adresa tampon
E7F7' C1                POP     BC              ;si contorul
E7F8' FE 0D             CP      CR              ;este CARRIAGE RETURN?
E7FA' CA E8C1'          JP      Z,BICRS         ;Da, trateaza-l
E7FD' FE 0A             CP      LF              ;este LINE FEED?
E7FF' CA E8C1'          JP      Z,BICRS         ;Da, salt
E802' FE 08             CP      BKSP            ;BACK SPACE?
E804' C2 E816'          JP      NZ,BINOBS       ;Nu-i, salt
E807' 78                LD      A,B             ;Da. Pentru primul
E808' B7                OR      A               ;caracter in tampon?
E809' CA E7EF'          JP      Z,FRSTPOZ       ;Da, prima pozitie
E80C' 05                DEC     B               ;altfel decrementeaza contor
E80D' 3A E90C'          LD      A,(LINCNT)      ;preia contorul pe linie si-l
E810' 32 E90A'          LD      (DELCOSW),A     ;salveaza pentru stergere
E813' C3 E870'          JP      PRNDEL  ;comutare. Sterge caracter de pe ecran
E816'                   BINOBS:
E816' FE 7F             CP      127             ;este RUB OUT (DEL)?
E818' C2 E826'          JP      NZ,BINRUB       ;Nu, salt
E81B' 78                LD      A,B             ;este primul caracter in tampon?
E81C' B7                OR      A
E81D' CA E7EF'          JP      Z,FRSTPOZ       ;Da, de la inceput
E820' 7E                LD      A,(HL)          ;altfel muta caracter din zona
E821' 05                DEC     B               ;tampon si decrementeaza contor
E822' 2B                DEC     HL              ;si adresa tampon
E823' C3 E8A9'          JP      BISVV           ;Salt la caracter ecou
E826'                   BINRUB:                 ;Acum RUB OUT
E826' FE 05             CP      5               ;Este ^E
E828' C2 E837'          JP      NZ,BINABO       ;Nu, mai cauta
E82B' C5                PUSH    BC              ;salveaza contor
E82C' E5                PUSH    HL              ;si pozitie
E82D' CD E7C9'          CALL    CRLF            ;sari la linie noua
E830' AF                XOR     A               ;careia ii anulezi contoarele
E831' 32 E90B'          LD      (IBLCNT),A      ;de pozitie
E834' C3 E7F1'          JP      BIREDO          ;Reluare
E837'                   BINABO:
E837' FE 10             CP      16              ;este ^P?
E839' C2 E848'          JP      NZ,BINHRD       ;Nu, mai cauta
E83C' E5                PUSH    HL              ;salveaza adresa
E83D' 21 E90D'          LD      HL,HRDCOPY      ;adresa cocomutator HARD COPY
F040' 3E 01             LD      A,1             ;pe care-l
F042' 96                SUB     (HL)
```

```
F043'  77                        LD    (HL),A        ;modifica
F044'  E1                        POP   HL            ;reface adresa
F045'  C3 EFEF'                  JP    FRSTPOZ       ;si din nou in bucla
F048'                    BINHRD:
F048'  FE 18                     CP    24            ;este 'X?
F04A'  C2 F05F'                  JP    NZ,BINCLR     ;Nu, mai insista
F04D'  E1                        POP   HL            ;reface stiva
F0'                      CLRLOOP:
E84E'  3A E90B'                  LD    A,(IBLCNT)    ;pozitia primei zone tampon
E851'  21 E90C'                  LD    HL,LINCNT     ;contorul pentru linie este mai
E854'  BE                        CP    (HL)          ;mare decit valoarea de start?
E855'  D2 E7E1'                  JP    NC,INPBUF     ;Nu mai sint caractere de sters
E858'  35                        DEC   (HL)          ;Decrementeaza valoarea de start
E859'  CD E7A4'                  CALL  BSSPBS        ;Sterge ultimul caracter de pe
E85C'  C3 E84E'                  JP    CLRLOOP       ;ecran si reia bucla
E85F'                    BINCLR:
E85F'  FE 15                     CP    21            ;ar putea fi 'U?
E861'  C2 E86B'                  JP    NZ,BINCU      ;Nici, salt
E864'  CD E7B1'                  CALL  RLPOZ         ;linie noua si restaureaza
E867'  E1                        POP   HL            ;pozitia de pe ecran. Reface
E868'  C3 E7E1'                  JP    INPBUF        ;stiva si reia preluarea
E86B'                    BINCU:
E86B'  FE 12                     CP    18            ;sau poate 'R?
E86D'  C2 E8A6'                  JP    NZ,BINORM     ;Nu, secventa normala
E870'                    PRNDEL:
E870'  C5                        PUSH  BC      ;salveaza contor
E871'  CD E7B1'                  CALL  RLPOZ   ;linie noua si restaurare pozitie de
E874'  C1                        POP   BC      ;inceput a zonei tampon. Restaureaza
E875'  E1                        POP   HL      ;contor si adresa
E876'  E5                        PUSH  HL      ;si le salveaza
E877'  C5                        PUSH  BC      ;din nou
E878'                    PRNLOOP:
E878'  78                        LD    A,B           ;preia contor
E879'  B7                        OR    A             ;este epuizat?
E87A'  CA E88A'                  JP    Z,BIDLO       ;Da, salt
E87D'  23                        INC   HL            ;incrementeaza adresa
E87E'  4E                        LD    C,(HL)        ;preia caracterul
E87F'  05                        DEC   B             ;si-l afiseaza.
E880'  C5                        PUSH  BC            ;Va afisa caractere din zona
E881'  E5                        PUSH  HL            ;tampon pina cind contorul de
E882'  CD E77F'                  CALL  CRTLCH        ;caractere devine zero
E885'  E1                        POP   HL            ;Reface adresa
E886'  C1                        POP   BC            ;si contorul
E887'  C3 E878'                  JP    PRNLOOP       ;Reia bucla
E88A'                    BIDLO:
E88A'  E5                        PUSH  HL            ;salveaza indicator zona tampon
E88B'  3A E90A'                  LD    A,(DELCOSW)   ;preia comutatorul de stergere
E88E'  B7                        OR    A             ;caracterul trebuie sters?
E88F'  CA E7F1'                  JP    Z,BIREDO      ;Nu, preia din nou
E892'  21 E90C'                  LD    HL,LINCNT     ;altfel salveaza in DELCOSW
E895'  96                        SUB   (HL)          ;numarul de caractere
E896'  32 E90A'                  LD    (DELCOSW),A   ;care trebuie sterse
E899'                    BIERA:
E899'  CD E7A4'                  CALL  BSSPBS        ;Sterge caracterul de pe ecran
E89C'  21 E90A'                  LD    HL,DELCOSW    ;incarca numarul de stergeri
E89F'  35                        DEC   (HL)          ;il decrementeaza
E8A0'  C2 E899'                  JP    NZ,BIERA      ;mai sint?
E8A3'  C3 E7F1'                  JP    BIREDO        ;Nu, iarasi intrare
E8A6'                    BINORM:
E8A6'  23                        INC   HL            ;incrementeaza adresa tampon
E8A7'  77                        LD    (HL),A        ;memoreaza caracterul
```

```
E8A8'  04                  INC     B               ;incrementeaza contorul
E8A9'                      BISVV:
E8A9'  C5                  PUSH    BC              ;salveaza contorul
E8AA'  E5                  PUSH    HL              ;si adresa tampon
E8AB'  4F                  LD      C,A             ;tipareste
E8AC'  CD E77F'            CALL    CRTLCH          ;caracterul specificat
E8AF'  E1                  POP     HL              ;reface adresa
E8B0'  C1                  POP     BC              ;si contorul
E8B1'  7E                  LD      A,(HL)          ;preia caracterul
E8B2'  FE 03               CP      3               ;este ^C?
E8B4'  78                  LD      A,B             ;muta numar octeti
E8B5'  C2 E8BD'            JP      NZ,BINOO        ;Salt daca nu-i ^C
E8B8'  FE 01               CP      1               ;Altfel test daca-i primul
E8BA'  CA 0000             JP      Z,0  ;caracter in linie, apoi restartare, altfel
E8BD'                      BINOO:
E8BD'  B9                  CP      C               ;Numarul a atins valoarea maxima?
E8BE'  DA E7EF'            JP      C,FRSTPOZ       ;Daca nu, intrare din nou, altfel
E8C1'                      BICRS:                  ;restaureaza adresa lungimii
E8C1'  E1                  POP     HL              ;zonei tampon
E8C2'  70                  LD      (HL),B          ;Memoreaza numar octeti
E8C3'  0E 0D               LD      C,CR            ;si afiseaza
E8C5'  C3 E748'            JP      LOCCON          ;CARRIAGE RETURN
E8C8'                      CONIN:
E8C8'  CD E706'            CALL    LCICO           ;Intrare/Iesire consola locala
E8CB'  C3 E901'            JP      SAVRET          ;Salveaza rezultat. Secventa re-
E8CE'                      READINP:                ;venire
E8CE'  CD F415'            CALL    BRDER           ;Apeleaza intrare cititor BIOS
E8D1'  C3 E901'            JP      SAVRET          ;Salveaza rezultat si iesire
E8D4'                      DIRCON:
E8D4'  79                  LD      A,C             ;Test daca este
E8D5'  3C                  INC     A               ;intrare la consola
E8D6'  CA E8E0'            JP      Z,CITRAN
E8D9'  3C                  INC     A               ;Test pentru stare consola
E8DA'  CA F406'            JP      Z,BCNSST
E8DD'  C3 F40C'            JP      BCONOUT         ;altfel afiseaza caracter
E8E0'                      CITRAN:
E8E0'  CD F406'            CALL    BCNSST          ;Stare consola
E8E3'  B7                  OR      A
E8E4'  CA F391'            JP      Z,RESUNF        ;Nu e gata nici un caracter
E8E7'  CD F409'            CALL    BCONIN          ;Preia caracter
E8EA'  C3 E901'            JP      SAVRET          ;Secventa revenire
                           ;
                           ;       Functia incarca octet I/E
                           ;       ************************
                           ;
E8ED'                      GETIOB:
E8ED'  3A 0003             LD      A,(3)
E8F0'  C3 E901'            JP      SAVRET
E8F3'                      SETIOB:
E8F3'  21 0003             LD      HL,3
E8F6'  71                  LD      (HL),C
E8F7'  C9                  RET
                           ;
                           ;       Functia afisare zona tampon
                           ;       ***************************
E8F8'                      PRNBUF:
E8F8'  EB                  EX      DE,HL
E8F9'  4D                  LD      C,L
E8FA'  44                  LD      B,H
E8FB'  C3 E7D3'            JP      BUFPRN
                           ;
```

```
                        ;               Functia stare consola
                        ;               *********************
                        ;
E8FE'                   CNSTAT:
E8FE'  CD E723'                 CALL    LCSTCI
                        ;
E901'                   SAVRET:
E901'  32 E945'                 LD      (RESULT),A
E904'                   NULLSUB:
E904'  C9                       RET
                        ;       *****   ******
E905'                   ONERET:
E905'  3E 01                    LD      A,1
E907'  C3 E901'                 JP      SAVRET
E90A'                   DELCOSW:    ;Sterge indicator. Caracterul zero pentru a fi afisat
E90A'  00                       DB      0       ;Altfel, numarul de caractere de sters
E90B'                   IBLCNT:             ;Pozitia liniei de iesire cind incepe 0
E90B'  00                       DB      0       ;functie de intrare in zona tampon
E90C'                   LINCNT:             ;Numarul liniei de iesire, de exemplu
E90C'  00                       DB      0       ;pozitia de pe ecran
E90D'                   HRDCOPY:            ;Comutator hard copy
E90D'  00                       DB      0       ;Zero=deconectat
E90E'                   CHRBUF:             ;Caracter zona tampon
E90E'  00                       DB      0
E90F'                   SPSAVE:             ;Salveaza indicator stiva utilizator
E90F'  0000                     DW      0
E911'                           DS      24*2    ;Rezerva loc pentru stiva
E941'                   LOCSTACK:
E941'                   USRCODE:            ;Cod utilizator
E941'  00                       DB      0
E942'                   DSKSPEC:            ;Disc specificat
E942'  00                       DB      0
E943'                   FCBSAVE:            ;Salveaza adresa FCB
E943'  0000                     DW      0
E945'                   RESULT:             ;Salveaza rezultat
E945'  0000                     DW      0
                        ;       *****   ******
E947'                   SELERR:
E947'  21 E60B'                 LD      HL,ERR02        ;Rutina selectare eroare
E94A'                   ERRBRU:
E94A'  5E                       LD      E,(HL)          ;Incarca adresa procedura eroare
E94B'  23                       INC     HL              ;CMPS si
E94C'  56                       LD      D,(HL)          ;CMS octet
E94D'  EB                       EX      DE,HL           ;mergi
E94E'  E9                       JP      (HL)            ;la rutina
                        ;       *****   ******
E94F'                   MVSRDH:                         ;Muta sir din (D,E) in (H,L)
E94F'  0C                       INC     C               ;
E950'                   MVSRLOOP:
E950'  0D                       DEC     C               ;Mai am de mutat?
E951'  C8                       RET     Z               ;Nu, revenire
E952'  1A                       LD      A,(DE)          ;preia caracterul
E953'  77                       LD      (HL),A          ;si-l muta
E954'  13                       INC     DE              ;urmatoarea sursa
E955'  23                       INC     HL              ;urmatoarea destinatie
E956'  C3 E950'                 JP      MVSRLOOP        ;mai sint?
                        ;       *****   ******
E959'                   LOAD$DPB:                       ;Incarca blocul parametrilor de
E959'  3A E942'                 LD      A,(DSKSPEC)     ;disc. Incarca si
E95C'  4F                       LD      C,A             ;salveaza discul specificat
E95D'  CD F41B'                 CALL    BSELDSK         ;Selecteaza discul BIOS
```

```
E960'  7C                LD    A,H          ;Test daca e
E961'  B5                OR    L            ;selectata eroare
E962'  C8                RET   Z            ;Revenire daca H,L=0
E963'  5E                LD    E,(HL)       ;altfel incarca in registrul
E964'  23                INC   HL           ;D,E tabela de
E965'  56                LD    D,(HL)       ;translatare
E966'  23                INC   HL
E967'  22 F3B3'          LD    (SCRH0),HL   ;Salveaza adresa SCRH0,
E96A'  23                INC   HL
E96B'  23                INC   HL
E96C'  22 F3B5'          LD    (SCRH1),HL   ;adresa SCRH1
E96F'  23                INC   HL
E970'  23                INC   HL
E971'  22 F3B7'          LD    (SCRH2),HL   ;si adresa SCRH2
E974'  23                INC   HL
E975'  23                INC   HL           ;Urmatoarea adresa DPB
E976'  EB                EX    DE,HL
E977'  22 F3D0'          LD    (TRANADD),HL ;Salveaza adresa tabelei de
E97A'  21 F3B9'          LD    HL,DIRBUF    ;translatare. Salveaza, din BIOS,
E97D'  0E 08             LD    C,8          ;DIRBUF, DPBASE si adresele cim-
E97F'  CD E94F'          CALL  MVSRDH       ;purilor de verificare si alocare
E982'  2A F3BB'          LD    HL,(DSKPBL)  ;Acum transfera
E985'  EB                EX    DE,HL        ;din BIOS
E986'  21 F3C1'          LD    HL,SECTRK    ;parametri de disc
E989'  0E 0F             LD    C,15         ;in zona tampon
E98B'  CD E94F'          CALL  MVSRDH       ;a componentei BDOS
E98E'  2A F3C6'          LD    HL,(DSKSIZE) ;Incarca dimensiunea discului
E991'  7C                LD    A,H          ;Test daca este
E992'  21 F3DD'          LD    HL,SIZEFLG   ;mai mare decit 255
E995'  36 FF             LD    (HL),255     ;Daca da, pune indicatorul
E997'  B7                OR    A            ;de dimensiune la zero
E998'  CA E99D'          JP    Z,SDSFLG     ;altfel pune #FF
E99B'  36 00             LD    (HL),0
E99D'                 SDSFLG:
E99D'  3E FF             LD    A,255        ;Anuleaza
E99F'  B7                OR    A            ;indicatorul Z
E9A0'  C9                RET                ;Nu se selecteaza eroare
                      ;  *****  ******
E9A1'                 SCRHOME:              ;Initializare zona 'SCRATCH'
E9A1'  CD F418'          CALL  BHOME        ;Intoarce in BIOS
E9A4'  AF                XOR   A            ;Anuleaza acumulator
E9A5'  2A F3B5'          LD    HL,(SCRH1)   ;Zona 'SCRATCH'1...
E9A8'  77                LD    (HL),A
E9A9'  23                INC   HL
E9AA'  77                LD    (HL),A
E9AB'  2A F3B7'          LD    HL,(SCRH2)   ;si zona 'SCRATCH'2
E9AE'  77                LD    (HL),A
E9AF'  23                INC   HL
E9B0'  77                LD    (HL),A       ;initializate cu zero
E9B1'  C9                RET
                      ;  *****  ******
E9B2'                 DKREAD:
E9B2'  CD F427'          CALL  BREAD
E9B5'  C3 E9BB'          JP    DKRESULT
                      ;  *****  ******
E9B8'                 DKWRITE:              ;Citire inregistrare disc
E9B8'  CD F42A'          CALL  BWRITE       ;Apeleaza BIOS
E9BB'                 DKRESULT:
E9BB'  B7                OR    A            ;Test daca este
E9BC'  C8                RET   Z            ;eroare permanenta
E9BD'  21 E609'          LD    HL,ERR01     ;Adresa procedura eroare
```

```
E9C0'  C3 E94A'          JP    ERRBRU
               ;         *****  ******
E9C3'                    DIRTRKSC:      ;Pune pista si sector pt.intrare in DIRECTORY
E9C3'  2A F3EA'          LD    HL,(DIRCOUNT)  ;Numar intrari in DIRECTORY
E9C6'  0E 02             LD    C,2            ;Imparte numar DIR cu 4
E9C8'  CD EAEA'          CALL  SHRHLC         ;4 intrari in fiecare inregistra-
E9CB'  22 F3E5'          LD    (SECTNB),HL    ;re pe disc. Salveaza numar inre-
E9CE'  22 F3EC'          LD    (CKCOUNT),HL   ;gistrare si numar control
               ;         *****  ******
E9D1'                    STTRKSEC:            ;Pune pista si sector
E9D1'  21 F3E5'          LD    HL,SECTNB      ;incarca numar inregistrare
E9D4'  4E                LD    C,(HL)         ;in registrul pereche
E9D5'  23                INC   HL             ;B,C
E9D6'  46                LD    B,(HL)
E9D7'  2A F3B7'          LD    HL,(SCRH2)     ;Incarca in
E9DA'  5E                LD    E,(HL)         ;registrul pereche D,E
E9DB'  23                INC   HL             ;rezultatul intermediar
E9DC'  56                LD    D,(HL)         ;de ex.:pista * sectoare pe pista
E9DD'  2A F3B5'          LD    HL,(SCRH1)     ;Incarca in
E9E0'  7E                LD    A,(HL)         ;registrul pereche H,L
E9E1'  23                INC   HL             ;pista implicata in
E9E2'  66                LD    H,(HL)         ;rezultatul intermediar
E9E3'  6F                LD    L,A            ;utilizat la marirea vitezei
E9E4'                    BACKWARD:            ;procedurii
E9E4'  79                LD    A,C            ;Test daca inregistrarea curenta
E9E5'  93                SUB   E              ;este < decit rezultatul interme-
E9E6'  78                LD    A,B            ;diar. Daca da, trebuie mers
E9E7'  9A                SBC   A,D            ;inapoi
E9E8'  D2 E9FA'          JP    NC,FOREWARD    ;altfel, inainte in piste
E9EB'  E5                PUSH  HL             ;Salveaza pista curenta
E9EC'  2A F3C1'          LD    HL,(SECTRK)    ;Numar de sectoare pe pista
E9EF'  7B                LD    A,E            ;Calculeaza rezultat intermediar
E9F0'  95                SUB   L              ;sectoare pe pista
E9F1'  5F                LD    E,A
E9F2'  7A                LD    A,D
E9F3'  9C                SBC   A,H
E9F4'  57                LD    D,A
E9F5'  E1                POP   HL             ;Restaureaza si
E9F6'  2B                DEC   HL             ;decrementeaza pista
E9F7'  C3 E9E4'          JP    BACKWARD       ;Mergi din nou
E9FA'                    FOREWARD:            ;Rezultat intermediar < decit
E9FA'  E5                PUSH  HL             ;inregistrarea. Salveaza pista
E9FB'  2A F3C1'          LD    HL,(SECTRK)    ;incarca sectoare pe pista
E9FE'  19                ADD   HL,DE          ;Urmatorul rezultat intermediar
E9FF'  DA EA0F'          JP    C,DONETRK      ;Daca este > decit OFFFFH
EA02'  79                LD    A,C            ;Test daca noul intermediar este
EA03'  95                SUB   L              ;> decit numarul de inregistrare.
EA04'  78                LD    A,B            ;de ex.:inregistrarea specificata
EA05'  9C                SBC   A,H            ;este pe pista curenta
EA06'  DA EA0F'          JP    C,DONETRK      ;Este gata
EA09'  EB                EX    DE,HL          ;altfel
EA0A'  E1                POP   HL             ;restaureaza si
EA0B'  23                INC   HL             ;incrementeaza pista
EA0C'  C3 E9FA'          JP    FOREWARD       ;Mergi din nou
EA0F'                    DONETRK:             ;Am gasit pista
EA0F'  E1                POP   HL             ;Restaureaza pista
EA10'  C5                PUSH  BC             ;Salveaza numarul de inregistrare
EA11'  D5                PUSH  DE             ;Rezultat intermediar
EA12'  E5                PUSH  HL             ;si pista
EA13'  EB                EX    DE,HL          ;D,E pista
EA14'  2A F3CE'          LD    HL,(TRKOFF)    ;Incarca si
```

```
EA17'  19                      ADD     HL,DE           ;aduna compensarea pentru pista
EA18'  44                      LD      B,H
EA19'  4D                      LD      C,L
EA1A'  CD F41E'                CALL    BSETTRK         ;Apeleaza 'SETTRK' din BIOS
EA1D'  D1                      POP     DE              ;Restaureaza pista
EA1E'  2A F3B5'                LD      HL,(SCRH1)      ;si
EA21'  73                      LD      (HL),E          ;salveaza in
EA22'  23                      INC     HL              ;zona SCRATCH 1
EA23'  72                      LD      (HL),D
EA24'  D1                      POP     DE              ;Restaureaza rezultatul interme-
EA25'  2A F3B7'                LD      HL,(SCRH2)      ;diar si
EA28'  73                      LD      (HL),E          ;salveaza in
EA29'  23                      INC     HL              ;zona SCRATCH 2
EA2A'  72                      LD      (HL),D
EA2B'  C1                      POP     BC              ;Restaureaza numar inregistrare
EA2C'  79                      LD      A,C             ;Calculeaza numarul
EA2D'  93                      SUB     E               ;de sector
EA2E'  4F                      LD      C,A             ;pe pista specificata
EA2F'  78                      LD      A,B             ;Sector=inregistrare-pista*
EA30'  9A                      SBC     A,D             ;sectoare pe pista
EA31'  47                      LD      B,A             ;Sector in registrul B,C
EA32'  2A F3D0'                LD      HL,(TRANADD)    ;Tabela de translatare
EA35'  EB                      EX      DE,HL
EA36'  CD F430'                CALL    BSTRAN          ;Translatare sector BIOS
EA39'  4D                      LD      C,L
EA3A'  44                      LD      B,H
EA3B'  C3 F421'                JP      BSETSEC         ;Pune sector BIOS
                       ;       *****   ******
                       ;
                       ;Aceasta subrutina calculeaza adresa arborelui in FCB
                       ;incepind cu inregistrarea curenta si extensia specificata
                       ;
EA3E'                  COMP$CLSTR$NB:                  ;Calculeaza numar de arbore in
EA3E'  21 F3C3'                LD      HL,BLKSHIFT     ;FCB. Incarca registrul C
EA41'  4E                      LD      C,(HL)          ;cu deplasarea blocului
EA42'  3A F3E3'                LD      A,(TMPCR)       ;Incarca inregistrarea curenta
EA45'                  CLTR$LOOP:
EA45'  B7                      OR      A               ;Anuleaza CARRY
EA46'  1F                      RRA                     ;A=A/2
EA47'  0D                      DEC     C               ;Decrementeaza deplasare bloc
EA48'  C2 EA45'                JP      NZ,CLTR$LOOP    ;din nou
EA4B'  47                      LD      B,A         ;Registrul B=CR/(2*deplasare bloc). De
   mutator HARD COPY
E840'  3E 01                   LD      A,1             ;pe care-l
E842'  96                      SUB     (HL)
E843'  77                      LD      (HL),A          ;modifica
E844'  E1                      POP     HL              ;reface adresa
E845'  C3 E7EF'                JP      FRSTPOZ         ;si din nou in bucla
E848'                  BINHRD:
E848'  FE 18                   CP      24              ;este 'X?
E84A'  C2 E85F'                JP      NZ,BINCLR       ;Nu, mai insista
E84D'  E1                      POP     HL              ;reface stiva
E84E EA54'   CA EA5C'          JP      Z,CLTR$END      ;Gata
EA57'  B7                      OR      A               ;anuleaza CARRY
EA58'  17                      RLA                     ;A=A*2
EA59'  C3 EA53'                JP      CLTR$EXT        ;Din nou
EA5C'                  CLTR$END:
EA5C'  80                      ADD     A,B             ;Aduna arbore in extensie
EA5D'  C9                      RET     ;Intoarce in reg.A o valoare egala cu CR/(nr. de
                       ;inregistrari pe arbore)+extensie*(nr.de extensii pe intrare DIR)
                       ;       *****   ******
```

```
                        ;
                        ;Aceasta rutina intoarce identificatorul de arbore dind
                        ;deplasamentul relativ al arborelui in FCB din reg.B,C
                        ;
EA5E'                   LOAD$CLUSTER:
EA5E'  2A E943'             LD      HL,(FCBSAVE)    ;Incarca adresa FCB
EA61'  11 0010              LD      DE,16           ;Start arbore
EA64'  19                   ADD     HL,DE           ;Acum in H,L
EA65'  09                   ADD     HL,BC           ;aduna deplasament
EA66'  3A F3DD'             LD      A,(SIZEFLG)     ;Test pentru identificatorul de
EA69'  B7                   OR      A               ;arbore pe octet, sau cuvint pe
EA6A'  CA EA71'             JP      Z,TWO$CLUSTER   ;doi octeti
EA6D'  6E                   LD      L,(HL)          ;Incarca identificatorul de arbo-
EA6E'  26 00                LD      H,0             ;re, CMPS octet. CMPS este zero
EA70'  C9                   RET                     ;H,L = identificator de arbore
EA71'                   TWO$CLUSTER:
EA71'  09                   ADD     HL,BC           ;Aduna deplasament (2 octeti pen-
EA72'  5E                   LD      E,(HL)          ;tru fiecare arbore). Incarca
EA73'  23                   INC     HL              ;CMPS  octet, apoi
EA74'  56                   LD      D,(HL)          ;si CMS pentru A
EA75'  EB                   EX      DE,HL           ;Muta in H,L
EA76'  C9                   RET
                        ;       *****   ******
                        ;
                        ;Aceasta rutina calculeaza si salveaza identificatorul
                        ;de arbore implicat in inregistrarea curenta si extensie
                        ;
EA77'                   COM$SAV$CLUST:
EA77'  CD EA3E'             CALL    COMP$CLSTR$NB   ;Calculeaza deplasamentul arbore-
EA7A'  4F                   LD      C,A             ;lui in FCB
EA7B'  06 00                LD      B,0
EA7D'  CD EA5E'             CALL    LOAD$CLUSTER    ;Incarca identificatorul de arbo-
EA80'  22 F3E5'             LD      (SECTNB),HL     ;re din FCB. Salveaza in sectorul
EA83'  C9                   RET                     ;NB
                        ;       *****   ******
                        ;
EA84'                   TEST$CLUST:                 ;Test pentru arbore activ
EA84'  2A F3E5'             LD      HL,(SECTNB)     ;Incarca identificator de arbore
EA87'  7D                   LD      A,L             ;CMPS si
EA88'  B4                   OR      H               ;CMS octet
EA89'  C9                   RET                     ;Z = 1 pentru arbore inactiv
                        ;       *****   ******
                        ;
                        ;Calculeaza numar inregistrare incepind de la
                        ;identificatorul de arbore si inregistrarea curenta. Inregistra-
                        ;rea=identificator arbore*inregistrare pe arbore+CR mascat
                        ;
EA8A'                   COMP$RECORD$NB:
EA8A'  3A F3C3'             LD      A,(BLKSHIFT)    ;Incarca deplasare bloc (2*inre-
EA8D'  2A F3E5'             LD      HL,(SECTNB)     ;gistrari pe arbore).
EA90'                   CLUST$MPY:                  ;Identificator de arbore
EA90'  29                   ADD     HL,HL           ;Arbore * 2
EA91'  3D                   DEC     A               ;se mai multiplica
EA92'  C2 EA90'             JP      NZ,CLUST$MPY    ;Dupa aceasta H,L=identificator
                                                    ;arbore*dimensiune arbore
EA95'  22 F3E7'             LD      (MPYCLST),HL    ;Salveaza rezultat temporar
EA98'  3A F3C4'             LD      A,(BLKMASK)     ;Incarca masca pentru bloc
EA9B'  4F                   LD      C,A             ;in registrul C
EA9C'  3A F3E3'             LD      A,(TMPCR)       ;Inregistrare curenta
EA9F'  A1                   AND     C               ;Numar de masca in arbore
EAA0'  B5                   OR      L               ;Calculeaza
```

```
EAA1'  6F                    LD      L,A             ;H,L=numar de inregistrare
EAA2'  22 F3E5'              LD      (SECTNB),HL     ;Salveaza rezultat
EAA5'  C9                    RET
                     ;       *****   ******
                     ;
                     ;Aceasta rutina intoarce adresa extensie FCB
                     ;
EAA6'                FCB$EXT$ADD:
EAA6'  2A E943'              LD      HL,(FCBSAVE)    ;Adresa FCB
EAA9'  11 000C               LD      DE,12           ;Deplasament extensie
EAAC'  19                    ADD     HL,DE
EAAD'  C9                    RET
                     ;       *****   ******
                     ;
                     ;Aceasta rutina intoarce in H, L adresa FCB-CR si
                     ;in D,E adresa FCB-RC
                     ;
EAAE'                CR$RC$ADD:
EAAE'  2A E943'              LD      HL,(FCBSAVE)    ;Adresa FCB
EAB1'  11 000F               LD      DE,15           ;Deplasament RC
EAB4'  19                    ADD     HL,DE
EAB5'  EB                    EX      DE,HL
EAB6'  21 0011               LD      HL,17           ;CR din deplasament RC
EAB9'  19                    ADD     HL,DE
EABA'  C9                    RET
                     ;       *****   ******
                     ;
                     ;Aceasta rutina salveaza CR, RC si extensia
                     ;mascata (extensia in intrarea DIR curenta)
                     ;
EABB'                SV$CR$RC$EX:
EABB'  CD EAAE'              CALL    CR$RC$ADD       ;Adresele CR si RC
EABE'  7E                    LD      A,(HL)
EABF'  32 F3E3'              LD      (TMPCR),A       ;Memoreaza CR
EAC2'  EB                    EX      DE,HL
EAC3'  7E                    LD      A,(HL)
EAC4'  32 F3E1'              LD      (TMPRC),A       ;Memoreaza RC
EAC7'  CD EAA6'              CALL    FCB$EXT$ADD
EACA'  3A F3C5'              LD      A,(EXTMASK)
EACD'  A6                    AND     (HL)            ;Masca extesie
EACE'  32 F3E2'              LD      (MEXT),A        ;Memoreaza extensia mascata
EAD1'  C9                    RET
                     ;       *****   ******
                     ;
                     ;Aceasta rutina incrementeaza inregistrarea curenta
                     ;
EAD2'                INCR$CR:
EAD2'  CD EAAE'              CALL    CR$RC$ADD       ;Adresa CR si RC
EAD5'  3A F3D5'              LD      A,(MODSWTCH)    ;Incarca comutatorul de mod
EAD8'  FE 02                 CP      2               ;2 pentru acces direct (fara
EADA'  C2 EADE'              JP      NZ,INCR00       ;increment )
EADD'  AF                    XOR     A               ;Curata increment
EADE'                INCR00:
EADE'  4F                    LD      C,A             ;Salveaza increment
EADF'  3A F3E3'              LD      A,(TMPCR)       ;Incarca CR BDOS
EAE2'  81                    ADD     A,C             ;aduna increment
EAE3'  77                    LD      (HL),A          ;si salveaza in FCB-CR
EAE4'  EB                    EX      DE,HL
EAE5'  3A F3E1'              LD      A,(TMPRC)       ;Incarca RC, BDOS
EAE8'  77                    LD      (HL),A          ;salveaza in FCB-RC
EAE9'  C9                    RET
```

```
                        ;       *****   ******
                        ;
                        ;Aceasta rutina roteste la dreapta registrul pereche
                        ;H,L de un numar de ori specificat in registrul C
                        ;
EAEA'                   SHRHLC:
EAEA'   0C                      INC     C
EAEB'                   SHRLOOP:
EAEB'   0D                      DEC     C               ;Test pentru numar de deplasari
EAEC'   C8                      RET     Z               ;Gata
EAED'   7C                      LD      A,H             ;Incarca din registrul H
EAEE'   B7                      OR      A               ;anuleaza CARRY
EAEF'   1F                      RRA                     ;deplaseaza la dreapta
EAF0'   67                      LD      H,A             ;restaureaza H
EAF1'   7D                      LD      A,L             ;la fel cu registrul L
EAF2'   1F                      RRA
EAF3'   6F                      LD      L,A
EAF4'   C3 EAEB'                JP      SHRLOOP         ;Din nou
                        ;       *****   ******
                        ;
                        ;Aceasta rutina calculeaza octetul de control
                        ;pentru o intrare in DIRECTORY
                        ;
EAF7'                   COM$CHECK:
EAF7'   0E 80                   LD      C,128           ;Lungime DIRECTORY
EAF9'   2A F3B9'                LD      HL,(DIRBUF)     ;Adresa zona tampon DIRECTORY
EAFC'   AF                      XOR     A               ;anuleaza control initial
EAFD'                   CK$LOOP:
EAFD'   86                      ADD     A,(HL)          ;Aduna octetul curent
EAFE'   23                      INC     HL              ;Urmatorul
EAFF'   0D                      DEC     C               ;Decrementeaza numar
EB00'   C2 EAFD'                JP      NZ,CK$LOOP      ;Mai sint octeti? Da, salt
EB03'   C9                      RET
                        ;       *****   ******
                        ;
                        ;Aceasta rutina roteste la stinga registrul pereche
                        ;H,L de un numar de ori dat de registrul C
                        ;
EB04'                   MPY$HL$C:
EB04'   0C                      INC     C
EB05'                   HL$LOOP:
EB05'   0D                      DEC     C               ;Numar test
EB06'   C8                      RET     Z
EB07'   29                      ADD     HL,HL           ;Rotire stinga
EB08'   C3 EB05'                JP      HL$LOOP         ;Din nou
                        ;       *****   ******
                        ;
                        ;Aceasta rutina actualizeaza vectorul de conectare
                        ;din B,C cu drive-ul specificat. Rezultatul in H,L
                        ;
EB0B'                   ACT$LOGGIN:
EB0B'   C5                      PUSH    BC              ;Salveaza vechea conectare
EB0C'   3A E942'                LD      A,(DSKSPEC)     ;discul specificat
EB0F'   4F                      LD      C,A             ;in reg.C
EB10'   21 0001                 LD      HL,1            ;Bit disc conectat
EB13'   CD EB04'                CALL    MPY$HL$C        ;Deplaseaza la locatia corespun-
EB16'   C1                      POP     BC              ;zatoare. Restaureaza vechiul
EB17'   79                      LD      A,C             ;vector de conectare. Muta in
EB18'   B5                      OR      L               ;reg.A. Mascheaza noul disc
EB19'   6F                      LD      L,A             ;conectat
EB1A'   78                      LD      A,B
```

```
EB1B'  B4                      OR      H
EB1C'  67                      LD      H,A             ;si salveaza rezultatul in pere-
EB1D'  C9                      RET                     ;chea H,L
                       ;       *****   ******
                       ;
                       ;Aceasta rutina incarca si testeaza vectorul
                       ;"numai citire" pentru discul specificat
                       ;
EB1E'                  TST$RO$VEC:
EB1E'  2A F3AD'                LD      HL,(ROMARK)     ;Incarca vectorul "numai citire"
EB21'  3A E942'                LD      A,(DSKSPEC)     ;Incarca discul specificat
EB24'  4F                      LD      C,A
EB25'  CD EAEA'                CALL    SHRHLC          ;Vector deplasare
EB28'  7D                      LD      A,L             ;si masca
EB29'  E6 01                   AND     1               ;bit numai citire
EB2B'  C9                      RET                     ;Z = 0 pentru numai citire
                       ;       *****   ******
                       ;
                       ;Aceasta functie protejeaza discul specificat
                       ;
EB2C'                  PROTECT:
EB2C'  21 F3AD'                LD      HL,ROMARK       ;Incarca  vectorul  numai citire
EB2F'  4E                      LD      C,(HL)          ;Muta in
EB30'  23                      INC     HL              ;registrul pereche
EB31'  46                      LD      B,(HL)          ;B,C
EB32'  CD EB0B'                CALL    ACT$LOGGIN      ;calculeaza noul vector numai
EB35'  22 F3AD'                LD      (ROMARK),HL     ;citire. Salveaza
EB38'  2A F3C8'                LD      HL,(DIRMAX)     ;incarca
EB3B'  23                      INC     HL              ;DIRMAX
EB3C'  EB                      EX      DE,HL
EB3D'  2A F3B3'                LD      HL,(SCRHO)      ;SCRATCHO = numar de cautari
EB40'  73                      LD      (HL),E          ;in DIRECTORY
EB41'  23                      INC     HL              ;este pus la maxim
EB42'  72                      LD      (HL),D
EB43'  C9                      RET
                       ;       *****   ******
                       ;
                       ;Aceasta rutina testeaza daca fisierul este "numai citire"
                       ;
EB44'                  TST$FIL$RO:
EB44'  CD EB5E'                CALL    DIR$ENT$ADD     ;Incarca intrarile DIRECTORY
                       ;       *****   ******
EB47'                  FIL$RO$FLG:
EB47'  11 0009                 LD      DE,9            ;Atribut numai citire
EB4A'  19                      ADD     HL,DE
EB4B'  7E                      LD      A,(HL)          ;Incarca in A
EB4C'  17                      RLA                     ;CARRY = 0 daca nu e numai citire
EB4D'  D0                      RET     NC              ;altfel
EB4E'  21 E60F'                LD      HL,ERR04        ;eroare fatala
EB51'  C3 E94A'                JP      ERRBRU
                       ;       *****   ******
                       ;
                       ;Aceasta rutina testeaza daca discul este "numai citire"
                       ;
EB54'                  TST$DSK$RO:
EB54'  CD EB1E'                CALL    TST$RO$VEC      ;Test pentru "numai citire"
EB57'  C8                      RET     Z               ;Revenire daca nu
EB58'  21 E60D'                LD      HL,ERR03        ;altfel eroare fatala
EB5B'  C3 E94A'                JP      ERRBRU
                       ;       *****   ******
                       ;
```

```
                        ;Aceasta rutina intoarce adresa curenta a punctului de intrare
                        ;in DIRECTORY
                        ;
EB5E'                   DIR$ENT$ADD:
EB5E'  2A F3B9'                 LD      HL,(DIRBUF)     ;Incarca adresa zona tampon DIR
EB61'  3A F3E9'                 LD      A,(DIRDISP)     ;Deplasament pentru intrarea
                        ;       *****   ******          ;curenta
                        ;
                        ;Aceasta rutina aduna la H,L registrul A
                        ;
EB64'                   ADD$DISP:
EB64'  85                       ADD     A,L
EB65'  6F                       LD      L,A
EB66'  D0                       RET     NC
EB67'  24                       INC     H
EB68'  C9                       RET
                        ;       *****   ******
                        ;
                        ;Aceasta rutina intoarce FCB-S2
                        ;
EB69'                   LOAD$F$S2:
EB69'  2A E943'                 LD      HL,(FCBSAVE)    ;Incarca adresa FCB
EB6C'  11 000E                  LD      DE,14           ;Deplasament S2
EB6F'  19                       ADD     HL,DE
EB70'  7E                       LD      A,(HL)          ;Incarca
EB71'  C9                       RET
                        ;       *****   ******
                        ;
                        ;Aceasta rutina initializeaza FCB-S2
                        ;
EB72'                   CLR$F$S2:
EB72'  CD EB69'                 CALL    LOAD$F$S2       ;Incarca FCB-S2
EB75'  36 00                    LD      (HL),0          ;Pune zero in FCB-S2
EB77'  C9                       RET
                        ;       *****   ******
EB78'                   SRD$FCB$S2:                     ;Pune bit F din FCB-S2 pe 1,
EB78'  CD EB69'                 CALL    LOAD$F$S2       ;functie "numai citire"
EB7B'  F6 80                    OR      128
EB7D'  77                       LD      (HL),A
EB7E'  C9                       RET
                        ;       *****   ******
                        ;
                        ;Test daca intrarea curenta DIR este verificata
                        ;de exemplu, ca numarul intrarii este > decit DIR
EB7F'                   TST$VERY:
EB7F'  2A F3EA'                 LD      HL,(DIRCOUNT)   ;Incarca numar DIRECTORY
EB82'  EB                       EX      DE,HL           ;Muta in D,E
EB83'  2A F3B3'                 LD      HL,(SCRH0)      ;Incarca adresa SCRATC0 (numar
EB86'  7B                       LD      A,E             ;verificat). Numar DIR
EB87'  96                       SUB     (HL)            ;Scade CMPS verificat
EB88'  23                       INC     HL              ;Urmatorul ordin
EB89'  7A                       LD      A,D             ;si CMS
EB8A'  9E                       SBC     A,(HL)          ;Scade ordinul CMS si imprumuta
EB8B'  C9                       RET                     ;CARRY este pus pe 1 daca numaru
                        ;       *****   ******          ;verificat este mai mare
                        ;
                        ;Aceasta rutina schimba numarul verificat (SCRATC0) daca este
                        ;necesar
                        ;
EB8C'                   ACT$VERY:
EB8C'  CD EB7F'                 CALL    TST$VERY        ;Test daca SCRATC0 trebuie modi-
```

```
EB8F' D8                RET     C               ;ficat. Revenire daca nu
EB90' 13                INC     DE              ;altfel
EB91' 72                LD      (HL),D          ;memoreaza
EB92' 2B                DEC     HL              ;noua valoare
EB93' 73                LD      (HL),E          ;din D,E
EB94' C9                RET
                ;       *****   ******
                ;
                ;Aceasta rutina scade H,L din D,E si intoarce rezultatul in H,L
                ;(D,E-nemodificat )
                ;
EB95'           SUB$16:
EB95' 7B                LD      A,E
EB96' 95                SUB     L
EB97' 6F                LD      L,A
EB98' 7A                LD      A,D
EB99' 9C                SBC     A,H
EB9A' 67                LD      H,A
EB9B' C9                RET
                ;       *****   ******
                ;
EB9C'           COM$CK$VECT:            ;Verifica intrarile DIR
EB9C' 0E FF             LD      C,255   ;reg.C=0FFH-indicator de verificare
                ;
                ;Aceasta rutina genereaza/verifica vectorul de control DIRECTORY
                ;
EB9E'           CK$ENTRY:
EB9E' 2A F3EC'          LD      HL,(CKCOUNT)    ;Numar de control
EBA1' EB                EX      DE,HL           ;In registrul D,E
EBA2' 2A F3CC'          LD      HL,(CKSIZE)     ;verifica dimensiunea in H,L
EBA5' CD EB95'          CALL    SUB$16          ;Test daca nu mai sint intrari
EBA8' D0                RET     NC              ;de verificat. Revenire
EBA9' C5                PUSH    BC              ;Salveaza indicator functie
EBAA' CD EAF7'          CALL    COM$CHECK       ;Calculeaza octet de control
EBAD' 2A F3BD'          LD      HL,(CHCKZON)    ;Incarca baza vectorului de
EBB0' EB                EX      DE,HL           ;control in D,E
EBB1' 2A F3EC'          LD      HL,(CKCOUNT)    ;Incarca numar de control si
EBB4' 19                ADD     HL,DE           ;calculeaza adresa octetului de
EBB5' C1                POP     BC              ;control. Restaureaza indicator
EBB6' 0C                INC     C               ;functie. Test pentru verificare
EBB7' CA EBC4'          JP      Z,STORE$CK      ;Salt pentru generare
EBBA' BE                CP      (HL)            ;Test daca coincide cu valoarea
EBBB' C8                RET     Z           ;memorata. Revenire pentrucoincidenta
EBBC' CD EB7F'          CALL    TST$VERY        ;Test daca aceasta intrare este
EBBF' D0                RET     NC              ;verificata. Revenire daca nu
EBC0' CD EB2C'          CALL    PROTECT         ;Protejeaza discul
EBC3' C9                RET
EBC4'           STORE$CK:
EBC4' 77                LD      (HL),A          ;Restaureaza octetul de control
EBC5' C9                RET
                ;       *****   ******
                ;
                ;Scrie ZTD (Zona Tampon Directory)
                ;
EBC6'           WRT$DIR$REC:
EBC6' CD EB9C'          CALL    COM$CK$VECT     ;Vector de control
EBC9' CD EBE0'          CALL    ST$DIR$DMA      ;Pune DMA pentru zona tampon DIR
EBCC' 0E 01             LD      C,1             ;Indicator scriere DIRECTORY
EBCE' CD E9B8'          CALL    DKWRITE         ;Scrie inregistrare disc
EBD1' C3 EBDA'          JP      ST$USR$DMA      ;Restaureaza DMA utilizator
                ;       *****   ******
```

```
                        ;
                        ;Citeste ZTD
                        ;
EBD4'                   RD$DIR$REC:
EBD4'  CD EBE0'                 CALL    ST$DIR$DMA      ;pune adresa DMA pe ZTD
EBD7'  CD E9B2'                 CALL    DKREAD          ;citeste sector
                        ;       *****   ******
                        ;
                        ;Pune adresa DMA utilizator
                        ;
EBDA'                   ST$USR$DMA:
EBDA'  21 F3B1'                 LD      HL,USRDMA       ;indicator adresa DMA utilizator
EBDD'  C3 EBE3'                 JP      ST$COM$DMA
                        ;       *****   ******
EBE0'                   ST$DIR$DMA:
EBE0'  21 F3B9'                 LD      HL,DIRBUF       ;indicator pentru adresa DIR DMA
EBE3'                   ST$COM$DMA:
EBE3'  4E                       LD      C,(HL)
EBE4'  23                       INC     HL
EBE5'  46                       LD      B,(HL)
EBE6'  C3 F424'                 JP      BSETDMA
                        ;       *****   ******
                        ;
                        ;Transfera ZTD in zona tampon DMA
                        ;
EBE9'                   TRF$DIR$USR:
EBE9'  2A F3B9'                 LD      HL,(DIRBUF)     ;incarca sursa
EBEC'  EB                       EX      DE,HL           ;in reg.D,E
EBED'  2A F3B1'                 LD      HL,(USRDMA)     ;incarca adresa destinatie
EBF0'  0E 80                    LD      C,128           ;lungimea sirului transferat
EBF2'  C3 E94F'                 JP      MVSRDH          ;muta zona
                        ;       *****   ******
                        ;
                        ;Test daca s-a gasit intrarea. Pentru Z=1 nu s-a gasit
                        ;
EBF5'                   DIR$RESULT:
EBF5'  21 F3EA'                 LD      HL,DIRCOUNT     ;Incarca contorul directorilor
EBF8'  7E                       LD      A,(HL)          ;la inceput OFFFFH
EBF9'  23                       INC     HL              ;daca inca este OFFFFH
EBFA'  BE                       CP      (HL)            ;dupa cautare
EBFB'  C0                       RET     NZ              ;pune Z=1 si A=0
EBFC'  3C                       INC     A
EBFD'  C9                       RET
                        ;       *****   ******
                        ;
                        ;Anuleaza indicator numar DIR (numar de directori)
                        ;
EBFE'                   RESET$DIR:
EBFE'  21 FFFF                  LD      HL,0FFFFH
EC01'  22 F3EA'                 LD      (DIRCOUNT),HL
EC04'  C9                       RET
                        ;       *****   ******
                        ;
                        ;Inspecteaza intrarea DIRECTORY si verifica/genereaza vectorul de
                        ;control, daca reg.C = 0FFH/0
                        ;
EC05'                   VERY$DIR$REC:
EC05'  2A F3C8'                 LD      HL,(DIRMAX)     ;incarca numar maxim directori
EC08'  EB                       EX      DE,HL           ;in reg.D,E
EC09'  2A F3EA'                 LD      HL,(DIRCOUNT)   ;preia contorul curent directori
EC0C'  23                       INC     HL              ;mai avem unul
```

```
EC0D'  22 F3EA'          LD      (DIRCOUNT),HL   ;salveaza noul contor directori
EC10'  CD EB95'          CALL    SUB$16          ;mai sint
EC13'  D2 EC19'          JP      NC,MORE$DIR$EN  ;intrari DIRECTORY
EC16'  C3 EBFE'          JP      RESET$DIR       ;altfel anuleaza indicator directori
EC19'                    MORE$DIR$EN:
EC19'  3A F3EA'          LD      A,(DIRCOUNT)    ;preia CPMS octet din contor
EC1C'  E6 03             AND     3               ;directori si-l mascheaza
EC1E'  06 05             LD      B,5             ;2*5 = 32 lungime director
EC20'                    PMY32:
EC20'  87                ADD     A,A
EC21'  05                DEC     B
EC22'  C2 EC20'          JP      NZ,PMY32
EC25'  32 F3E9'          LD      (DIRDISP),A     ;deplasament intrare director in
EC28'  B7                OR      A               ;tampon. E prima intrare?
EC29'  C0                RET     NZ              ;Revenire daca nu-i
EC2A'  C5                PUSH    BC        ;altfel salveaza indicator functie
EC2B'  CD E9C3'          CALL    DIRTRKSC  ;(verifica daca-i OFFH). Citeste
EC2E'  CD EBD4'          CALL    RD$DIR$REC;directorul in zona tampon. Acum
EC31'  C1                POP     BC              ;restaureaza indicatorul
EC32'  C3 EB9E'          JP      CK$ENTRY  ;Verifica sau calculeaza octet de
                                           ;control intrare director
                         ;       *****   ******
                         ;
                         ;Aceasta rutina intoarce in CARRY bitul de alocare pentru arbore
                         ;din B,C (inspecteaza vectorul de alocare)
                         ;
EC35'                    ALLOC$ENTRY:
EC35'  79                LD      A,C     ;Incarca CMPS octet
EC36'  E6 07             AND     7       ;mascheaza 3 octeti (8 arbori
EC38'  3C                INC     A       ;intr-un vector de alocare). Numar de
EC39'  5F                LD      E,A     ;deplasari in reg.E
EC3A'  57                LD      D,A     ;si D
EC3B'  79                LD      A,C     ;restaureaza CMPS octet al
EC3C'  0F                RRCA             ;identificatorului de arbore
EC3D'  0F                RRCA
EC3E'  0F                RRCA             ;imparte la 8 (calculeaza adresa octet de
EC3F'  E6 1F             AND     31      ;alocare). Mascheaza bitii inutili
EC41'  4F                LD      C,A     ;salveaza in reg.C
EC42'  78                LD      A,B     ;preia CMPS octet
EC43'  87                ADD     A,A     ;il inmulteste cu 32...
EC44'  87                ADD     A,A
EC45'  87                ADD     A,A
EC46'  87                ADD     A,A
EC47'  87                ADD     A,A
EC48'  B1                OR      C       ;si mascheaza CMPS 5 biti
EC49'  4F                LD      C,A     ;salveaza adresa CMPS pentru octetul de
EC4A'  78                LD      A,B     ;alocare. Incarca CMS octet
EC4B'  0F                RRCA             ;il imparte
EC4C'  0F                RRCA             ;la 8
EC4D'  0F                RRCA
EC4E'  E6 1F             AND     31      ;il mascheaza
EC50'  47                LD      B,A             ;in reg.B,C avem identificator
EC51'  2A F3BF'          LD      HL,(ALLOCZON)   ;arbore/8. Baza vectorului de
EC54'  09                ADD     HL,BC           ;alocare. Adresa octet alocare
EC55'  7E                LD      A,(HL)          ;preia octet
EC56'                    SHL$ALLOC:
EC56'  07                RLCA                    ;il deplaseaza
EC57'  1D                DEC     E               ;si duce bitul de alocare
EC58'  C2 EC56'          JP      NZ,SHL$ALLOC    ;in CARRY
EC5B'  C9                RET
                         ;       *****   ******
```

```
                        ;
                        ;Aceasta rutina aloca arborele specificat
                        ;
EC5C'                   SET$BIT$ALLOC:
EC5C'  D5               PUSH    DE              ;Salveaza masca de alocare
EC5D'  CD EC35'         CALL    ALLOC$ENTRY     ;preia bitul de alocare
EC60'  E6 FE            AND     0FEH            ;neglijeaza ceilalti biti
EC62'  C1               POP     BC              ;reface masca
EC63'  B1               OR      C               ;marcheaza daca-i specificat
                        ;       *****   ******
                        ;
EC64'                   SHR$ALLOC:
EC64'  0F               RRCA                    ;deplaseaza octet
EC65'  15               DEC     D               ;la configuratia corespunzatoare
EC66'  C2 EC64'         JP      NZ,SHR$ALLOC
EC69'  77               LD      (HL),A          ;si memoreaza noul octet de
EC6A'  C9               RET                     ;alocare
                        ;       *****   ******
                        ;
                        ;Aceasta rutina inspecteaza intrarea curenta in DIRECTORY
                        ;si pune bitii de alocare pentru fiecare arbore valid
                        ;
EC6B'                   ALL$RES$CLSTS:
EC6B'  CD EB5E'         CALL    DIR$ENT$ADD     ;calculeaza adresa de intrare in
EC6E'  11 0010          LD      DE,16           ;director. Deplasamentul primului
EC71'  19               ADD     HL,DE           ;arbore. Adresa primului arbore
EC72'  C5               PUSH    BC              ;salveaza indicator C=1 alocat,
EC73'  0E 11            LD      C,17            ;iar C=0 arbore liber. Treaba
                                                ;asta se face pentru toti arborii
EC75'                   ALL$MORE$CLS:
EC75'  D1               POP     DE              ;restaureaza indicatorii in D,E
EC76'  0D               DEC     C               ;pentru arbore. Decrementeaza
EC77'  C8               RET     Z               ;contorul si revine daca nu mai
EC78'  D5               PUSH    DE              ;sint arbori. Salveaza din nou
EC79'  3A F3DD'         LD      A,(SIZEFLG)     ;indicator. Identificator arbore
EC7C'  B7               OR      A               ;pe un singur octet? Daca-i zero,
EC7D'  CA EC88'         JP      Z,CLSTR$TWO$BYT ;atunci doi octeti.
EC80'  C5               PUSH    BC              ;salveaza numarul
EC81'  E5               PUSH    HL              ;si indicatorul arborelui
EC82'  4E               LD      C,(HL)          ;preia identificator arbore in C
EC83'  06 00            LD      B,0             ;CMS este zero (doar un octet?
EC85'  C3 EC8E'         JP      CLSTR$CONT      ;Da, eviti secventa doi octeti
EC88'                   CLSTR$TWO$BYT:          ;identificator arbore de doi
EC88'  0D               DEC     C               ;octeti. Decrementeaza din nou
EC89'  C5               PUSH    BC              ;contorul si-l salveaza
EC8A'  4E               LD      C,(HL)          ;Preia CMPS
EC8B'  23               INC     HL              ;si
EC8C'  46               LD      B,(HL)          ;CMS octet
EC8D'  E5               PUSH    HL              ;salveaza identificator arbore
EC8E'                   CLSTR$CONT:
EC8E'  79               LD      A,C             ;este
EC8F'  B0               OR      B               ;arbore activ?
EC90'  CA EC9D'         JP      Z,NEXT$CLSTR    ;Nu, caut altul
EC93'  2A F3C6'         LD      HL,(DSKSIZE)    ;altfel
EC96'  7D               LD      A,L             ;test daca-i arbore valid
EC97'  91               SUB     C               ;de exemplu, mai mic decit
EC98'  7C               LD      A,H             ;dimensiunea discului
EC99'  98               SBC     A,B             ;daca-i valid
EC9A'  D4 EC5C'         CALL    NC,SET$BIT$ALLOC;face alocare sau eliberare,
                                                ;functie de continutul reg.D,E
                                                ;bitul corespunzator in vectorul
```

```
                                                ;de alocare
EC9D'                   NEXT$CLSTR:
EC9D'  E1                       POP     HL      ;preia urmatorul arbore
EC9E'  23                       INC     HL      ;il reface
EC9F'  C1                       POP     BC      ;si incrementeaza indicatorul
ECA0'  C3 EC75'                 JP      ALL$MORE$CLS ;reface contorul
                                ;       *****   ******          ;reiau
                                ;
                                ;Aceasta functie initializeaza vectorul de alocare si apoi cauta
                                ;in director si marcheaza bitii de alocare pentru arbori valizi
                                ;ocupati
                                ;
ECA3'                   INSP$DIR$ALL:
ECA3'  2A F3C6'                 LD      HL,(DSKSIZE)    ;preia dimensiune disc
ECA6'  0E 03                    LD      C,3             ;mascheaza contorul
ECA8'  CD EAEA'                 CALL    SHRHLC          ;calculeaza lungimea vectorului
ECAB'  23                       INC     HL              ;de alocare in reg.H,L
ECAC'  44                       LD      B,H             ;salveaza lungimea alocata
ECAD'  4D                       LD      C,L             ;in perechea B,C
ECAE'  2A F3BF'                 LD      HL,(ALLOCZON)   ;preia adresa vectorului de
                                                        ;alocare
ECB1'                   CLR$ALL$ZONE:
ECB1'  36 00                    LD      (HL),0          ;anuleaza octetul de alocare
ECB3'  23                       INC     HL              ;urmatorul
ECB4'  0B                       DEC     BC              ;decrementeaza contorul si-l
ECB5'  78                       LD      A,B             ;testeaza
ECB6'  B1                       OR      C               ;daca-i nul
ECB7'  C2 ECB1'                 JP      NZ,CLR$ALL$ZONE ;Nu, din nou
ECBA'  2A F3CA'                 LD      HL,(ALLDIR)     ;incarca vectorul de alocare
ECBD'  EB                       EX      DE,HL           ;pentru director in reg.D,E
ECBE'  2A F3BF'                 LD      HL,(ALLOCZON)   ;preia adresa vectorului de
ECC1'  73                       LD      (HL),E          ;alocare si-l marcheaza "ocupat"
ECC2'  23                       INC     HL              ;pe arborele
ECC3'  72                       LD      (HL),D          ;utilizat de fisierul director
ECC4'  CD E9A1'                 CALL    SCRHOME         ;
ECC7'  2A F3B3'                 LD      HL,(SCRH0)      ;preia adresa SCRATCH 0
ECCA'  36 03                    LD      (HL),3          ;si memoreaza numar de intrari
ECCC'  23                       INC     HL              ;director la 3
ECCD'  36 00                    LD      (HL),0          ;i.e.prima zona tampon director
                                                        ;are 4 intrari 0,1,2,3
ECCF'  CD EBFE'                 CALL    RESET$DIR       ;pune comutator director la
                                                        ;OFFFFH
ECD2'                   ALL$NEXT$ENTRY:
ECD2'  0E FF                    LD      C,255           ;valideaza calcul pentru
                                                        ;indicator octet control, pe <ON>
ECD4'  CD EC05'                 CALL    VERY$DIR$REC    ;ia intrare director si cauta
ECD7'  CD EBF5'                 CALL    DIR$RESULT      ;daca mai exista intrari director
ECDA'  C8                       RET     Z               ;Nu, revenire
                                ;Test pentru o intrare numita $... (pentru SUBMIT-are)
ECDB'  CD EB5E'                 CALL    DIR$ENT$ADD     ;incarca adresa intrare directori
ECDE'  3E E5                    LD      A,0E5H          ;intrare activa
ECE0'  BE                       CP      (HL)            ;testata
ECE1'  CA ECD2'                 JP      Z,ALL$NEXT$ENTRY;daca nu, urmatoarea
ECE4'  3A E941'                 LD      A,(USRCODE)     ;Codul utilizator
ECE7'  BE                       CP      (HL)            ;coincide?
ECE8'  C2 ECF6'                 JP      NZ,GOTO$NEXT$ENT;Nu, urmatoarea
ECEB'  23                       INC     HL              ;primul caracter
ECEC'  7E                       LD      A,(HL)          ;este
ECED'  D6 24                    SUB     '$'             ;'$'
ECEF'  C2 ECF6'                 JP      NZ,GOTO$NEXT$ENT;Nu, urmatorul
ECF2'  3D                       DEC     A               ;altfel incarca in reg.a OFFH
```

```
ECF3' 32 E945'          LD      (RESULT),A      ;si salveaza "SUBMIT"
ECF6'                   GOTO$NEXT$ENT:
ECF6' 0E 01             LD      C,1             ;indicator alocare
ECF8' CD EC6B'          CALL    ALL$RES$CLSTS   ;inspecteaza intrarea si alocarea
ECFB' CD EB8C'          CALL    ACT$VERY        ;actualizeaza numar versiune
ECFE' C3 ECD2'          JP      ALL$NEXT$ENTRY  ;urmatoarea
ED01'                   RET$FF$01:
ED01' 3A F3D4'          LD      A,(FNDFLG)
ED04' C3 E901'          JP      SAVRET
                        ;       *****   ******
                        ;
                        ;Aceasta functie testeaza daca extensia DIR (reg.C) coincide cu
                        ;extensia FCB (reg.A) mascata de masca de extensie
                        ;
ED07'                   EXT$MATCH:
ED07' C5                PUSH    BC              ;salveaza extensia DIR
ED08' F5                PUSH    AF              ;si extensia FCB
ED09' 3A F3C5'          LD      A,(EXTMASK)     ;preia masca extensiei
ED0C' 2F                CPL                     ;o complementeaza
ED0D' 47                LD      B,A             ;si-o salveaza
ED0E' 79                LD      A,C             ;Preia extensia din DIR
ED0F' A0                AND     B               ;o mascheaza
ED10' 4F                LD      C,A             ;si salveaza rezultatul
ED11' F1                POP     AF              ;reface extensia FCB
ED12' A0                AND     B               ;o masc...aza
ED13' 91                SUB     C               ;test
ED14' E6 1F             AND     31              ;coincidenta
ED16' C1                POP     BC              ;reface extensia DIR
ED17' C9                RET
                        ;       *****   ******
                        ;
                        ;Aceasta functie gaseste prima intrare din director care
                        ;corespunde cu specificatia FCB
                        ;Reg.C memoreaza lungimea portiunii de coincidenta
                        ;
ED18'                   FIND$FIRST:
ED18' 3E FF             LD      A,255           ;initializeaza indicator
ED1A' 32 F3D4'          LD      (FNDFLG),A      ;"negasibil"  cu 0FFH
ED1D' 21 F3D8'          LD      HL,TMPLEN       ;salveaza
ED20' 71                LD      (HL),C          ;compara lungimea
ED21' 2A E943'          LD      HL,(FCBSAVE)    ;preia adresa FCB
ED24' 22 F3D9'          LD      (TMPFCB),HL     ;o salveaza
ED27' CD EBFE'          CALL    RESET$DIR       ;pune pe 0 indicatorul DIR
ED2A' CD E9A1'          CALL    SCRHOME         ;
                        ;       *****   ******
ED2D'                   FIND$NEXT:
ED2D' 0E 00             LD      C,0             ;preia comutator verificare
ED2F' CD EC05'          CALL    VERY$DIR$REC    ;testeaza octetul de control si
                                                ;la coincidenta pune "numai
                                                ;citire"
ED32' CD EBF5'          CALL    DIR$RESULT      ;mai sint intrari?
ED35' CA ED94'          JP      Z,ENT$NOT$FND   ;Nu, intoarce cod de eroare
ED38' 2A F3D9'          LD      HL,(TMPFCB)     ;Preia adresa FCB
ED3B' EB                EX      DE,HL           ;in reg.D,E
ED3C' 1A                LD      A,(DE)          ;preia
ED3D' FE E5             CP      0E5H            ;si verifica daca a fost stersa
ED3F' CA ED4A'          JP      Z,THIS$ERA      ;Da, salt
ED42' D5                PUSH    DE              ;salveaza adresa
ED43' CD EB7F'          CALL    TST$VERY        ;Test de verificare punct intrare
ED46' D1                POP     DE              ;goleste stiva
ED47' D2 ED94'          JP      NC,ENT$NOT$FND  ;punct de intrare negasibil
```

```
ED4A'                   THIS$ERA:
ED4A'  CD EB5E'             CALL    DIR$ENT$ADD     ;intoarce adresa punctului de
                                                    ;intrare in director
ED4D'  3A F3D8'             LD      A,(TMPLEN)      ;preia lungimea
ED50'  4F                   LD      C,A             ;in reg.C
ED51'  06 00                LD      B,0             ;pozitia sirului
ED53'                   NXT$CHR$ENT:
ED53'  79                   LD      A,C             ;Test daca nu mai
ED54'  B7                   OR      A               ;sint caractere de comparat
ED55'  CA ED83'             JP      Z,ENTRY$FOUND   ;Da, intrare gasita
ED58'  1A                   LD      A,(DE)          ;Preia caracter din FCB
ED59'  FE 3F                CP      '?'             ;Este "*"?
ED5B'  CA ED7C'             JP      Z,FIND$CHR$NXT  ;Da, mergi la urmatorul caracter
ED5E'  78                   LD      A,B             ;altfel
ED5F'  FE 0D                CP      13              ;sari peste caracterul S1
ED61'  CA ED7C'             JP      Z,FIND$CHR$NXT  ;din FCB
ED64'  FE 0C                CP      12              ;Test pentru pozitia extensiei
ED66'  1A                   LD      A,(DE)          ;preia caracter
ED67'  CA ED73'             JP      Z,EXT$MTCH$TST  ;Da, mergi la secventa extensiei
ED6A'  96                   SUB     (HL)            ;altfel compara cu directorul
ED6B'  E6 7F                AND     127             ;anuleaza bitul de paritate
ED6D'  C2 ED2D'             JP      NZ,FIND$NEXT    ;nu coincid, cauta alt director
ED70'  C3 ED7C'             JP      FIND$CHR$NXT    ;preia urmatorul caracter
ED73'                   EXT$MTCH$TST:
ED73'  C5                   PUSH    BC              ;salveaza contorul
ED74'  4E                   LD      C,(HL)          ;incarca extensia DIR in reg.C
ED75'  CD ED07'             CALL    EXT$MATCH       ;Test coincidenta extensie
ED78'  C1                   POP     BC              ;reface contorul
ED79'  C2 ED2D'             JP      NZ,FIND$NEXT    ;nu coincid, cauta alta intrare
ED7C'                   FIND$CHR$NXT:
ED7C'  13                   INC     DE              ;urmatorul FCB
ED7D'  23                   INC     HL              ;urmatorul DIR
ED7E'  04                   INC     B               ;incrementeaza pozitia
ED7F'  0D                   DEC     C               ;decrementeaza contorul
ED80'  C3 ED53'             JP      NXT$CHR$ENT     ;din nou
ED83'                   ENTRY$FOUND:
ED83'  3A F3EA'             LD      A,(DIRCOUNT)    ;preia numarul de directori ai
ED86'  E6 03                AND     3               ;intrarii gasite. Salveaza codul
ED88'  32 E945'             LD      (RESULT),A      ;DIRECTORY
ED8B'  21 F3D4'             LD      HL,FNDFLG       ;preia
ED8E'  7E                   LD      A,(HL)          ;si modifica...
ED8F'  17                   RLA
ED90'  D0                   RET     NC
ED91'  AF                   XOR     A               ;indicatorul "negasit"
ED92'  77                   LD      (HL),A
ED93'  C9                   RET
ED94'                   ENT$NOT$FND:
ED94'  CD EBFE'             CALL    RESET$DIR       ;intrare negasita, pune pe 1
ED97'  3E FF                LD      A,255           ;indicatorul DIR si returneaza
ED99'  C3 E901'             JP      SAVRET          ;cod de eroare
                        ;   *****   ******
                        ;
                        ;Aceasta functie sterge intrarile specificate in director
                        ;
ED9C'                   DEL$ENTRY:
ED9C'  CD EB54'             CALL    TST$DSK$RO      ;disc numai pe citire?
ED9F'  0E 0C                LD      C,12            ;compara lungimea numai pentru
EDA1'  CD ED18'             CALL    FIND$FIRST      ;utilizator si nume fisier. Cauta
                                                    ;prima intrare
EDA4'                   DEL$NXT$ENT:
EDA4'  CD EBF5'             CALL    DIR$RESULT      ;Mai sint intrari?
```

```
EDA7'  C8                      RET.    Z               ;Nu, revenire
EDA8'  CD EB44'                CALL    TST$FIL$RO      ;Fisierul este numai citibil?
EDAB'  CD EB5E'                CALL    DIR$ENT$ADD     ;intoarce adresa intrare director
EDAE'  36 E5                   LD      (HL),0E5H       ;memoreaza masca de stergere
EDB0'  0E 00                   LD      C,0             ;anuleaza indicator arbore ocupat
EDB2'  CD EC6B'                CALL    ALL$RES$CLSTS   ;elibereaza toti arborii
EDB5'  CD EBC6'                CALL    WRT$DIR$REC     ;calculeaza noul octet de control
                                                       ;si scrie director
EDB8'  CD ED2D'                CALL    FIND$NEXT       ;cauta urmatoarea intrare
EDBB'  C3 EDA4'                JP      DEL$NXT$ENT     ;reluare
                       ;       *****   ******
                       ;
                       ;Aceasta rutina gaseste primul arbore liber si il pune pe ocupat
                       ;Intoarce identificatorul de arbore in reg.H,L
                       ;
EDBE'                  FIND$FREE:
EDBE'  50                      LD      D,B             ;preia primul arbore de controlat
EDBF'  59                      LD      E,C             ;din ultimul arbore alocat
EDC0'                  NEXT$FREE:
EDC0'  79                      LD      A,C             ;primul arbore in FCB
EDC1'  B0                      OR      B               ;sau nealocat
EDC2'  CA EDD1'                JP      Z,CLSTR$NOT$ZE  ;salt inapoi
EDC5'  0B                      DEC     BC              ;Cauta arborele anterior
EDC6'  D5                      PUSH    DE              ;salveaza urmatorul
EDC7'  C5                      PUSH    BC              ;si ultimul
EDC8'  CD EC35'                CALL    ALLOC$ENTRY     ;preia bitul de alocare
EDCB'  1F                      RRA                     ;si verifica daca nu-i
EDCC'  D2 EDEC'                JP      NC,MAKE$BUSY    ;CARRY, arbore liber, il declara
EDCF'  C1                      POP     BC              ;ocupat. Altfel reface
EDD0'  D1                      POP     DE              ;indicatorii
EDD1'                  CLSTR$NOT$ZE:
EDD1'  2A F3C6'                LD      HL,(DSKSIZE)    ;preia dimensiune disc
EDD4'  7B                      LD      A,E             ;si test daca indicatorul urmator
EDD5'  95                      SUB     L               ;este prea mare
EDD6'  7A                      LD      A,D             ;Indicatorul urmator
EDD7'  9C                      SBC     A,H             ;in reg.D,E
EDD8'  D2 EDF4'                JP      NC,NOT$FRE$CL   ;eroare daca este sub limita
EDDB'  13                      INC     DE              ;altfel incrementeaza
EDDC'  C5                      PUSH    BC              ;salveaza indicatorul ultim
EDDD'  D5                      PUSH    DE              ;si urmator
EDDE'  42                      LD      B,D
EDDF'  4B                      LD      C,E             ;muta la urmatorul indicator
EDE0'  CD EC35'                CALL    ALLOC$ENTRY     ;test bit alocare
EDE3'  1F                      RRA                     ;daca
EDE4'  D2 EDEC'                JP      NC,MAKE$BUSY    ;nu-i CARRY, arbore liber
EDE7'  D1                      POP     DE              ;altfel
EDE8'  C1                      POP     BC              ;restaureaza si din nou
EDE9'  C3 EDC0'                JP      NEXT$FREE
EDEC'                  MAKE$BUSY:
EDEC'  17                      RLA                     ;reface octetul de alocare
EDED'  3C                      INC     A               ;arbore ocupat
EDEE'  CD EC64'                CALL    SHR$ALLOC       ;inapoi la locatia corespunzatoare
EDF1'  E1                      POP     HL              ;in reg.H,L identificator arbore
EDF2'  D1                      POP     DE              ;goleste stiva
EDF3'  C9                      RET
EDF4'                  NOT$FRE$CL:
EDF4'  79                      LD      A,C             ;mai sint
EDF5'  B0                      OR      B               ;indicatori din urma?
EDF6'  C2 EDC0'                JP      NZ,NEXT$FREE    ;Mai, salt
EDF9'  21 0000                 LD      HL,0            ;altfel incarca rezultat eroare
EDFC'  C9                      RET
```

```
                        ;       *****   ******
                        ;
                        ;Aceasta rutina transfera toate zonele FCB in zona tampon DIR si
                        ;apoi scrie zona tampon DIRECTORY
                        ;
EDFD'                   WRT$FR$FCB:
EDFD'   0E 00                   LD      C,0             ;deplasament FCB
EDFF'   1E 20                   LD      E,32            ;lungimea de transferat
                        ;       *****   ******
EE01'                   WR$SEC$DIR:
EE01'   D5                      PUSH    DE              ;salveaza lungimea
EE02'   06 00                   LD      B,0
EE04'   2A E943'                LD      HL,(FCBSAVE)    ;adresa FCB
EE07'   09                      ADD     HL,BC           ;aduna deplasament
EE08'   EB                      EX      DE,HL           ;in D,E
EE09'   CD EB5E'                CALL    DIR$ENT$ADD     ;Preia adresa de intrare director
EE0C'   C1                      POP     BC              ;reface lungimea
EE0D'   CD E94F'                CALL    MVSRDH          ;muta la intrare director
EE10'                   WR$SEQ$TRKSC:
EE10'   CD E9C3'                CALL    DIRTRKSC        ;Calculeaza pista si sectorul
EE13'   C3 EBC6'                JP      WRT$DIR$REC     ;scrie sectorul director
                        ;       *****   ******
                        ;
                        ;Aceasta rutina redenumeste fisierele specificate
                        ;
EE16'                   STRT$REN$SEQ:
EE16'   CD EB54'                CALL    TST$DSK$RO      ;discul este "numai citire"
EE19'   0E 0C                   LD      C,12            ;compara lungimea utilizator+nume
EE1B'   CD ED18'                CALL    FIND$FIRST      ;cauta prima intrare
EE1E'   2A E943'                LD      HL,(FCBSAVE)    ;preia adresa FCB
EE21'   7E                      LD      A,(HL)          ;Preia vechiul fisier utilizator
EE22'   11 0010                 LD      DE,16
EE25'   19                      ADD     HL,DE
EE26'   77                      LD      (HL),A          ;si salveaza in noul fisier
EE27'                   NEXT$REN:
EE27'   CD EBF5'                CALL    DIR$RESULT      ;test daca s-a gasit intrare
EE2A'   C8                      RET     Z               ;Nu, revenire
EE2B'   CD EB44'                CALL    TST$FIL$RO      ;fisier "numai citire"?
EE2E'   0E 10                   LD      C,16    ;deplasament FCB (aici incepe noul nume)
EE30'   1E 0C                   LD      E,12            ;transfera lungimea
EE32'   CD EE01'                CALL    WR$SEC$DIR      ;transfera din FCB in zona tampon
EE35'   CD ED2D'                CALL    FIND$NEXT       ;DIR si scrie. Cauta urmatoarea
EE38'   C3 EE27'                JP      NEXT$REN        ;intrare. Urmatoarea
                        ;       *****   ******
                        ;
                        ;Aceasta rutina pune noile atribute
                        ;
EE3B'                   STRT$ATTRIB:
EE3B'   0E 0C                   LD      C,12            ;compara lungimea
EE3D'   CD ED18'                CALL    FIND$FIRST      ;cauta intrarea
EE40'                   NEXT$ATTRIB:
EE40'   CD EBF5'                CALL    DIR$RESULT      ;a gasit-o?
EE43'   C8                      RET     Z
EE44'   0E 00                   LD      C,0             ;deplasament FCB
EE46'   1E 0C                   LD      E,12            ;transfera lungimea
EE48'   CD EE01'                CALL    WR$SEC$DIR      ;scrie noul director
EE4B'   CD ED2D'                CALL    FIND$NEXT       ;urmatoarea intrare
EE4E'   C3 EE40'                JP      NEXT$ATTRIB     ;Din nou
                        ;       *****   ******
                        ;
                        ;Aceasta rutina deschide o extensie
```

```
                        ;
EE51'                   STRT$OPEN:
EE51'  0E 0F                    LD      C,15            ;Compara lungimea
EE53'  CD ED18'                 CALL    FIND$FIRST      ;cauta prima intrare
EE56'  CD E3F5'                 CALL    DIR$RESULT      ;a gasit-o?
EE59'  C8                       RET     Z               ;Nu, revenire
                        ;       *****   ******
                        ;
EE5A'                   PREP$ENTRY:
EE5A'  CD EAA6'                 CALL    FCB$EXT$ADD     ;preia
EE5D'  7E                       LD      A,(HL)          ;si salveaza
EE5E'  F5                       PUSH    AF              ;extensia FCB
EE5F'  E5                       PUSH    HL              ;si adresa EXT FCB
EE60'  CD EB5E'                 CALL    DIR$ENT$ADD     ;preia adresa intrare director
EE63'  EB                       EX      DE,HL           ;in D,E
EE64'  2A E943'                 LD      HL,(FCBSAVE)    ;preia adresa FCB
EE67'  0E 20                    LD      C,20H           ;transfera lungimea
EE69'  D5                       PUSH    DE              ;salveaza adresa de intrare director
EE6A'  CD E94F'                 CALL    MVSRDH          ;transfer din DIR in FCB
EE6D'  CD EB78'                 CALL    SRD$FCB$S2      ;pune bit 7 din FCB-S2 (operatie
EE70'  D1                       POP     DE              ;citire-READ OP). Reface adresa
EE71'  21 000C                  LD      HL,12           ;DIR. Deplasament extensie
EE74'  19                       ADD     HL,DE           ;adresa in reg.H,L
EE75'  4E                       LD      C,(HL)          ;preia extensia DIR
EE76'  21 000F                  LD      HL,15           ;calculeaza
EE79'  19                       ADD     HL,DE           ;RC din adresa DIR
EE7A'  46                       LD      B,(HL)          ;preia DIR RC
EE7B'  E1                       POP     HL              ;reface adresa extensie FCB
EE7C'  F1                       POP     AF              ;si extensia FCB
EE7D'  77                       LD      (HL),A          ;salveaza inapoi in FCB
EE7E'  79                       LD      A,C             ;si compara
EE7F'  BE                       CP      (HL)            ;extensia DIR
EE80'  78                       LD      A,B             ;DIR RC
EE81'  CA EE8B'                 JP      Z,EQUAL$EXT     ;Daca extensia coincide cu cea
                                                        ;curenta, RC este din DIR
EE84'  3E 00                    LD      A,0             ;Daca extensia FCB > extensia DIR
                                                        ;atunci RC=0. I.e. extensie
                                                        ;negasita
EE86'  DA EE8B'                 JP      C,EQUAL$EXT     ;
EE89'  3E 80                    LD      A,128           ;altfel actualul RC=128
EE8B'                   EQUAL$EXT:
EE8B'  2A E943'                 LD      HL,(FCBSAVE)    ;preia adresa FCB
EE8E'  11 000F                  LD      DE,15           ;deplasament RC
EE91'  19                       ADD     HL,DE           ;si salveaza numar inregistrare
EE92'  77                       LD      (HL),A          ;curenta (RC-record count)
                                                        ;calculat
EE93'  C9                       RET
                        ;       *****   ******
                        ;
                        ;Urmeaza o rutina utilizata la incarcarea si verificarea
                        ;identificatorilor de arbore, pe doi octeti, in secventa inchisa
                        ;
EE94'                   TWO$BYTE$CLSTR:
EE94'  7E                       LD      A,(HL)
EE95'  23                       INC     HL
EE96'  B6                       OR      (HL)
EE97'  2B                       DEC     HL
EE98'  C0                       RET     NZ
EE99'  1A                       LD      A,(DE)
EE9A'  77                       LD      (HL),A
EE9B'  13                       INC     DE
```

```
EE9C'  23                      INC     HL
EE9D'  1A                      LD      A,(DE)
EE9E'  77                      LD      (HL),A
EE9F'  1B                      DEC     DE
EEA0'  2B                      DEC     HL
EEA1'  C9                      RET
                       ;       *****   ******
                       ;
                       ;Aceasta rutina inchide extensia curenta
                       ;
EEA2'                  CLOSE$ENTEN:
EEA2'  AF                      XOR     A               ;Anuleaza
EEA3'  32 E945'                LD      (RESULT),A      ;RESULT
EEA6'  32 F3EA'                LD      (DIRCOUNT),A    ;si numar directori
EEA9'  32 F3EB'                LD      (HIGHDIR),A
EEAC'  CD EB1E'                CALL    TST$RO$VEC      ;disc "numai citire"? (R/O)
EEAF'  C0                      RET     NZ              ;Da, revenire
EEB0'  CD EB69'                CALL    LOAD$F$S2       ;Este operatie
EEB3'  E6 80                   AND     128             ;R/O?
EEB5'  C0                      RET     NZ              ;Daca-i asa, revenire
EEB6'  0E 0F                   LD      C,15            ;Compara lungimea
EEB8'  CD ED18'                CALL    FIND$FIRST      ;cauta prima intrare
EEBB'  CD EBF5'                CALL    DIR$RESULT      ;Test daca-i gasita
EEBE'  C8                      RET     Z               ;Nu, revenire
EEBF'  01 0010                 LD      BC,16           ;Incarca
EEC2'  CD EB5E'                CALL    DIR$ENT$ADD     ;adresa primului arbore din
EEC5'  09                      ADD     HL,BC           ;director
EEC6'  EB                      EX      DE,HL           ;in reg. D,E
EEC7'  2A E943'                LD      HL,(FCBSAVE)    ;Incarca
EECA'  09                      ADD     HL,BC
EECB'  0E 10                   LD      C,16            ;adresa arborelui din FCB
EECD'                  CLSTR$LOOP$TR:
EECD'  3A F3DD'                LD      A,(SIZEFLG)     ;Este identificator arbore
EED0'  B7                      OR      A               ;pe un octet?
EED1'  CA EEE8'                JP      Z,CLOSE$THREE   ;pe doi octeti
EED4'  7E                      LD      A,(HL)          ;preia identificatorul arborelui
EED5'  B7                      OR      A               ;test daca arbore FCB activ
EED6'  1A                      LD      A,(DE)          ;preia arborele DIR
EED7'  C2 EEDB'                JP      NZ,CLOSE$ONE    ;arbore FCB inactiv
EEDA'  77                      LD      (HL),A          ;salveaza arbore DIR in FCB
EEDB'                  CLOSE$ONE:
EEDB'  B7                      OR      A               ;Arbore DIR
EEDC'  C2 EEE1'                JP      NZ,CLOSE$TWO    ;activ?
EEDF'  7E                      LD      A,(HL)          ;preia din FCB
EEE0'  12                      LD      (DE),A          ;memoreaza in DIR
EEE1'                  CLOSE$TWO:
EEE1'  BE                      CP      (HL)            ;Compara FCB si DIR
EEE2'  C2 EF1F'                JP      NZ,CLOSE$SIX$ERR;eroare la necoincidenta
EEE5'  C3 EEFD'                JP      CLOSE$FOUR      ;urmatorul
EEE8'                  CLOSE$THREE:
EEE8'  CD EE94'                CALL    TWO$BYTE$CLSTR  ;la fel ca mai sus
EEEB'  EB                      EX      DE,HL           ;dar pentru arbori pe doi octeti
EEEC'  CD EE94'                CALL    TWO$BYTE$CLSTR
EEEF'  EB                      EX      DE,HL
EEF0'  1A                      LD      A,(DE)
EEF1'  BE                      CP      (HL)
EEF2'  C2 EF1F'                JP      NZ,CLOSE$SIX$ERR
EEF5'  13                      INC     DE
EEF6'  23                      INC     HL
EEF7'  1A                      LD      A,(DE)
EEF8'  BE                      CP      (HL)
```

```
EEF9' C2 EF1F'          JP      NZ,CLOSE$SIX$ERR
EEFC' 0D                DEC     C
EEFD'                   CLOSE$FOUR:                     ;arborele urmator
EEFD' 13                INC     DE                      ;din DIR
EEFE' 23                INC     HL                      ;FCB
EEFF' 0D                DEC     C                       ;decrementeaza contor
EF00' C2 EECD'          JP      NZ,CLSTR$LOOP$TR;mai?
EF03' 01 FFEC           LD      BC,-14H                 ;calculeaza
EF06' 09                ADD     HL,BC                   ;adresa extensiei FCB
EF07' EB                EX      DE,HL                   ;in D,E
EF08' 09                ADD     HL,BC                   ;si DIR EXT in H,L
EF09' 1A                LD      A,(DE)                  ;Incarca extensia DIR
EF0A' BE                CP      (HL)                    ;Compara cu extensia FCB
EF0B' DA EF17'          JP      C,CLOSE$FIVE            ;salt daca extensia FCB>extensia
EF0E' 77                LD      (HL),A                  ;DIR (trebuie salvata). Altfel
EF0F' 01 0003           LD      BC,3                    ;salveaza extensie DIR in FCB.
EF12' 09                ADD     HL,BC                   ;Preia adresa FCB RC
EF13' EB                EX      DE,HL                   ;in D,E
EF14' 09                ADD     HL,BC                   ;si DIR RC in H,L
EF15' 7E                LD      A,(HL)                  ;Preia DIR-RC
EF16' 12                LD      (DE),A                  ;si salveaza in FCB-RC
EF17'                   CLOSE$FIVE:
EF17' 3E FF             LD      A,255                   ;Incarca indicator pentru
EF19' 32 F3D2'          LD      (EXCLFLG),A             ;extensie scrisa
EF1C' C3 EE10'          JP      WR$SEQ$TRKSC            ;scrie tampon DIR
EF1F'                   CLOSE$SIX$ERR:
EF1F' 21 E945'          LD      HL,RESULT               ;Eroare
EF22' 35                DEC     (HL)                    ;indicator
EF23' C9                RET
                        ;       *****   ******
                        ;
                        ;Rutina ce creaza o noua intrare DIRECTORY (extensie)
                        ;
EF24'                   MAKE$ENTRY:
EF24' CD EB54'          CALL    TST$DSK$RO              ;disc R/O?
EF27' 2A E943'          LD      HL,(FCBSAVE)            ;Incarca
EF2A' E5                PUSH    HL                      ;si salveaza adresa FCB
EF2B' 21 F3AC'          LD      HL,DELENT               ;Preia un FCB fals care incepe cu
EF2E' 22 E943'          LD      (FCBSAVE),HL            ;semnul "DELETE" (0E5H) si
                                                        ;salveaza adresa FCB.
EF31' 0E 01             LD      C,1                     ;Coincidenta lungime (numai 0E5H)
EF33' CD ED18'          CALL    FIND$FIRST              ;si cauta o intrare libera de
EF36' CD EBF5'          CALL    DIR$RESULT              ;director. Test daca-i gasita
EF39' E1                POP     HL                      ;reface FCB
EF3A' 22 E943'          LD      (FCBSAVE),HL            ;si memoreaza
EF3D' C8                RET     Z                       ;Revenire daca nu-i gasita (Z=1)
EF3E' EB                EX      DE,HL                   ;adresa FCB in D,E
EF3F' 21 000F           LD      HL,15                   ;adresa RC
EF42' 19                ADD     HL,DE                   ;in H,L
EF43' 0E 11             LD      C,17                    ;anuleaza lungimea (RC si arbori)
EF45' AF                XOR     A                       ;anuleaza reg.A
EF46'                   CLR$ENT$CLSTR:
EF46' 77                LD      (HL),A                  ;anuleaza RC
EF47' 23                INC     HL                      ;si zona arbore
EF48' 0D                DEC     C                       ;Mai?
EF49' C2 EF46'          JP      NZ,CLR$ENT$CLSTR;Da, reiau
EF4C' 21 000D           LD      HL,13                   ;calculeaza
EF4F' 19                ADD     HL,DE                   ;adresa S2
EF50' 77                LD      (HL),A                  ;si anuleaza
EF51' CD EB8C'          CALL    ACT$VERY                ;Test daca-i verificat'
EF54' CD EDFD'          CALL    WRT$FR$FCB              ;calculeaza, verifica si scrie in
```

```
                                                ;fisierul director
EF57' C3 EB78'                  JP      SRD$FCB$S2      ;pune comutator citire
                        ;       *****   ******
                        ;
                        ;Rutina ce da/creaza o noua extensie
                        ;
EF5A'                   NXT$EXT$SEQ:
EF5A' AF                        XOR     A               ;Anuleaza
EF5B' 32 F3D2'                  LD      (EXCLFLG),A     ;indicator inchidere
                                                        ;(0 = extensie neinchisa)
EF5E' CD EEA2'                  CALL    CLOSE$ENTEN     ;inchide extensia curenta
EF61' CD EBF5'                  CALL    DIR$RESULT      ;A reusit?
EF64' C8                        RET     Z               ;Revenire la eroare de inchidere
                                                        ;( Z=1 )
EF65' 2A E943'                  LD      HL,(FCBSAVE)    ;incarca adresa FCB
EF68' 01 000C                   LD      BC,12           ;calculeaza
EF6B' 09                        ADD     HL,BC           ;adresa extensiei FCB
EF6C' 7E                        LD      A,(HL)          ;si incarca actualul numar de
                                                        ;extensie
EF6D' 3C                        INC     A               ;Incrementeaza numarul extensiei
EF6E' E6 1F                     AND     31              ;intre 0...31 (maxim 32 extensii)
EF70' 77                        LD      (HL),A          ;salveaza in FCB
EF71' CA EF83'                  JP      Z,EXT$CARRY     ;daca-i mai mare ca 31, Z=1
EF74' 47                        LD      B,A             ;noul numar extensie in B
EF75' 3A F3C5'                  LD      A,(EXTMASK)     ;Incarca masca extensie
EF78' A0                        AND     B               ;masca in noua extensie
EF79' 21 F3D2'                  LD      HL,EXCLFLG      ;preia indicatorul "inchidere"
EF7C' A6                        AND     (HL)            ;si cu extensia mascata
EF7D' CA EF8E'                  JP      Z,EXT$SEARCH    ;extensie neinchisa sau noua
                                                        ;extensie globala
EF80' C3 EFAC'                  JP      EXT$MSK$OVR     ;nu-i o noua intrare in extensie
EF83'                   EXT$CARRY:
EF83' 01 0002                   LD      BC,2            ;Calculeaza
EF86' 09                        ADD     HL,BC           ;adresa S2
EF87' 34                        INC     (HL)            ;si incrementeaza (transport
EF88' 7E                        LD      A,(HL)          ;extensie. Test
EF89' E6 0F                     AND     15              ;pentru depasire
EF8B' CA EFB6'                  JP      Z,EXT$ERR$SEQ   ;eroare numar extensie prea mare
EF8E'                   EXT$SEARCH:
EF8E' 0E 0F                     LD      C,15            ;cauta
EF90' CD ED18'                  CALL    FIND$FIRST      ;urmatoarea extensie
EF93' CD EBF5'                  CALL    DIR$RESULT      ;A gasit-o?
EF96' C2 EFAC'                  JP      NZ,EXT$MSK$OVR  ;Da, OK
EF99' 3A F3D3'                  LD      A,(RDWRFLG)     ;extensie neidentificata, test
EF9C' 3C                        INC     A               ;operatie citire
EF9D' CA EFB6'                  JP      Z,EXT$ERR$SEQ   ;Daca-i citire, eroare
EFA0' CD EF24'                  CALL    MAKE$ENTRY      ;daca-i scriere creaza o noua
EFA3' CD EBF5'                  CALL    DIR$RESULT      ;intrare. A reusit?
EFA6' CA EFB6'                  JP      Z,EXT$ERR$SEQ   ;Eroare
EFA9' C3 EFAF'                  JP      EXT$EXIT$OK     ;Revenire
EFAC'                   EXT$MSK$OVR:
EFAC' CD EE5A'                  CALL    PREP$ENTRY      ;calculeaza R si CR
EFAF'                   EXT$EXIT$OK:
EFAF' CD EABB'                  CALL    SV$CR$RC$EX     ;memoreaza in locatiile de
EFB2' AF                        XOR     A               ;temporizare. Anuleaza indicator
EFB3' C3 E901'                  JP      SAVRET          ;de erori si revine
EFB6'                   EXT$ERR$SEQ:
EFB6' CD E905'                  CALL    ONERET          ;pune indicator eroare
EFB9' C3 EB78'                  JP      SRD$FCB$S2      ;pune S2 pentru citire si
                                                        ;revenire
```

```
                        ;Am ajuns si la rutina care citeste inregistrare!
                        ;
EFBC'                   STRT$READ$REC:
EFBC'  3E 01                    LD      A,1             ;pune indicatorul de
EFBE'  32 F3D5'                 LD      (MODSWTCH),A    ;acces secvential
                        ;       *****   ******
                        ;
EFC1'                   COMMON$READ:
EFC1'  3E FF                    LD      A,255           ;pozitioneaza indicatorul
EFC3'  32 F3D3'                 LD      (RDWRFLG),A     ;de citire
EFC6'  CD EABB'                 CALL    SV$CR$RC$EX     ;Salveaza RC, CR, EXT. in locatii
                                                        ;temporare
EFC9'  3A F3E3'                 LD      A,(TMPCR)       ;preia inregistrarea curenta
EFCC'  21 F3E1'                 LD      HL,TMPRC        ;si compara cu
EFCF'  BE                       CP      (HL)            ;numar inregistrare (RC)
EFD0'  DA EFE6'                 JP      C,NEW$EXTENSION ;Sare daca CR inca-i < RC
EFD3'  FE 80                    CP      128             ;Compara CR cu 128 (ultimul in
                                                        ;extensie)
EFD5'  C2 EFFB'                 JP      NZ,CLSTR$INACTV ;eroare arbore inactiv
EFD8'  CD EF5A'                 CALL    NXT$EXT$SEQ     ;daca-i ultimul in extensie, o ia
                                                        ;pe urmatoarea
EFDB'  AF                       XOR     A               ;anuleaza CR
EFDC'  32 F3E3'                 LD      (TMPCR),A
EFDF'  3A E945'                 LD      A,(RESULT)      ;si testeaza daca...
EFE2'  B7                       OR      A
EFE3'  C2 EFFB'                 JP      NZ,CLSTR$INACTV ;s-a gasit extensia
EFE6'                   NEW$EXTENSION:
EFE6'  CD EA77'                 CALL    COM$SAV$CLUST   ;preia identificatorul arborelui
EFE9'  CD EA84'                 CALL    TEST$CLUST      ;Este activ?
EFEC'  CA EFFB'                 JP      Z,CLSTR$INACTV  ;eroare daca nu-i
EFEF'  CD EA8A'                 CALL    COMP$RECORD$NB  ;Calculeaza numar inregistrare
EFF2'  CD E9D1'                 CALL    STTRKSEC        ;defineste pista si sectorul
EFF5'  CD E9B2'                 CALL    DKREAD          ;preia sector de pe disc
EFF8'  C3 EAD2'                 JP      INCR$CR         ;incrementeaza CR
EFFB'                   CLSTR$INACTV:
EFFB'  C3 E905'                 JP      ONERET          ;indicator eroare si revine
                        ;
                        ;Rutina de scriere inregistrare
                        ;
EFFE'                   STRT$WRITE:
EFFE'  3E 01                    LD      A,1             ;indicator de
F000'  32 F3D5'                 LD      (MODSWTCH),A    ;acces secvential
                        ;       *****   ******
                        ;
F003'                   COMMON$WRITE:
F003'  3E 00                    LD      A,0             ;salveaza
F005'  32 F3D3'                 LD      (RDWRFLG),A     ;indicatorul de scriere
F008'  CD EB54'                 CALL    TST$DSK$RO      ;disc R/O?
F00B'  2A E943'                 LD      HL,(FCBSAVE)    ;preia adresa FCB
F00E'  CD EB47'                 CALL    FIL$RO$FLG      ;fisier R/O?
F011'  CD EABB'                 CALL    SV$CR$RC$EX     ;preia si salveaza CR, RC si EXT
F014'  3A F3E3'                 LD      A,(TMPCR)       ;preia inregistrarea curenta
F017'  FE 80                    CP      128             ;si verifica daca-i valoare
F019'  D2 E905'                 JP      NC,ONERET       ;valida. Nu, indicator eroare si
F01C'  CD EA77'                 CALL    COM$SAV$CLUST   ;revenire. Preia identificatorul
F01F'  CD EA84'                 CALL    TEST$CLUST      ;arborelui. Este arbore activ?
F022'  0E 00                    LD      C,0             ;anuleaza indicatorul de alocare
F024'  C2 F06E'                 JP      NZ,WRT$ACT$CLS  ;al arborelui. Arbore nealocat.
F027'  CD EA3E'                 CALL    COMP$CLSTR$NB   ;Calculeaza numarul arborelui
F02A'  32 F3D7'                 LD      (CLSTNB),A      ;si-l salveaza
F02D'  01 0000                  LD      BC,0            ;Start identificator arbore
```

```
F030'  B7                        OR     A              ;pentru alocare. Daca este primul
F031'  CA F03B'                  JP     Z,WRT$001      ;in extensia curenta, sare
F034'  4F                        LD     C,A            ;altfel numar de arbore anterior
F035'  0B                        DEC    BC
F036'  CD EA5E'                  CALL   LOAD$CLUSTER   ;incarca START (pentru cautare)
F039'  44                        LD     B,H            ;de identificator arbore
F03A'  4D                        LD     C,L            ;in reg.B,C
F03B'                    WRT$001:
F03B'  CD EDBE'                  CALL   FIND$FREE      ;cauta arbore liber
F03E'  7D                        LD     A,L            ;test daca
F03F'  B4                        OR     H              ;s-a gasit
F040'  C2 F048'                  JP     NZ,WRT$002     ;arbore
F043'  3E 02                     LD     A,2            ;Nu, cod eroare (nu-i spatiu)
F045'  C3 E901'                  JP     SAVRET         ;si revenire
F048'                    WRT$002:
F048'  22 F3E5'                  LD     (SECTNB),HL    ;salveaza identificator arbore
F04B'  EB                        EX     DE,HL          ;in D,E
F04C'  2A E943'                  LD     HL,(FCBSAVE)   ;incarca adresa FCB
F04F'  01 0010                   LD     BC,16          ;adresa zonei
F052'  09                        ADD    HL,BC          ;arborelui
F053'  3A F3DD'                  LD     A,(SIZEFLG)    ;Este
F056'  B7                        OR     A              ;arbore pe un singur octet?
F057'  3A F3D7'                  LD     A,(CLSTNB)     ;incarca numar arbore
F05A'  CA F064'                  JP     Z,WRT$CLSTR$2  ;identificator arbore pe doi
F05D'  CD EB64'                  CALL   ADD$DISP       ;octeti. Calculeaza adresa arbore
F060'  73                        LD     (HL),E         ;in FCB. Salveaza identificator
F061'  C3 F06C'                  JP     WRT$003        ;arbore in FCB. Salt peste doi
                                                       ;octeti, secventa similara
F064'                    WRT$CLSTR$2:
F064'  4F                        LD     C,A            ;numar arbore
F065'  06 00                     LD     B,0            ;CMS este zero
F067'  09                        ADD    HL,BC
F068'  09                        ADD    HL,BC          ;Calculeaza adresa
F069'  73                        LD     (HL),E         ;si-i salveaza partea CMPS
F06A'  23                        INC    HL
F06B'  72                        LD     (HL),D         ;si CMS
F06C'                    WRT$003:
F06C'  0E 02                     LD     C,2            ;indicator alocare
F06E'                    WRT$ACT$CLS:
F06E'  3A E945'                  LD     A,(RESULT)     ;Este
F071'  B7                        OR     A              ;eroare?
F072'  C0                        RET    NZ             ;Revine daca nu-i 0
F073'  C5                        PUSH   BC             ;Salveaza indicator
F074'  CD EA8A'                  CALL   COMP$RECORD$NB ;Calculeaza numar inregistrare
F077'  3A F3D5'                  LD     A,(MODSWTCH)   ;preia comutator mod (0=aleator,
F07A'  3D                        DEC    A              ;1=direct, 2=arbore gol)
F07B'  3D                        DEC    A              ;si se uita daca-i 2
F07C'  C2 F0BB'                  JP     NZ,WRT$NORM    ;scriere secventiala
F07F'  C1                        POP    BC             ;reface indicator alocare
F080'  C5                        PUSH   BC             ;si-l mai salveaza odata
F081'  79                        LD     A,C            ;in reg.A
F082'  3D                        DEC    A              ;verifica daca-i
F083'  3D                        DEC    A              ;2
F084'  C2 F0BB'                  JP     NZ,WRT$NORM    ;Daca nu-i 2, scriere normala
F087'  E5                        PUSH   HL             ;Aceasta secventa incepe cu
                                                       ;scriere aleatoare cind se scrie
                                                       ;prima inregistrare in arbore
                                                       ;Salveaza adresa inregistrare
F088'  2A F3B9'                  LD     HL,(DIRBUF)    ;preia adresa zona tampon DIR
F08B'  57                        LD     D,A            ;incarca in reg.D 0
F08C'                    FILL$BUFFER:
```

```
F08C' 77                    LD      (HL),A          ;sterge tot
F08D' 23                    INC     HL              ;Adresa zona tampon DIR
F08E' 14                    INC     D               ;urmatoare
F08F' C3 F08C'              JP      FILL$BUFFER     ;Din nou daca (D)<128
F092' CD EBE0'              CALL    ST$DIR$DMA      ;Pune adresa DMA DIR
F095' 2A F3E7'              LD      HL,(MPYCLST)    ;incarca primul numar de
                                                    ;inregistrare in arborele curent
F098' 0E 02                 LD      C,2             ;incarca cod scriere (primul in
                                                    ;arbore)

                        ;
                        ;Aceasta secventa sterge toate inregistrarile din arbore
                        ;
F09A                    CLRT$REC$MSK:
F09A' 22 F3E5'              LD      (SECTNB),HL     ;salveaza numarul inregistrarii
F09D' C5                    PUSH    BC              ;salveaza indicatorii
F09E' CD E9D1'              CALL    STTRKSEC        ;pune pista si sectorul
F0A1' C1                    POP     BC              ;reface codul scrierii
F0A2' CD E9B8'              CALL    DKWRITE         ;scrie inregistrare
F0A5' 2A F3E5'              LD      HL,(SECTNB)     ;preia numarul inregistrarii
F0A8' 0E 00                 LD      C,0             ;Nu-i prima inregistrare-n arbore
F0AA' 3A F3C4'              LD      A,(BLKMASK)     ;preia masca bloc
F0AD' 47                    LD      B,A             ;si o salveaza in B
F0AE' A5                    AND     L               ;numar masca inregistrare
F0AF' B8                    CP      B               ;si test daca mai sint in
F0B0' 23                    INC     HL              ;arborele curent. Incrementeaza
F0B1' C2 F09A'              JP      NZ,CLRT$REC$MSK ;numar inregistrare. Din nou,
F0B4' E1                    POP     HL              ;urmatoarea inregistrare.
                                                    ;Restaureaza numar inregistrare.
F0B5' 22 F3E5'              LD      (SECTNB),HL     ;Salveaza
F0B8' CD EBDA'              CALL    ST$USR$DMA      ;Restaureaza adresa DMA
                                                    ;utilizator
F0BB                    WRT$NORM:
F0BB' CD E9D1'              CALL    STTRKSEC        ;pune pista si sector
F0BE' C1                    POP     BC              ;reface
F0BF' C5                    PUSH    BC              ;si salveaza codul scrierii
F0C0' CD E9B8'              CALL    DKWRITE         ;scrie sector
F0C3' C1                    POP     BC              ;reface codul scrierii
F0C4' 3A F3E3'              LD      A,(TMPCR)       ;preia CR
F0C7' 21 F3E1'              LD      HL,TMPRC        ;si
F0CA' BE                    CP      (HL)            ;test cu numar inregistrare (RC)
F0CB' DA F0D2'              JP      C,SKP$ACT$RC    ;Daca CR<RC sare ->actualizare RC
F0CE' 77                    LD      (HL),A          ;altfel salveaza CR
F0CF' 34                    INC     (HL)            ;si-l incrementeaza
F0D0' 0E 02                 LD      C,2             ;incarca primul in indicator
                                                    ;arbore
F0D2                    SKP$ACT$RC:
F0D2' 0D                    DEC     C               ;Test daca e primul in arbore
F0D3' 0D                    DEC     C               ;sau RC a fost =< CR
F0D4' C2 F0DF'              JP      NZ,NO$CR$MOD    ;Daca nu
F0D7' F5                    PUSH    AF              ;salveaza CR
F0D8' CD EB69'              CALL    LOAD$F$S2       ;Preia FCB S2 si anuleaza bitul 7
F0DB' E6 7F                 AND     127             ;(indicator operatie scriere),
                                                    ;pentru a semnala operatie
                                                    ;scriere pentru secventa inchisa
F0DD' 77                    LD      (HL),A          ;memoreaza FCB
F0DE' F1                    POP     AF              ;reface  CR
F0DF                    NO$CR$MOD:
F0DF' FE 7F                 CP      127             ;Test daca CR este ultimul in
                                                    ;extensia curenta. Daca nu, pune
F0E1' C2 F100'              JP      NZ,WRT$EX2      ;RC si CR in FCB, revine si
                                                    ;incrementeaza CR daca-i necesar
```

```
F0E4'  3A F3D5'          LD      A,(MODSWTCH)    ;Test
F0E7'  FE 01             CP      1               ;daca-i scriere secventiala
F0E9'  C2 F100'          JP      NZ,WRT$EX2      ;Nu-i secventiala, salt
F0EC'  CD EAD2'          CALL    INCR$CR         ;Este, incrementeaza CR (ultimul)
F0EF'  CD EF5A'          CALL    NXT$EXT$SEQ     ;e necesara o noua extensie
F0F2'  21 E945'          LD      HL,RESULT       ;Test daca-i
F0F5'  7E                LD      A,(HL)          ;eroare
F0F6'  B7                OR      A
F0F7'  C2 F0FE'          JP      NZ,WRT$EX1      ;Daca nu-i eroare
F0FA'  3D                DEC     A               ;Altfel pune in CR 0FFH
F0FB'  32 F3E3'          LD      (TMPCR),A
F0FE'                    WRT$EX1:
F0FE'  36 00             LD      (HL),0          ;Sterge rezultat
F100'                    WRT$EX2:
F100'  C3 EAD2'          JP      INCR$CR         ;incrementeaza CR
                         ;       *****   ******
                         ;
                         ;Aceasta rutina calculeaza (din adresa aleatoare) numarul de
                         ;si inregistrare si deschide extensia specificata
                         ;               reg.C= FF pentru citire si
                         ;                   C= 00 pentru scriere
                         ;
F103'                    COM$RAND:
F103'  AF                XOR     A               ;Memoreaza comutator
F104'  32 F3D5'          LD      (MODSWTCH),A    ;Acces direct
                         ;       *****   ******
                         ;
F107'                    RAND$COMM$ENTRY:
F107'  C5                PUSH    BC              ;Indicator operatie salvare
F108'  2A E943'          LD      HL,(FCBSAVE)    ;Incarca adresa FCB
F10B'  EB                EX      DE,HL           ;in D,E
F10C'  21 0021           LD      HL,33           ;adresa indicator indirect
F10F'  19                ADD     HL,DE           ;in H,L
F110'  7E                LD      A,(HL)          ;Incarca R0
F111'  E6 7F             AND     127             ;Masca pentru CR
F113'  F5                PUSH    AF              ;Salveaza in stiva
F114'  7E                LD      A,(HL)          ;Incarca din nou
F115'  17                RLA                     ;si CARRY=bit 7 (CMPS bit pentru
F116'  23                INC     HL              ;extensie). Incarca
F117'  7E                LD      A,(HL)          ;R1
F118'  17                RLA                     ;si deplaseaza bitul CMPS al ex-
F119'  E6 1F             AND     31              ;tensiei.Mascheaza numar extensie
F11B'  4F                LD      C,A             ;si salveaza in registrul C
F11C'  7E                LD      A,(HL)          ;Incarca din nou R1
F11D'  1F                RRA                     ;si roteste la
F11E'  1F                RRA                     ;dreapta
F11F'  1F                RRA
F120'  1F                RRA                     ;pentru a genera
F121'  E6 0F             AND     15              ;extensia S2
F123'  47                LD      B,A             ;Salveaza S2 in registrul B
F124'  F1                POP     AF              ;Restaureaza CR
F125'  23                INC     HL              ;Incarca
F126'  6E                LD      L,(HL)          ;R2 in registrul L
F127'  2C                INC     L               ;si test
F128'  2D                DEC     L               ;daca nu e zero
F129'  2E 06             LD      L,6             ;Cod eroare in L (nr.inregistrare
F12B'  C2 F18B'          JP      NZ,RAND$TO$LARGE;prea mare). Eroare iesire
F12E'  21 0020           LD      HL,32           ;Incarca
F131'  19                ADD     HL,DE           ;adresa FCB-CR
F132'  77                LD      (HL),A          ;si salveaza valoare
F133'  21 000C           LD      HL,12           ;extensie
```

```
F136'  19                      ADD     HL,DE
F137'  79                      LD      A,C             ;Salveaza extensie in registrul A
F138'  96                      SUB     (HL)            ;Test daca coincide extensia cu-
F139'  C2 F147'                JP      NZ,RAND$NO$MATCH;renta, altfel deschide o noua
F13C'  21 000E                 LD      HL,14           ;extensie. Adresa S2
F13F'  19                      ADD     HL,DE           ;in H,L
F140'  78                      LD      A,B
F141'  96                      SUB     (HL)            ;Test daca coincide
F142'  E6 7F                   AND     127             ;(inlatura bitul 7 )
F144'  CA F17F'                JP      Z,EXTEN$FORCED  ;Deschide noua extensie
F147'                  RAND$NO$MATCH:
F147'  C5                      PUSH    BC              ;Salveaza extensia si S2
F148'  D5                      PUSH    DE              ;Salveaza adresa FCB
F149'  CD EEA2'                CALL    CLOSE$ENTEN     ;Inchide extensia curenta
F14C'  D1                      POP     DE              ;Restaureaza FCB
F14D'  C1                      POP     BC              ;si EXT, S2
F14E'  2E 03                   LD      L,3             ;Cod eroare (nu se poate inchide
F150'  3A E945'                LD      A,(RESULT)      ;extensia). Test daca operatia
F153'  3C                      INC     A               ;de inchidere OK
F154'  CA F184'                JP      Z,RAND$ERR$SQ   ;Eroare iesire
F157'  21 000C                 LD      HL,12           ;Adresa extensie FCB
F15A'  19                      ADD     HL,DE           ;in H,L
F15B'  71                      LD      (HL),C          ;Salveaza noua extensie in FCB
F15C'  21 000E                 LD      HL,14           ;la fel
F15F'  19                      ADD     HL,DE           ;cu S2
F160'  70                      LD      (HL),B          ;salveaza S2
F161'  CD EE51'                CALL    STRT$OPEN       ;deschide extensie
F164'  3A E945'                LD      A,(RESULT)      ;Test daca
F167'  3C                      INC     A               ;e OK
F168'  C2 F17F'                JP      NZ,EXTEN$FORCED ;Iesire fara eroare
F16B'  C1                      POP     BC              ;Restaureaza operatie
F16C'  C5                      PUSH    BC              ;si salveaza din nou
F16D'  2E 04                   LD      L,4             ;Cod eroare (extensie negasita)
F16F'  0C                      INC     C               ;Test daca e operatie de citire
F170'  CA F184'                JP      Z,RAND$ERR$SQ   ;Daca e citire, eroare
F173'  CD EF24'                CALL    MAKE$ENTRY      ;Daca e scriere, creaza extensie
F176'  2E 05                   LD      L,5             ;Cod eroare (nu se poate deschide
F178'  3A E945'                LD      A,(RESULT)      ;extensia). Test daca s-a inche-
F17B'  3C                      INC     A               ;iat cu succes
F17C'  CA F184'                JP      Z,RAND$ERR$SQ   ;Eroare iesire
F17F'                  EXTEN$FORCED:
F17F'  C1                      POP     BC              ;Restaureaza operatie
F180'  AF                      XOR     A               ;anuleaza eroarea
F181'  C3 E901'                JP      SAVRET          ;salveaza rezultat si revenire
F184'                  RAND$ERR$SQ:
F184'  E5                      PUSH    HL              ;Salveaza cod eroare
F185'  CD EB69'                CALL    LOAD$F$S2       ;pune
F188'  36 C0                   LD      (HL),0C0H       ;FCB-S2 = C0
F18A'  E1                      POP     HL              ;restaureaza cod
F18B'                  RAND$TO$LARGE:
F18B'  C1                      POP     BC              ;restaureaza operatie
F18C'  7D                      LD      A,L             ;incarca cod eroare
F18D'  32 E945'                LD      (RESULT),A
F190'  C3 EB78'                JP      SRD$FCB$S2      ;pune S2 pe citire
                       ;       *****   ******
                       ;
                       ;Aceasta rutina citeste o inregistrare (aleator)
                       ;
F193'                  STRT$RND$READ:
F193'  0E FF                   LD      C,255           ;Citeste cod
F195'  CD F103'                CALL    COM$RAND        ;calculeaza si deschide extensie
```

```
F198'  CC EFC1'          CALL   Z,COMMON$READ  ;citeste daca nu e eroare
F19B'  C9                RET
                         ;      *****  ******
                         ;
                         ;Aceasta rutina scrie o inregistrare (aleator)
                         ;
F19C'                    STRT$RND$WRITE:
F19C'  0E 00             LD     C,0            ;Scrie cod
F19E'  CD F103'          CALL   COM$RAND       ;calculeaza extensie
F1A1'  CC F003'          CALL   Z,COMMON$WRITE ;scrie daca nu e eroare
F1A4'  C9                RET
                         ;      *****  ******
                         ;
                         ;Calculeaza numarul direct de inregistrare incepind cu
                         ;extensia curenta si CR (sau pentru dimensiune fisier)
F1A5'                    COM$REC$NUM:
F1A5'  EB                EX     DE,HL          ;Acum in D,E adresa FCB si in H,L
                                                ;RC sau CR. Deplasament
F1A6'  19                ADD    HL,DE          ;CR sau RC
F1A7'  4E                LD     C,(HL)         ;Incarca
F1A8'  06 00             LD     B,0            ;cuvint RC (CR)
F1AA'  21 000C           LD     HL,12          ;Extensie
F1AD'  19                ADD    HL,DE          ;Adresa
F1AE'  7E                LD     A,(HL)         ;in A
F1AF'  0F                RRCA                  ;Roteste bitul CMS R0
F1B0'  E6 80             AND    128            ;Inlatura
F1B2'  81                ADD    A,C            ;calculeaza R0
F1B3'  4F                LD     C,A            ;salveaza
F1B4'  3E 00             LD     A,0            ;si CMS
F1B6'  88                ADC    A,B
F1B7'  47                LD     B,A            ;salveaza
F1B8'  7E                LD     A,(HL)         ;muta extensie
F1B9'  0F                RRCA                  ;roteste extensie
F1BA'  E6 0F             AND    15             ;inlatura bitii neutilizati
F1BC'  80                ADD    A,B            ;aduna CMS
F1BD'  47                LD     B,A            ;si salveaza
F1BE'  21 000E           LD     HL,14          ;adresa S2
F1C1'  19                ADD    HL,DE
F1C2'  7E                LD     A,(HL)         ;in A
F1C3'  87                ADD    A,A            ;A = A*16
F1C4'  87                ADD    A,A            ;de exemplu, deplaseaza
F1C5'  87                ADD    A,A            ;S2 in BIT7-BIT4
F1C6'  87                ADD    A,A            ;din R1
F1C7'  F5                PUSH   AF             ;salveaza pentru test
F1C8'  80                ADD    A,B            ;calculeaza R1
F1C9'  47                LD     B,A            ;salveaza in B
F1CA'  F5                PUSH   AF             ;si stiva
F1CB'  E1                POP    HL             ;Test
F1CC'  7D                LD     A,L            ;daca e
F1CD'  E1                POP    HL             ;depasire
F1CE'  B5                OR     L
F1CF'  E6 01             AND    1              ;Depasire <-- Z=0
F1D1'  C9                RET
                         ;      *****  ******
                         ;
                         ;Aceasta rutina intoarce dimensiunea fisierului
                         ;
F1D2'                    COM$FILE$SIZE:
F1D2'  0E 0C             LD     C,12           ;Coincidenta lungime(utilizator+
F1D4'  CD ED18'          CALL   FIND$FIRST     ;nume). Gaseste intrare
F1D7'  2A E943'          LD     HL,(FCBSAVE)   ;Incarca adresa FCB
```

```
F1DA'  11 0021          LD    DE,33          ;si deplasament R0
F1DD'  19               ADD   HL,DE          ;adresa R0
F1DE'  E5               PUSH  HL             ;salveaza in stiva adresa R0
F1DF'  72               LD    (HL),D         ;initializeaza R0
F1E0'  23               INC   HL
F1E1'  72               LD    (HL),D         ;R1
F1E2'  23               INC   HL
F1E3'  72               LD    (HL),D         ;si R2
F1E4'                   SIZE$LOOP:
F1E4'  CD EBF5'         CALL  DIR$RESULT     ;Negasit ?
F1E7'  CA F20C'         JP    Z,SIZE$EXIT    ;OK
F1EA'  CD EB5E'         CALL  DIR$ENT$ADD    ;Intoarce adresa intrare DIR
F1ED'  11 000F          LD    DE,15          ;Deplasament RC
F1F0'  CD F1A5'         CALL  COM$REC$NUM    ;Calculeaza R0,R1,R2 din extensie
F1F3'  E1               POP   HL             ;si Rc. Restaureaza
F1F4'  E5               PUSH  HL             ;si salveaza adresa R0
F1F5'  5F               LD    E,A            ;interschimba R0, R1, R2
F1F6'  79               LD    A,C            ;daca noua valoare
F1F7'  96               SUB   (HL)           ;este mai
F1F8'  23               INC   HL             ;mare
F1F9'  78               LD    A,B            ;decit
F1FA'  9E               SBC   A,(HL)         ;vechea valoare R0, R1, R2
F1FB'  23               INC   HL
F1FC'  7B               LD    A,E
F1FD'  9E               SBC   A,(HL)
F1FE'  DA F206'         JP    C,SKIP$CHANGE
F201'  73               LD    (HL),E         ;Aici valorile sint interschimbate
F202'  2B               DEC   HL
F203'  70               LD    (HL),B
F204'  2B               DEC   HL
F205'  71               LD    (HL),C
F206'                   SKIP$CHANGE:
F206'  CD ED2D'         CALL  FIND$NEXT      ;Gaseste urmatoarea intrare
F209'  C3 F1E4'         JP    SIZE$LOOP      ;din nou
F20C'                   SIZE$EXIT:
F20C'  E1               POP   HL             ;goleste stiva
F20D'  C9               RET
                        ;     *****  ******
                        ;
                        ;Aceasta rutina selecteaza numarul de inregistrare
                        ;incepind cu extensia si inregistrarea curenta
                        ;
F20E'                   SELRND:
F20E'  2A E943'         LD    HL,(FCBSAVE)   ;Incarca adresa FCB
F211'  11 0020          LD    DE,32          ;Deplasament CR
F214'  CD F1A5'         CALL  COM$REC$NUM    ;calculeaza
F217'  21 0021          LD    HL,33          ;si salveaza in R0, R1, R2
F21A'  19               ADD   HL,DE          ;numarul de inregistrare
F21B'  71               LD    (HL),C         ;R0
F21C'  23               INC   HL
F21D'  70               LD    (HL),B         ;R1
F21E'  23               INC   HL
F21F'  77               LD    (HL),A         ;R2
F220'  C9               RET
                        ;     *****  ******
                        ;
                        ;Aceasta functie selecteaza discul specificat
                        ;
F221'                   STRT$SEL$DSK:
F221'  2A F3AF'         LD    HL,(LOGMARK)   ;Incarca vector conectat
F224'  3A E942'         LD    A,(DSKSPEC)    ;Incarca disc specificat
```

```
F227'  4F                      LD      C,A             ;si salveaza in C
F228'  CD EAEA'                CALL    SHRHLC          ;salveaza bit conectare in L0
F22B'  E5                      PUSH    HL              ;salveaza in stiva
F22C'  EB                      EX      DE,HL           ;salveaza in D,E
F22D'  CD E959'                CALL    LOAD$DPB        ;transfera parametrii de disc si
                                                       ;selecteaza (er.Z=1)
F230'  E1                      POP     HL              ;Restaureaza conectare
F231'  CC E947'                CALL    Z,SELERR        ;selecteaza eroare
F234'  7D                      LD      A,L             ;test daca e disc conectat
F235'  1F                      RRA                     ;bit conectare in CARRY
F236'  D8                      RET     C               ;revenire daca e deja
F237'  2A F3AF'                LD      HL,(LOGMARK)    ;altfel
F23A'  4D                      LD      C,L             ;pune bit
F23B'  44                      LD      B,H             ;conectare bit
F23C'  CD EB0B'                CALL    ACT$LOGGIN      ;discul specificat
F23F'  22 F3AF'                LD      (LOGMARK),HL    ;si salveaza noul vector conecta-
F242'  C3 ECA3'                JP      INSP$DIR$ALL    ;re.Inspecteaza DIR si calculeaza
                       ;       *****   ******          ;vectorul de control alocare
                       ;
                       ;Aceasta functie selecteaza discul specificat
                       ;
F245'                  SELDSK:
F245'  3A F3D6'                LD      A,(DRVFLG)      ;Test daca
F248'  21 E942'                LD      HL,DSKSPEC      ;noul disc
F24B'  BE                      CP      (HL)            ;coincide cu cel vechi
F24C'  C8                      RET     Z               ;Nici o actiune daca e acelasi
F24D'  77                      LD      (HL),A          ;disc, altfel pune disc nou
F24E'  C3 F221'                JP      STRT$SEL$DSK    ;si selecteaza
                       ;       *****   ******
                       ;
                       ;aceasta functie inspecteaza FCB, inlatura specificatia
                       ;de disc, selecteaza "drive" si salveaza cod utilizator
                       ;
F251'                  PREPARE$FCB:
F251'  3E FF                   LD      A,255           ;Pune indicator operatie
F253'  32 F3DE'                LD      (DSKINV),A      ;disc (OFFH pentru operatie disc
F256'  2A E943'                LD      HL,(FCBSAVE)    ;BDOS. Incarca adresa FCB
F259'  7E                      LD      A,(HL)          ;incarca specificatie disc
F25A'  E6 1F                   AND     31              ;inlatura disc utilizator
F25C'  3D                      DEC     A               ;cod disc conform BDOS
F25D'  32 F3D6'                LD      (DRVFLG),A      ;memoreaza noua locatie disc
F260'  FE 1E                   CP      30              ;test pentru specificatie
F262'  D2 F275'                JP      NC,PREP001      ;implicit (OFFH). Salt peste
F265'  3A E942'                LD      A,(DSKSPEC)     ;selectie daca-i implicit
F268'  32 F3DF'                LD      (TLOGDSK),A
F26B'  7E                      LD      A,(HL)
F26C'  32 F3E0'                LD      (TSPCDSK),A
F26F'  E6 E0                   AND     0E0H
F271'  77                      LD      (HL),A
F272'  CD F245'                CALL    SELDSK
F275'                  PREP001:
F275'  3A E941'                LD      A,(USRCODE)     ;Restaureaza cod
F278'  2A E943'                LD      HL,(FCBSAVE)    ;utilizator
F27B'  B6                      OR      (HL)
F27C'  77                      LD      (HL),A
F27D'  C9                      RET
                       ;       *****   ******
                       ;
                       ;Aceasta functie intoarce numarul de versiune
                       ;
F27E'                  GETVERS:
```

```
F27E'  3E 22                    LD      A,22H           ;Versiune 2.2
F280'  C3 E901'                 JP      SAVRET          ;Revenire
                        ;       *****   ******
                        ;
                        ;Aceasta functie initializeaza DOS si pune DMA la 80H
                        ;
F283'                   DOSINT:
F283'  21 0000                  LD      HL,0            ;Toate discurile
F286'  22 F3AD'                 LD      (ROMARK),HL     ;sint citire/scriere
F289'  22 F3AF'                 LD      (LOGMARK),HL    ;si deconectate
F28C'  AF                       XOR     A               ;disc A
F28D'  32 E942'                 LD      (DSKSPEC),A     ;specificat
F290'  21 0080                  LD      HL,128          ;pune DMA
F293'  22 F3B1'                 LD      (USRDMA),HL     ;la 80H
F296'  CD EBDA'                 CALL    ST$USR$DMA
F299'  C3 F221'                 JP      STRT$SEL$DSK    ;si selecteaza disc A
                        ;       *****   ******
                        ;
                        ;Aceasta rutina deschide un fisier
                        ;
F29C'                   OPEN:
F29C'  CD EB72'                 CALL    CLR$F$S2        ;Curata  FCB S2
F29F'  CD F251'                 CALL    PREPARE$FCB     ;inspecteaza FCB
F2A2'  C3 EE51'                 JP      STRT$OPEN       ;Secventa deschidere
                        ;       *****   ******
                        ;
                        ;Aceasta rutina inchide un fisier
                        ;
F2A5'                   CLOSE:
F2A5'  CD F251'                 CALL    PREPARE$FCB     ;Inspecteaza FCB
F2A8'  C3 EEA2'                 JP      CLOSE$ENTEN     ;Secventa inchidere
                        ;       *****   ******
                        ;
                        ;Aceasta rutina gaseste prima intrare DIR care coincide cu FCB
                        ;
F2AB'                   SRCFIRST:
F2AB'  0E 00                    LD      C,0             ;Lungime comparare nula daca spe
                                                        ;cificatia de disc="?"
F2AD'  EB                       EX      DE,HL           ;Adresa FCB in H,L
F2AE'  7E                       LD      A,(HL)          ;Test specificatie disc
F2AF'  FE 3F                    CP      '?'             ;este "?"?
F2B1'  CA F2C2'                 JP      Z,QUEST$SKIP    ;Salt peste pregatire FCB
F2B4'  CD EAA6'                 CALL    FCB$EXT$ADD     ;Incarca adresa extensie
F2B7'  7E                       LD      A,(HL)          ;Incarca si
F2B8'  FE 3F                    CP      '?'             ;test daca este "?"
F2BA'  C4 EB72'                 CALL    NZ,CLR$F$S2     ;daca nu s-a gasit prima extensie
F2BD'  CD F251'                 CALL    PREPARE$FCB     ;(EXT=0). Pregateste FCB
F2C0'  0E 0F                    LD      C,15            ;compara lungime
F2C2'                   QUEST$SKIP:
F2C2'  CD ED18'                 CALL    FIND$FIRST      ;Gasit prima intrare
F2C5'  C3 EDE9'                 JP      TRF$DIR$USR     ;si transfera zona tampon DIR in
                                                        ;DMA
                        ;       *****   ******
                        ;
                        ;Aceasta rutina gaseste urmatoarea intrare DIRECTORY
                        ;
F2C8'                   SRCNEXT:
F2C8'  2A F309'                 LD      HL,(TMPFCB)     ;Adresa FCB salvata prin gasirea
F2CB'  22 E943'                 LD      (FCBSAVE),HL    ;primei secvente. Salveaza adresa
F2CE'  CD F251'                 CALL    PREPARE$FCB     ;FCB. Pregateste FCB
F2D1'  CD ED2D'                 CALL    FIND$NEXT       ;gaseste urmatoarea
```

```
F2D4'  C3 EBE9'          JP      TRF$DIR$USR     ;Transfer in DMA utilizator
                         ;       *****   ******
                         ;
                         ;Aceasta rutina sterge toate intrarile care coincid cu FCB
                         ;
F2D7'                    ERASE:
F2D7'  CD F251'                  CALL    PREPARE$FCB     ;Pregateste FCB
F2DA'  CD ED9C'                  CALL    DEL$ENTRY       ;sterge toate intrarile care
F2DD'  C3 ED01'                  JP      RET$FF$01       ;coincid. Incarca rezultat si
                                                         ;salveaza
                         ;       *****   ******
                         ;
                         ;Aceasta rutina citeste o inregistrare
                         ;
F2E0'                    READREC:
F2E0'  CD F251'                  CALL    PREPARE$FCB     ;Pregateste FCB
F2E3'  C3 EFBC'                  JP      STRT$READ$REC   ;Start citire
                         ;       *****   ******
                         ;
                         ;Aceasta rutina scrie o inregistrare
                         ;
F2E6'                    WRITEREC:
F2E6'  CD F251'                  CALL    PREPARE$FCB     ;Pregateste FCB
F2E9'  C3 EFFE'                  JP      STRT$WRITE      ;Start operatie
                         ;       *****   ******
                         ;
                         ;Aceasta rutina creaza o noua intrare DIRECTORY
                         ;
F2EC'                    MAKEFILE:
F2EC'  CD EB72'                  CALL    CLR$F$S2        ;Curata FCB S2
F2EF'  CD F251'                  CALL    PREPARE$FCB     ;Pregateste FCB
F2F2'  C3 EF24'                  JP      MAKE$ENTRY      ;Creaza intrare DIRECTORY
                         ;       *****   ******
                         ;
                         ;Aceasta rutina redenumeste toate intrarile DIR care coincid
                         ;
F2F5'                    RENAME:
F2F5'  CD F251'                  CALL    PREPARE$FCB     ;Pregateste FCB
F2F8'  CD EE16'                  CALL    STRT$REN$SEQ    ;Secventa redenumire
F2FB'  C3 ED01'                  JP      RET$FF$01       ;Incarca rezultat si salveaza
                         ;       *****   ******
                         ;
                         ;Aceasta rutina intoarce vectorul conectare
                         ;
F2FE'                    LOGVECT:
F2FE'  2A F3AF'                  LD      HL,(LOGMARK)    ;Incarca conectare
F301'  C3 F329'                  JP      RESL$EXIT$LD    ;revenire
                         ;       *****   ******
                         ;
                         ;Aceasta rutina intoarce discul conectat
                         ;
F304'                    LOGDSK:
F304'  3A E942'                  LD      A,(DSKSPEC)     ;Incarca disc conectat
F307'  C3 E901'                  JP      SAVRET          ;Salveaza, revenire
                         ;       *****   ******
                         ;
                         ;Aceasta rutina pune adresa DMA
                         ;
F30A'                    STDMA:
F30A'  EB                        EX      DE,HL           ;DMA utilizator in H,L
F30B'  22 F3B1'                  LD      (USRDMA),HL     ;Salveaza
```

```
F30E'  C3 EBDA'          JP      ST$USR$DNA      ;si pune
                         ;       *****   ******
                         ;
                         ;Aceasta rutina intoarce adresa vector alocare
                         ;
F311'                    ALLVEC:
F311'  2A F38F'          LD      HL,(ALLOCZON)   ;Incarca adresa alocare
F314'  C3 F329'          JP      RESL$EXIT$LD    ;Intoarce rezultat
                         ;       *****   ******
                         ;
                         ;Aceasta rutina intoarce vector numai citire
                         ;
F317'                    ROVECT:
F317'  2A F3AD'          LD      HL,(ROMARK)     ;Citeste vector numai citire
F31A'  C3 F329'          JP      RESL$EXIT$LD    ;Salveaza rezultat si revenire
                         ;       *****   ******
                         ;
                         ;Aceasta functie schimba atributele unui fisier
                         ;
F31D'                    ATTRIB:
F31D'  CD F251'          CALL    PREPARE$FCB     ;Pregateste FCB
F320'  CD EE3B'          CALL    STRT$ATTRIB     ;Secventa redenumire
F323'  C3 ED01'          JP      RET$FF$01       ;Intoarce rezultat
                         ;       *****   ******
                         ;
                         ;Aceasta rutina intoarce adresa blocului cu parametrii de disc
                         ;
F326'                    GETDPB:
F326'  2A F3B8'          LD      HL,(DSKPBL)     ;Incarca adresa
F329'                    RESL$EXIT$LD:
F329'  22 E945'          LD      (RESULT),HL     ;salveaza rezultat
F32C'  C9                RET
                         ;       *****   ******
                         ;
                         ;Aceasta rutina pune/intoarce codul utilizator
                         ;
F32D'                    GPUSER:
F32D'  3A F3D6'          LD      A,(DRVFLG)      ;Incarca
F330'  FE FF             CP      255             ;operatie
F332'  C2 F33B'          JP      NZ,SKP$USR$ERR  ;Este cod pus ?
F335'  3A E941'          LD      A,(USRCODE)     ;altfel, incarca
F338'  C3 E901'          JP      SAVRET          ;si intoarce cod utilizator
F33B'                    SKP$USR$ERR:
F33B'  E6 1F             AND     31              ;mascheaza bitii in plus
F33D'  32 E941'          LD      (USRCODE),A     ;memoreaza nou cod utilizator
F340'  C9                RET
                         ;       *****   ******
                         ;
                         ;Aceasta rutina citeste o inregistrare (mod direct)
                         ;
F341'                    RANREAD:
F341'  CD F251'          CALL    PREPARE$FCB
F344'  C3 F193'          JP      STRT$RND$READ
                         ;       *****   ******
                         ;
                         ;Aceasta rutina scrie o inregistrare (mod direct)
                         ;
F347'                    RAN$RITE:
F347'  CD F251'          CALL    PREPARE$FCB
F34A'  C3 F19C'          JP      STRT$RND$WRITE
                         ;       *****   ******
```

```
                        ;
                        ;Aceasta rutina intoarce dimensiunea fisierului
                        ;
F34D'                   GETSIZE:
F34D'  CD F251'                 CALL    PREPARE$FCB
F350'  C3 F1D2'                 JP      COM$FILE$SIZE
                        ;       *****   ******
                        ;
                        ;Aceasta rutina deconecteaza un disc
                        ;
F353'                   LOGOFF:
F353'  2A E943'                 LD      HL,(FCBSAVE)    ;Incarca masca deconectare
F356'  7D                       LD      A,L
F357'  2F                       CPL
F358'  5F                       LD      E,A
F359'  7C                       LD      A,H
F35A'  2F                       CPL                     ;si complementeaza
F35B'  2A F3AF'                 LD      HL,(LOGMARK)    ;incarca vector conectare
F35E'  A4                       AND     H               ;mascheaza
F35F'  57                       LD      D,A             ;disc specificat
F360'  7D                       LD      A,L
F361'  A3                       AND     E
F362'  5F                       LD      E,A
F363'  2A F3AD'                 LD      HL,(ROMARK)     ;Acelasi vector numai citire
F366'  EB                       EX      DE,HL
F367'  22 F3AF'                 LD      (LOGMARK),HL    ;Salveaza noul vector conectare
F36A'  7D                       LD      A,L
F36B'  A3                       AND     E
F36C'  6F                       LD      L,A
F36D'  7C                       LD      A,H
F36E'  A2                       AND     D
F36F'  67                       LD      H,A
F370'  22 F3AD'                 LD      (ROMARK),HL     ;Salveaza noul vector numai
F373'  C9                       RET                     ;citire
                        ;       *****   ******
                        ;
                        ;Aceasta este secventa de iesire pentru cele mai
                        ;multe din functiile BDOS
                        ;
F374'                   EXISEQ:
F374'  3A F3DE'                 LD      A,(DSKINV)      ;Test daca functia BDOS a impli-
F377'  B7                       OR      A               ;cat o operatie cu fisier
F378'  CA F391'                 JP      Z,RESUNF        ;salt daca nu a fost operatie cu
F37B'  2A E943'                 LD      HL,(FCBSAVE)    ;discul. Incarca adresa FCB si
F37E'  36 00                    LD      (HL),0          ;pune o specificatie de disc la 0
F380'  3A F3E0'                 LD      A,(TSPCDSK)     ;incarca discul specificat si
F383'  B7                       OR      A               ;test daca e implicit
F384'  CA F391'                 JP      Z,RESUNF        ;salt daca discul e implicit
F387'  77                       LD      (HL),A          ;altfel restaureaza disc
F388'  3A F3DF'                 LD      A,(TLOGDSK)     ;incarca vechiul disc conectat
F38B'  32 F3D6'                 LD      (DRVFLG),A
F38E'  CD F245'                 CALL    SELDSK          ;si reselecteaza
F391'                   RESUNF:
F391'  2A E90F'                 LD      HL,(SPSAVE)     ;Restaureaza indicator stiva
F394'  F9                       LD      SP,HL           ;utilizator
F395'  2A E945'                 LD      HL,(RESULT)     ;incarca rezultat
F398'  7D                       LD      A,L             ;in H,L
F399'  44                       LD      B,H             ;si in B,A
F39A'  C9                       RET                     ;revenire
                        ;       *****   ******
                        ;
```

```
                        ;Aceasta rutina scrie o inregistrare directa pe disc anterior
                        ;operatiei de scriere inregistrare. Toti arborii vor fi stersi
                        ;
F39B'                   RNDERA:
F39B'  CD F251'                 CALL    PREPARE$FCB     ;Pregateste FCB
F39E'  3E 02                    LD      A,2             ;scrie/sterge indicator arbore
F3A0'  32 F3D5'                 LD      (MODSWTCH),A    ;memorat
F3A3'  0E 00                    LD      C,0             ;scrie indicator
F3A5'  CD F107'                 CALL    RAND$COMM$ENTRY ;calculeaza adresa
F3A8'  CC F003'                 CALL    Z,COMMON$WRITE  ;scrie
F3AB'  C9                       RET
                        ;
F3AC'  E5               DELENT: DB      0E5H    ;Sterge semn intrare
F3AD'  0000             ROMARK: DW      0       ;Vector numai citire
F3AF'  0000             LOGMARK:DW      0       ;Vector conectare
F3B1'  0080             USRDMA: DW      128     ;Adresa DMA utilizator (80H implicit)
F3B3'  0000             SCRH0:  DW      0       ;Adresa intrare verificata (BIOS)
F3B5'  0000             SCRH1:  DW      0     ;Adresa margine (inregistrari )pista (BIOS)
F3B7'  0000             SCRH2:  DW      0       ;Adresa pista BIOS
F3B9'  0000             DIRBUF: DW      0       ;Adresa zona tampon DIR (BIOS)
F3BB'  0000             DSKPBL: DW      0       ;Adresa bloc parametri de disc (BIOS)
F3BD'  0000             CHCKZON:DW      0       ;si adresa vector control (BIOS)
F3BF'                   ALLOCZON:               ;Adresa vector alocare (BIOS)
F3BF'  0000                     DW      0
F3C1'  0000             SECTRK: DW      0       ;Sector pe pista (BIOS)
F3C3'                   BLKSHIFT:         ;Deplasare bloc (numar inregistrari pe arbore)
F3C3'  00                       DB      0
F3C4'  00               BLKMASK:DB      0       ;Masca bloc
F3C5'  00               EXTMASK:DB      0       ;Masca extensie
F3C6'  0000             DSKSIZE:DW      0       ;Dimensiune disc-1
F3C8'  0000             DIRMAX: DW      0       ;Intrari DIRECTORY
F3CA'  0000             ALLDIR: DW      0       ;Alocare DIR
F3CC'  0000             CKSIZE: DW      0       ;Dimensiune vector control
F3CE'  0000             TRKOFF: DW      0       ;Compensare pista
F3D0'  0000             TRANADD:DW      0       ;Adresa translatare (BIOS)
F3D2'  00               EXCLFLG:DB      0       ;Indicator extensie inchisa
F3D3'  00               RDWRFLG:DB      0       ;Indicator scriere/citire(OFFH/0)
F3D4'  00               FNDFLG: DB      0       ;Indicator gasit (OFFH negasit)
F3D5'                   MODSWTCH:               ;Comutator mod acces (0-direct,
F3D5'  00                       DB      0       ;1-secvential, 2-scrie+sterge
F3D6'  00               DRVFLG: DB      0       ;disc specificat
F3D7'  00               CLSTNB: DB      0       ;Numar arbore (pozitie FCB)
F3D8'  00               TMPLEN: DB      0       ;Lungime temporara de coincidenta
F3D9'  0000 0000        TMPFCB: DW      0,0     ;Salveaza FCB
F3DD'  00               SIZEFLG:DB      0       ;Indicator dimensiune disc
F3DE'  00               DSKINV: DB      0       ;Indicator disc implicat (OFFH-operatie
F3DF'  00               TLOGDSK:DB      0       ;cu discul). Salveaza disc conectat
F3E0'  00               TSPCDSK:DB      0       ;Salveaza specificatie disc
F3E1'  00               TMPRC:  DB      0       ;Salveaza RC
F3E2'  00               MEXT:   DB      0       ;Salveaza extensie mascata
F3E3'  0000             TMPCR:  DW      0       ;Salveaza CR
F3E5'  0000             SECTNB: DW      0       ;Salveaza numar inregistrare
F3E7'  0000             MPYCLST:DW      0       ;Salveaza margine arbore
F3E9'  00               DIRDISP:DB      0       ;Deplasament DIR
F3EA'                   DIRCOUNT:               ;CMPS octet din numar
F3EA'  00                       DB      0
F3EB'  00               HIGHDIR:DB      0       ;DIRECTORY si CMS
F3EC'  0000             CKCOUNT:DW      0       ;Numar control
F3EE'                           DS      18

                                ORG     BDOS - 806H + 1600H
```

```
F400'                   BBOOT   EQU     $
F403'                   WBOOT   EQU     $+3
F406'                   BCNSST  EQU     $+6
F409'                   BCONIN  EQU     $+9
F40C'                   BCONOUT EQU     $+0CH
F40F'                   LISTOUT EQU     $+0FH
F412'                   PUNCH   EQU     $+12H
F415'                   BRDER   EQU     $+15H
F418'                   BHOME   EQU     $+18H
F41B'                   BSELDSK EQU     $+1BH
F41E'                   BSETTRK EQU     $+1EH
F421'                   BSETSEC EQU     $+21H
F424'                   BSETDMA EQU     $+24H
F427'                   BREAD   EQU     $+27H
F42A'                   BWRITE  EQU     $+2AH
F42D'                   LISTST  EQU     $+2DH
F430'                   BSTRAN  EQU     $+30H

                                END
```

Macros:

Symbols:

EB0B'	ACT$LOGGIN	EB8C'	ACT$VERY	EB64'	ADD$DISP
E24B'	ADDCH	E059'	ADDHL	E647'	ADDTAB
EC75'	ALL$MORE$CLS	ECD2'	ALL$NEXT$ENTRY	EC6B'	ALLRESCLSTS
F3CA'	ALLDIR	EC35'	ALLOC$ENTRY	F3BF'	ALLOCZON
F311'	ALLVEC	F31D'	ATTRIB	E9E4'	BACKWARD
E606'	BBDOS	F400'	BBOOT	F406'	BCNSST
F409'	BCONIN	F40C'	BCONOUT	E5F1'	BCOUNT
DEF4'	BDFLAG	0005	BDOS	DEC3'	BDRES
F418'	BHOME	E8C1'	BICRS	E88A'	BIDLO
E899'	BIERA	E8BD'	BINOO	E837'	BINABO
E85F'	BINCLR	E86B'	BINCU	E848'	BINHRD
E816'	BINOBS	E8A6'	BINORM	E826'	BINRUB
E7F1'	BIREDO	E8A9'	BISVV	0008	BKSP
DEA2'	BLANCK	F3C4'	BLKMASK	F3C3'	BLKSHIFT
F415'	BRDER	F427'	BREAD	F41B'	BSELDSK
F424'	BSETDMA	F421'	BSETSEC	F41E'	BSETTRK
E7AC'	BSPRN	E7A4'	BSSPBS	E611'	BSTART
F430'	BSTRAN	DE07'	BUFLEN	DE06'	BUFMAX
DE88'	BUFPOINT	E7D3'	BUFPRN	F42A'	BWRITE
DFBA'	CAPSEND	DF30'	CAPSTR	DEDA'	CCLOSE
DE8C'	CCONOUT	DE00'	CCPBASE	E128'	CCPVER
DE98'	CCRLF	E56B'	CERR01	F3BD'	CHCKZON
E90E'	CHRBUF	DE08'	CINPBUF	E8E0'	CITRAN
EB9E'	CK$ENTRY	EAFD'	CK$LOOP	F3EC'	CKCOUNT
F3CC'	CKSIZE	E5AB'	CLOCSTACK	E5EF'	CLOGDSK
F2A5'	CLOSE	EEA2'	CLOSE$ENTEN	EF17'	CLOSE$FIVE
EEFD'	CLOSE$FOUR	EED8'	CLOSE$ONE	EF1F'	CLOSESIXERR
EEE9'	CLOSE$THREE	EEE1'	CLOSE$TWO	ECB1'	CLRALLZONE
EF46'	CLRENTCLSTR	EB72'	CLRFS2	E84E'	CLRLOOP
F09A'	CLRTRECMSK	F3D7'	CLSTNB	EC8E'	CLSTR$CONT
EFFB'	CLSTR$INACTV	EECD'	CLSTR$LOOP$TR	EDD1'	CLSTRNOTZE
EC88'	CLSTRTWOBYT	EA5C'	CLTR$END	EA53'	CLTR$EXT
EA45'	CLTR$LOOP	EA90'	CLUST$MPY	E8FE'	CNSTAT
DFC2'	CNSTCI	EAF7'	COM$CHECK	EB9C'	COMCKVECT
F1D2'	COM$FILE$SIZE	F103'	COM$RAND	F1A5'	COMRECNUM
EA77'	COMSAVCLUST	E1C1'	COMADRS	E5ED'	COMCR
E586'	COMEXIT	E583'	COMEXT	E5CD'	COMFCB
EFC1'	COMMON$READ	F003'	COMMON$WRITE	DED0'	COMOPEN
EA3E'	COMP$CLSTR$NB	EA8A'	COMP$RECORD$NB	DEFE'	COMREAD

DEE9'	COMSRCH	E110'	COMTAB	E8C8'	CONIN
E790'	CONOUT	DECB'	COPEN	DE92'	COSVB
000D	CR	EAAE'	CRRCADD	DF09'	CREATE
DF0E'	CRENAME	E7C9'	CRLF	E77F'	CRTLCH
DEBD'	CSELDSK	E15C'	CSTART	E208'	DEC1
E233'	DEC2	E1F8'	DECBIN	ED9C'	DEL$ENTRY
EDA4'	DELNXTENT	E90A'	DELCOSW	F3AC'	DELENT
E030'	DELIM	EB5E'	DIRENTADD	E8F5'	DIR$RESULT
E288'	DIR1	E30F'	DIR10	E31B'	DIR11
E28F'	DIR2	E298'	DIR3	E2CC'	DIR4
E2D4'	DIR5	E2D9'	DIR6	E2F7'	DIR7
E2F9'	DIR8	E30E'	DIR9	F3B9'	DIRBUF
E277'	DIRCOM	E8D4'	DIRCON	F3EA'	DIRCOUNT
F3E9'	DIRDISP	E5EE'	DIRENTRY	F3C8'	DIRMAX
E9C3'	DIRTRKSC	E6E5'	DISPERR	E9B2'	DKREAD
E98B'	DKRESULT	E9B8'	DKWRITE	EA0F'	DONETRK
DE88'	DOSINIT	F283'	DOSINT	E600'	DOSVER
F3D6'	DRVFLG	F3DE'	DSKINV	F38B'	DSKPBL
F3C6'	DSKSIZE	E942'	DSKSPEC	E5F2'	DUMMY
ED94'	ENTNOTFND	ED83'	ENTRY$FOUND	E699'	EPRC1
E6A5'	EPRC2	E6AB'	EPRC3	E6B1'	EPRC4
EE8B'	EQUAL$EXT	E342'	ERA1	E31F'	ERACOM
DEEF'	ERAFILE	E352'	ERAMES	F2D7'	ERASE
DFDD'	ERASUB	E609'	ERR01	E60B'	ERR02
E60D'	ERR03	E60F'	ERR04	E94A'	ERRBRU
E009'	ERROR	F3D2'	EXCLFLG	F374'	EXISEQ
E589'	EXIT1	EF83'	EXT$CARRY	EFB6'	EXTERRSEQ
EFAF'	EXT$EXIT$OK	ED07'	EXT$MATCH	EFAC'	EXTMSKOVR
ED73'	EXT$MTCH$TST	EF8E'	EXT$SEARCH	E5D6'	EXTEN
F17F'	EXTEN$FORCED	F3C5'	EXTMASK	E1CF'	FATAL
EAA6'	FCBEXTADD	E5BA'	FCBS2	E943'	FCBSAVE
EB47'	FILROFLG	F08C'	FILL$BUFFER	E5CE'	FILNAME
ED7C'	FINDCHRNXT	ED18'	FIND$FIRST	EDBE'	FIND$FREE
ED2D'	FIND$NEXT	E098'	FN1	E0DF'	FN10
E0E9'	FN11	E0F0'	FN12	E0F2'	FN13
E101'	FN14	E109'	FN15	E0A9'	FN2
E0AB'	FN3	E0AF'	FN4	E0B9'	FN5
E0C0'	FN6	E0C8'	FN7	E0D9'	FN8
E0DB'	FN9	F3D4'	FNDFLG	E096'	FNSTART
E9FA'	FOREWARD	E7EF'	FRSTPOZ	E198'	GETCOM
F326'	GETDPB	E8ED'	GETIOB	F34D'	GETSIZE
DF13'	GETUSER	F27E'	GETVERS	ECF6'	GOTO$NEXT$ENT
F32D'	GPUSER	F3EB'	HIGHDIR	EB05'	HL$LOOP
E90D'	HRDCOPY	E90B'	IBLCNT	EAD2'	INCR$CR
EADE'	INCR00	E7E1'	INPBUF	ECA3'	INSPDIRALL
E706'	LCICQ	E742'	LCST1	E745'	LCST2
E723'	LCSTCI	E501'	LDEND	E571'	LDERR
E57A'	LDMESS	E54F'	LEAVE	000A	LF
E90C'	LINCNT	DF39'	LINEINP	F40F'	LISTOUT
F42D'	LISTST	E00F'	LNERR	EA5E'	LOAD$CLUSTER
E959'	LOAD$DPB	EB69'	LOADFS2	E6FB'	LOCCI
E748'	LOCCON	E941'	LOCSTACK	F304'	LOGDSK
DFD0'	LOGGIN	F3AF'	LOGMARK	F353'	LOGOFF
F2FE'	LOGVECT	DFAB'	LOOPCAPS	EDEC'	MAKE$BUSY
EF24'	MAKE$ENTRY	F2EC'	MAKEFILE	E1F0'	MESNOF
E6BA'	MESS0	E6C6'	MESS1	E6CA'	MESS2
E6D5'	MESS3	E6DC'	MESS4	F6E1'	MESS5
DEAC'	MESSAGE	F3E2'	MEXT	F3D5'	MODSWTCH
EC19'	MOREDIREN	E240'	MOV3DH	E242'	MOVDHB
E530'	MOVLINE	EB04'	MPYHLC	F3E7'	MPYCLST
E53E'	MVL0	E543'	MVL1	E94F'	MVSRDH

E950′	MVSRLOOP	EFE6′	NEW$EXTENSION	EE40′	NEXT$ATTRIB
EC9D′	NEXT$CLSTR	EDC0′	NEXT$FREE	EE27′	NEXT$REN
DEA7′	NLMESS	F0DF′	NOCRMOD	E1EA′	NOFERR
E089′	NOSPEC	DF96′	NOSUB	EDF4′	NOTFRECL
E779′	NOTAB	E762′	NOTPR	E904′	NULLSUB
ED53′	NXTCHRENT	EF5A′	NXTEXTSEQ	E905′	ONERET
F29C′	OPEN	E133′	PICK1	E13C′	PICK2
E14F′	PICK3	E154′	PICK4	E12E′	PICKCOM
EC20′	PMY32	DE8A′	POINSKP	EE5A′	PREP$ENTRY
F275′	PREP001	F251′	PREPARE$FCB	E6B4′	PRNERRABO
E8F8′	PRNBUF	E870′	PRNDEL	E878′	PRNLOOP
EB2C′	PROTECT	F412′	PUNCH	DF15′	PUTUSER
E022′	QMARK	F2C2′	QUEST$SKIP	F107′	RAND$COMM$ENTRY
F184′	RANDERRSQ	F147′	RANDNOMATCH	F188′	RANDTOLARGE
F341′	RANREAD	F347′	RANWRITE	EBD4′	RDDIRREC
E1D9′	RDERR	E1DF′	RDMESS	F3D3′	RDWRFLG
E8CE′	READIMP	F2E0′	READREC	E158′	REDOST
E43F′	REN1	E459′	REN2	E46D′	REN3
E473′	REN4	E479′	REN5	E482′	REN6
F2F5′	RENAME	E410′	RENCOM	EBFE′	RESET$DIR
F329′	RESL$EXIT$LD	DE96′	RESTB	E945′	RESULT
E182′	RESUME	F391′	RESUNF	ED01′	RETFF01
E7B1′	RLPOZ	F39B′	RNDERA	F3AD′	ROMARK
F317′	ROVECT	E3D4′	SAVE1	E3F1′	SAVE2
E3FB′	SAVE3	E401′	SAVE4	E407′	SAVE5
E3AD′	SAVECOM	E901′	SAVRET	DF1D′	SAVUSER
F3B3′	SCRH0	F3B5′	SCRH1	F3B7′	SCRH2
E9A1′	SCRHOME	DFD5′	SDMAD	E99D′	SDSFLG
E5DD′	SECFN	F3E5′	SECTNB	F3C1′	SECTRK
E254′	SELD1	E266′	SELD2	F245′	SELDSK
E947′	SELERR	F20E′	SELRND	DEF9′	SEQREAD
EC5C′	SETBITALLOC	DFD8′	SETDMA	E8F3′	SETIOB
E7B9′	SETPOZ	EC56′	SHL$ALLOC	EC64′	SHR$ALLOC
EAEA′	SHRHLC	EAEB′	SHRLOOP	F20C′	SIZE$EXIT
F1E4′	SIZE$LOOP	F3DD′	SIZEFLG	F206′	SKIP$CHANGE
F0D2′	SKPACTRC	F33B′	SKPUSRERR	E04F′	SKPBLK
0020	SPACE	E090′	SPCDRV	E5F0′	SPECDRIVE
E90F′	SPSAVE	F2AB′	SRCFIRST	DEDF′	SRCHFIRST
DEE4′	SRCHNEXT	F2C8′	SRCNEXT	EB78′	SRDFCBS2
EBE3′	STCOMDMA	EBE0′	STDIRDMA	EBDA′	STUSRDMA
F30A′	STDMA	DF29′	STLOGG	EBC4′	STORE$CK
EE3B′	STRT$ATTRIB	EE51′	STRT$OPEN	EFBC′	STRT$READ$REC
EE16′	STRTRENSEQ	F193′	STRTRNDREAD	F19C′	STRTRNDWRITE
F221′	STRTSELDSK	EFFE′	STRT$WRITE	DFA7′	STTRAN
E9D1′	STTRKSEC	EB95′	SUB$16	E5CC′	SUBCR
E5AC′	SUBFCB	E5BB′	SUBRC	E5AB′	SUBSWITCH
EABB′	SVCRRC$EX	0009	TAB	E796′	TABEXP
EA84′	TEST$CLUST	ED4A′	THIS$ERA	F3DF′	TLOGDSK
F3E3′	TMPCR	F3D9′	TMPFCB	F3D8′	TMPLEN
F3E1′	TMPRC	E05E′	TRABUF	F3D0′	TRANADD
E4A5′	TRANCOM	E060′	TRB01	EBE9′	TRFDIRUSR
F3CE′	TRKOFF	E4C4′	TRN1	E4E1′	TRN2
F3E0′	TSPCDSK	EB54′	TSTDSKRO	EB44′	TSTFILRO
EB1E′	TSTROVEC	EB7F′	TST$VERY	E714′	TSTPRN
EE94′	TWO$BYTE$CLSTR	EA71′	TWO$CLUSTER	E374′	TYPE1
E387′	TYPE2	E3A0′	TYPE3	E3A7′	TYPE4
E35D′	TYPECOM	E48E′	USERCOM	E941′	USRCODE
F3B1′	USRDMA	DF1A′	USSAV	DFFD′	VERLOOP
EC05′	VERYDIRREC	DFF5′	VRSVER	F403′	WBOOT
EE01′	WRSECDIR	EE10′	WRSEQTRKSC	DF04′	WRITE
F2E6′	WRITEREC	F03B′	WRT$001	F048′	WRT$002

```
F06C'   WRT$003          F06E'   WRT$ACT$CLS      F064'   WRT$CLSTR$2
EBC6'   WRT$DIR$REC      F0FE'   WRT$EX1           F100'   WRT$EX2
EDFD'   WRT$FR$FCB       F0BB'   WRT$NORM

No Fatal error(s)
```

```
                        ;          7.3 Componenta BIOS
                        ;
                        ;-Interfata cu consola in standard VT52:
                        ; concept by Barbulescu Corneliu
                        ;-Interfata grafica standard Tektronix si drivere imprimante,
                        ; interfata lector si perforator:
                        ; concept by Horatiu Moldovan

                        ;  ================
                        ;  ! 7.3.1 MainBios !
                        ;  ================

                        .PHASE  0F400H
F400  C3 F590           START:  JP      BOOT
F403  C3 F596           WBOOTE: JP      WBOOT
F406  C3 F5A8                   JP      CONST
F409  C3 F59C                   JP      CONIN
F40C  C3 F5A2                   JP      CONOUT
F40F  C3 F5C0                   JP      LIST
F412  C3 F5AE                   JP      PUNCH
F415  C3 F5B4                   JP      READER
F418  C3 F528                   JP      HOME
F41B  C3 F5CF                   JP      SELDSK
F41E  C3 F52B                   JP      SETTRK
F421  C3 F530                   JP      SETSEC
F424  C3 F523                   JP      SETDMA
F427  C3 F5D4                   JP      READ
F42A  C3 F5D9                   JP      WRITE
F42D  C3 F545                   JP      LISTST
F430  C3 F5DE                   JP      SECTR

F433  FB70              ZZKSYO: DW      ZKST0
F435  A3F0              ZZOBIOS:DW      ZOBIOS
F437  F5F8              ZZINTR: DW      ZINTR
                        ;emite un parametru catre 8272
                        ;Intrari: A-codul parametrului
                        ;
F439  F3                L400:   DI
F43A  01 2FFD                   LD      BC,2FFDH  ;in BC adresa MSR, registru de stare,
F43D  F5                        PUSH    AF        ;(main status register)
F43E  F5                        PUSH    AF
F43F  CD F45B                   CALL    MOON      ;ar fi bine sa fie motorul pornit!
F442  ED 78             JR001:  IN      A,(C)     ;preiau stare 8272 din MSR
F444  87                        ADD     A,A
F445  30 FB                     JR      NC,JR001  ;RQM=1? (8272 e pe faza?) Nu,mai astept
F447  87                        ADD     A,A       ;Este
F448  30 03                     JR      NC,JR002  ;DIO=0? (ne potrivim la sens?)
F44A  F1                        POP     AF        ;Nu, avem aceleasi intentii, si el vrea
F44B  F1                        POP     AF        ;sa-mi trimita un octet! Abandon
F44C  C9                        RET
F44D  F1                JR002:  POP     AF        ;aduc codul parametrului din stiva
F44E  06 3F                     LD      B,03FH    ;adresa DR (registru de date) in BC
F450  ED 79                     OUT     (C),A     ;emit parametrul
F452  06 2F                     LD      B,02FH    ;refac MSR in BC
F454  3E 07                     LD      A,007H    ;o mica temporizare...
F456  3D                JR003:  DEC     A
F457  20 FD                     JR      NZ,JR003
F459  F1                        POP     AF
F45A  C9                        RET               ;si-am scapat!
                        ;Verifica daca motorul este pornit
                        ;Daca nu-i, il porneste si asteapta pina se stabilizeaza turatia
```

```
                        ;
F45B  FD 21 FB70  MOON:  LD    IY,ZKST0
F45F  FD CB 10 7E        BIT   7,(IY+10H)      ;motorul este pornit?
F463  20 22              JR    NZ,L401         ;Da, continui comanda
F465  FD CB 10 F6        SET   6,(IY+10H)      ;Nu, activez fanion start motor
F469  FD 7E 14           LD    A,(IY+14H)      ;preiau imaginea portului #1FFD
F46C  F6 08              OR    008H            ;fortez bitul D3 pe "1" (start)
F46E  FD 77 14           LD    (IY+14H),A      ;actualizez imagine port #1FFD
F471  C5                 PUSH  BC
F472  01 1FFD            LD    BC,1FFDH
F475  ED 79              OUT   (C),A           ;start motor
F477  FD 7E 11           LD    A,(IY+11H);preiau nr.rotatii pentru stabilizare
F47A  FD 77 13           LD    (IY+13H),A;le duc in contorul CMAN
F47D  FB                 EI              ;astept trecerea timpului necesar
F47E  FD CB 10 7E  L402: BIT   7,(IY+10H);stabilizarii = (CMAN)*20 milisecunde
F482  28 FA              JR    Z,L402
F484  C1                 POP   BC
F485  F3                 DI              ;motor operational
F486  C9                 RET
F487  FD 7E 12     L401: LD    A,(IY+12H);reactualizez contorul pe 9 biti pentru
F48A  FD 77 13           LD    (IY+13H),A;o noua comanda disc
F48D  FD CB 10 EE        SET   5,(IY+10H);setez bitul noua, cel mai semnificativ
F491  C9                 RET
                        ;aceiasi functie cu rutina L400, cu deosebirea ca pastreaza
                        ;valoarea lui BC
                        ;
F492  C5           LL400: PUSH  BC
F493  CD F439            CALL  L400
F496  C1                 POP   BC
F497  C9                 RET

                        ;emite un bloc de parametri catre 8272, modificind in prealabil
                        ;pe primul si pe cel de-al treilea conform fetei fizice (capului)
                        ;la care se face acces
                        ;Intrari: HL-adresa blocului de parametri
                        ;         A -codul comenzii
                        ;         C -numarul de parametri
                        ;
F498  E5           AD13: PUSH  HL
F499  2A FB51            LD    HL,(DSKNO)      ;memoreaza vechiul numar disc
F49C  22 F756            LD    (DDSKNO),HL     ;la DDSKNO
F49F  F5                 PUSH  AF
F4A0  CB 95              RES   2,L
F4A2  AF                 XOR   A
F4A3  CB 3C              SRL   H
F4A5  17                 RLA
F4A6  32 FB53            LD    (HEAD),A    ;pozitionez bit0 din HEAD (capul)
F4A9  17                 RLA
F4AA  17                 RLA
F4AB  B5                 OR    L
F4AC  6F                 LD    L,A
F4AD  22 FB51            LD    (DSKNO),HL  ;pozitionez bit2 din DSKNO (nr.disc)
F4B0  F1                 POP   AF
F4B1  E1                 POP   HL
                        ;emisie bloc de parametri
                        ;
F4B2  F3           D13:  DI
F4B3  CD F492            CALL  LL400       ;emit codul comenzii
F4B6  79                 LD    A,C
F4B7  21 FB51            LD    HL,RWTBL
F4BA  F5           L41C: PUSH  AF          ;emit, succesiv, numarul de parametri
```

```
F4BB  7E           LD    A,(HL)      ;specificat in intrare prin registrul
F4BC  CD F439      CALL  L400        ;C
F4BF  23           INC   HL
F4C0  F1           POP   AF
F4C1  3D           DEC   A
F4C2  20 F6        JR    NZ,L41C
F4C4  C9           RET

                   ;reface vechiul numar de disc
                   ;
F4C5  E5           BD13: PUSH  HL
F4C6  2A F756            LD    HL,(DDSKNO)
F4C9  22 FB51            LD    (DSKNO),HL
F4CC  E1                 POP   HL
F4CD  C9                 RET

                   ;rutina preluare rezultat de la 8272
                   ;Intrari: BC-adresa MSR
                   ;
F4CE  D5           L427: PUSH  DE
F4CF  21 FB45            LD    HL,DSTS     ;adresa tabelei pentru preluare
F4D2  16 00              LD    D,000H      ;parametri rezultat
F4D4  E5                 PUSH  HL
F4D5  ED 78        D10:  IN    A,(C)
F4D7  FE C0              CP    0C0H
F4D9  38 FA              JR    C,D10
F4DB  06 3F              LD    B,03FH
F4DD  ED 78              IN    A,(C)
F4DF  06 2F              LD    B,02FH
F4E1  23                 INC   HL
F4E2  77                 LD    (HL),A
F4E3  14                 INC   D
F4E4  3E 05              LD    A,005H
F4E6  3D           JR004: DEC  A
F4E7  20 FD              JR    NZ,JR004
F4E9  ED 78              IN    A,(C)
F4EB  E6 40              AND   040H
F4ED  20 E6              JR    NZ,D10
F4EF  E1                 POP   HL
F4F0  72                 LD    (HL),D
F4F1  23                 INC   HL
F4F2  7E                 LD    A,(HL)
F4F3  E6 C0              AND   0C0H
F4F5  D1                 POP   DE
F4F6  C9                 RET

                   ;rutina pentru scriere sector
                   ;Intrari: HL-adresa sectorului in memorie
                   ;         BC-adresa MSR
                   ;
F4F7  D5           MINIWR: PUSH  DE          ;salvam reg.DE
F4F8  1E 20               LD    E,20H       ;masca pentru fanionul EXM al
F4FA  18 06               JR    MINIWR1     ;circuitului 8272, mod nonDMA
F4FC  06 3F        L48B:  LD    B,03FH      ;adresa DR
F4FE  ED A3               OUTI              ;scrie octetul (HL)-->DR(8272)
F500  06 2F               LD    B,02FH      ;adresa MSR
F502  ED 78        MINIWR1:IN   A,(C)       ;preia stare MSR
F504  F2 F502             JP    P,MINIWR1   ;Daca fanionul RQM="1", mai astept
F507  A3                  AND   E           ;8272 mai vrea octeti?
F508  C2 F4FC             JP    NZ,L48B     ;Da, ii mai dam!
F50B  D1                  POP   DE          ;Nu, am scris sectorul. Refacem
```

```
F50C  C9                  RET               ;reg.DE si iesim.

                          ;rutina pentru citire sector
                          ;Intrari: HL-adresa din memorie a sectorului
                          ;         BC-adresa MSR
                          ;
F50D  D5          MINIRD: PUSH  DE          ;salvam reg.DE
F50E  1E 20               LD    E,20H       ;masca EXM
F510  18 06               JR    MINIRD1
F512  06 3F       L49C:   LD    B,03FH      ;adresa DR
F514  ED A2               INI               ;citeste octetul DR(8272)-->(HL)
F516  06 2F               LD    B,02FH      ;adresa MSR
F518  ED 78       MINIRD1:IN    A,(C)       ;preia stare MSR
F51A  F2 F518             JP    P,MINIRD1   ;daca RQM="1", salt
F51D  A3                  AND   E           ;mai avem de citit?
F51E  C2 F512             JP    NZ,L49C     ;Da, un nou octet
F521  D1                  POP   DE          ;refacem reg.DE
F522  C9                  RET               ;O.K.

                          ;defineste noua adresa DMA, memorata in BC
                          ;
F523  ED 43 FB3A  SETDMA: LD    (DMAADD),BC
F527  C9                  RET

                          ;recalibrare pentru drive-ul curent
                          ;
F528  01 0000     HOME:   LD    BC,0
                          ;defineste noua pista, memorata in BC
                          ;
F52B  ED 43 FB36  SETTRK: LD    (SEKTRK),BC
F52F  C9                  RET

                          ;defineste noul sector, memorat in C
                          ;
F530  79          SETSEC: LD    A,C
F531  32 FB35             LD    (LACSEC),A
F534  C9                  RET
                          ;
F535  CD F643     RWP:    CALL  COMUT2
F538  CD F498             CALL  AD13
F53B  2A FB3A             LD    HL,(DMAADD)
F53E  C9                  RET
F53F  CD F68A     RWRET:  CALL  COMUT0
F542  C3 8222             JP    RWRET1

                          ;set pentru eroare permanenta
                          ;
F545  AF          LISTST: XOR   A
F546  3D                  DEC   A
F547  C9                  RET
                          ;
F548  01 0CFD     KAPOOR: LD    BC,0CFDH
F54B  32 FB6D             LD    (AMAN),A
F54E  ED 78               IN    A,(C)
F550  32 FB6C             LD    (RDCFD),A
F553  ED 59               OUT   (C),E
F555  3A FB6D             LD    A,(AMAN)
F558  06 7F               LD    B,7FH
F55A  ED 79               OUT   (C),A
F55C  D9                  EXX
F55D  06 00               LD    B,0
```

```
F55F  08                      EX      AF,AF'
F560  28 07                   JR      Z,RDS2
F562  30 04                   JR      NC,RDS5
F564  ED 44                   NEG
F566  CB BF                   RES     7,A
F568  3F              RDS5:   CCF
F569  4F              RDS2:   LD      C,A
F56A  08                      EX      AF,AF'
F56B  ED B0                   LDIR
F56D  01 7FFD                 LD      BC,07FFDH
F570  3A FB87                 LD      A,(P7FFD)
F573  ED 79                   OUT     (C),A
F575  06 0C                   LD      B,0CH
F577  3A FB6C                 LD      A,(RDCFD)
F57A  ED 79                   OUT     (C),A
F57C  C3 85EF                 JP      KAP
                      ;
F57F  CD F643         MOWGLI: CALL    COMUT2
F582  01 803F                 LD      BC,0803FH
F585  20 05                   JR      NZ,WERDSK
F587  ED B2           RERDSK: INIR
F589                  BAGHEERA:
F589  AF                      XOR     A
F58A  18 66                   JR      SHERKAN
F58C  ED B3           WERDSK: OTIR
F58E  18 F9                   JR      BAGHEERA
                      ;
F590  21 00A9         BOOT:   LD      HL,VIOBYT
F593  22 0003                 LD      (IOBYTE),HL
F596  E5              WBOOT:  PUSH    HL
F597  21 A457                 LD      HL,WBOOT1
F59A  18 2E                   JR      COMUT
                      ;
F59C  E5              CONIN:  PUSH    HL
F59D  21 851E                 LD      HL,CONIN1
F5A0  18 28                   JR      COMUT
                      ;
F5A2  E5              CONOUT: PUSH    HL
F5A3  21 8523                 LD      HL,CONOUT1
F5A6  18 22                   JR      COMUT
                      ;
F5A8  E5              CONST:  PUSH    HL
F5A9  21 8531                 LD      HL,CONST1
F5AC  18 1C                   JR      COMUT
                      ;
F5AE  E5              PUNCH:  PUSH    HL
F5AF  21 84FC                 LD      HL,PUNCH1
F5B2  18 16                   JR      COMUT
                      ;
F5B4  E5              READER: PUSH    HL
F5B5  21 8507                 LD      HL,READER1
F5B8  18 10                   JR      COMUT
F5BA  E5              RDRST:  PUSH    HL
F5BB  21 850F                 LD      HL,RDRST1
F5BE  18 0A                   JR      COMUT
                      ;
F5C0  E5              LIST:   PUSH    HL
F5C1  21 8514                 LD      HL,LIST1
F5C4  18 04                   JR      COMUT
                      ;
F5C6  E5              ERASE:  PUSH    HL
```

```
F5C7  21 9459          LD      HL,ERASE1
F5CA  CD F5E1          COMUT:  CALL    COMUT3
F5CD  E1               POP     HL
F5CE  C9               RET
                       ;
F5CF  21 836C          SELDSK: LD      HL,SELDSK1
F5D2  18 0D            JR      COMUT3
                       ;
F5D4  21 8099          READ:   LD      HL,READ1
F5D7  18 08            JR      COMUT3
                       ;
F5D9  21 8095          WRITE:  LD      HL,WRITE1
F5DC  18 03            JR      COMUT3
                       ;
F5DE  21 8327          SECTR:  LD      HL,SECTR1
F5E1  F3               COMUT3: DI
F5E2  ED 73 FF00               LD      (0FF00H),SP
F5E6  31 FF00                  LD      SP,0FF00H
F5E9  CD F68A                  CALL    COMUT0
F5EC  CD 8000                  CALL    COMUT1
F5EF  CD F643                  CALL    COMUT2
F5F2  ED 7B FF00       SHERKAN:LD      SP,(0FF00H)
F5F6  FB                       EI
F5F7  C9                       RET
                       ;
F5F8  F3               ZINTR:  DI
F5F9  E5                       PUSH    HL
F5FA  21 8F76                  LD      HL,ZINTR1
F5FD  ED 73 FE00               LD      (0FE00H),SP
F601  31 FE00                  LD      SP,0FE00H
F604  F5                       PUSH    AF
F605  C5                       PUSH    BC
F606  3A FB87                  LD      A,(P7FFD)
F609  32 FB88                  LD      (IP7FFD),A
F60C  3E 00                    LD      A,0
F60E  01 7FFD                  LD      BC,7FFDH
F611  32 FB87                  LD      (P7FFD),A
F614  ED 79                    OUT     (C),A
F616  06 0C                    LD      B,0CH
F618  ED 78                    IN      A,(C)
F61A  32 FB89                  LD      (IOCFD),A
F61D  E6 F9                    AND     0F9H
F61F  ED 79                    OUT     (C),A
F621  C1                       POP     BC
F622  F1                       POP     AF
F623  CD 8000                  CALL    COMUT1
F626  F5                       PUSH    AF
F627  C5                       PUSH    BC
F628  01 0CFD                  LD      BC,0CFDH
F62B  3A FB89                  LD      A,(IOCFD)
F62E  ED 79                    OUT     (C),A
F630  3A FB88                  LD      A,(IP7FFD)
F633  06 7F                    LD      B,7FH
F635  32 FB87                  LD      (P7FFD),A
F638  ED 79                    OUT     (C),A
F63A  C1                       POP     BC
F63B  F1                       POP     AF
F63C  ED 7B FE00               LD      SP,(0FE00H)
F640  E1                       POP     HL
F641  FB                       EI
F642  C9                       RET
```

```
F643  F5          ;
                  COMUT2: PUSH    AF
F644  C5                  PUSH    BC
F645  01 0CFD             LD      BC,0CFDH
F648  3A FB8B             LD      A,(C0CFD)
F64B  ED 79               OUT     (C),A
F64D  3A FB8A     COMUT2C:LD      A,(C7FFD)
F650  06 7F               LD      B,7FH
F652  32 FB87             LD      (P7FFD),A
F655  ED 79       COMUT2A:OUT     (C),A
F657  C1                  POP     BC
F658  F1                  POP     AF
F659  C9                  RET
                  ;
F65A  F5          SCOMUT2:PUSH    AF
F65B  C5                  PUSH    BC
F65C  01 0CFD             LD      BC,0CFDH
F65F  ED 78               IN      A,(C)
F661  32 FB6E             LD      (XC0CFD),A
F664  3A FB8B             LD      A,(C0CFD)
F667  ED 79               OUT     (C),A
F669  3A FB87             LD      A,(P7FFD)
F66C  32 FB6F             LD      (XC7FFD),A
F66F  18 DC               JR      COMUT2C
                  ;
F671  CD F65A     PIR:    CALL    SCOMUT2
F674  ED B0               LDIR
                  ;
F676  F5          SCOMUT0:PUSH    AF
F677  C5                  PUSH    BC
F678  01 7FFD             LD      BC,7FFDH
F67B  3A FB6F             LD      A,(XC7FFD)
F67E  32 FB87             LD      (P7FFD),A
F681  ED 79               OUT     (C),A
F683  06 0C               LD      B,0CH
F685  3A FB6E             LD      A,(XC0CFD)
F688  18 CB               JR      COMUT2A
                  ;
F68A  F5          COMUT0: PUSH    AF
F68B  C5                  PUSH    BC
F68C  01 7FFD             LD      BC,7FFDH
F68F  3A FB87             LD      A,(P7FFD)
F692  32 FB8A             LD      (C7FFD),A
F695  3E 00               LD      A,0
F697  32 FB87             LD      (P7FFD),A
F69A  ED 79               OUT     (C),A
F69C  06 0C               LD      B,0CH
F69E  ED 78               IN      A,(C)
F6A0  32 FB8B             LD      (C0CFD),A
F6A3  E6 F9               AND     0F9H
F6A5  18 AE               JR      COMUT2A
                  ;
F6A7  CD F643     BALOO:  CALL    COMUT2
F6AA  FB                  EI
F6AB  C3 DE00     SFR1:   JP      CCP

                  ;parametri pentru dubla densitate, 256 octeti/sector:
                  ;
F6AE  01          D2VAL:  DB      1       ;cod=(nr. sectoare logice/sector fizic)-1
F6AF  10                  DB      16      ;numar maxim de sectoare logice
F6B0  0E                  DB      0EH     ;lungime gap
```

```
F6B1  FF                    DB    255     ;lungimea cimpului de date; irelevant
F6B2  F6BA                  DW    DPBD2   ;pointer pentru blocul parametrilor
                                          ;discului

                      ;parametri pentru dubla densitate, 512 octeti/sector:
                      ;
F6B4  02              D5VAL: DB   2       ;cod=(nr. sectoare logice/sector fizic
F6B5  09                    DB    9       ;numar maxim de sectoare logice
F6B6  0E                    DB    0EH     ;lungime gap
F6B7  FF                    DB    0FFH    ;lungimea cimpului de date; irelevant
F6B8  F6C9                  DW    DPBD5   ;pointer pentru blocul parametrilor
                                          ;discului

                      ;blocul parametrilor discului (DPB) pentru 256 octeti/sector:
                      ;
F6BA  0020            DPBD2: DW   32      ;sectoare logice/pista
F6BC  03 07                 DB    3,7     ;shift si masca pentru bloc logic de 1
F6BE  00                    DB    0       ;extensie masca
F6BF  0097                  DW    151     ;numar maxim de blocuri pe disc
F6C1  003F                  DW    63      ;(numar maxim de directori)-1
F6C3  C0 00                 DB    0C0H,0  ;vector de alocare
F6C5  0010                  DW    16      ;dimensiune vector verificare director
F6C7  0002                  DW    2       ;numar de piste rezervate

                      ;blocul parametrilor discului pentru 512 octeti/sector:
                      ;
F6C9  0024            DPBD5: DW   36      ;sectoare logice/pista
F6CB  04 0F                 DB    4,15    ;shift si masca pentru bloc logic de 
F6CD  00                    DB    0       ;extensie masca
F6CE  015D                  DW    349     ;numar maxim de blocuri pe disc
F6D0  007F                  DW    127     ;(numar maxim de directori)-1
F6D2  C0 00                 DB    0C0H,0  ;vector de alocare
F6D4  0020                  DW    32      ;dimensiune vector verificare direct
F6D6  0004                  DW    4       ;numar de piste rezervate

                      ;tabela de definire DPH pentru discul A (DPH="tabela parametr
                      ;
F6D8  F73A 0000       DPBASE: DW  XLT1,0,0,0,DIRBUF,0,CSV0,ALV0
F6DC  0000 0000
F6E0  F958 0000
F6E4  FA05 F9D8
F6E8  FF                    DB    0FFH    ;tip unitate si tip inregistrare (TUI
                      ;TUI=#FF: unitatea si inregistrarea (densitatea) necunoscute
                      ;TUI=#01: unitate 5.25" cu DD, 256b/sf
                      ;TUI=#02: unitate 5.25" cu DD, 512b/sf
                      ;TUI=#10: unitate virtuala (RAMDISC), 128K
                      ;TUI=#20: unitate virtuala (ERAMDISC), 0.5M sau 1M
                      ;DD="dubla densitate", iar xb/sf="x octeti/sector fizic"

                      ;tabela de definire disc DPH pentru drive-ul B
                      ;
F6E9  F73A 0000             DW    XLT1,0,0,0,DIRBUF,0,CSV1,ALV1
F6ED  0000 0000
F6F1  F958 0000
F6F5  FA53 FA26
F6F9  FF                    DB    0FFH

                      ;tabela de definire SYSTEM RAMDISC
                      ;
F6FA  0000 0000             DW    0,0,0,0,DIRBUF,DPBRDK,CSV2,ALV2
F6FE  0000 0000
```

```
F702  F958 F71C
F706  FA85 FA74
F70A  10                DB      10H

                        ;tabela de definire EXTENSION RAMDISC
                        ;
F70B  0000 0000         DW      0,0,0,0,DIRBUF,EDPBRDK,CSV3,ALV3
F70F  0000 0000
F713  F958 F72B
F717  FAD7 FA96
F71B  20                DB      20H
                        ;blocul parametrilor pentru SYSTEM RAMDISC, 512 octeti/sector:
                        ;
F71C  0100              DPBRDK: DW      256     ;sectoare logice/pista
F71E  03 07             DB      3,7     ;shift si masca pentru bloc logic de 1K
F720  00                DB      0       ;extensie masca
F721  006F              DW      127-DRAMD       ;numar maxim de blocuri pe disc
F723  003F              DW      63      ;(numar maxim de directori)-1
F725  C0 00             DB      0C0H,0  ;vector de alocare
F727  0000              DW      0       ;dimensiune vector verificare director
F729  0000              DW      0       ;numar de piste rezervate

                        ;blocul parametrilor pentru EXTENSION RAMDISC, 512 octeti/sector:
                        ;
F72B  0100              EDPBRDK:DW      256     ;sectoare logice/pista
F72D  04 0F             DB      4,15    ;shift si masca pentru bloc logic de 2K
F72F  00                DB      0       ;extensie masca
F730  01FF              DW      511     ;numar maxim de blocuri pe disc
F732  00FF              DW      255     ;(numar maxim de directori)-1
F734  F0 00             DB      0F0H,0  ;vector de alocare
F736  0000              DW      0       ;dimensiune vector verificare director
F738  0000              DW      0       ;numar de piste rezervate

                        ;tabela de translatare a sectoarelor pentru dubla densitate,
                        ;512 octeti/sector
                        ;
F73A  00 02 04 06       XLT1:   DB      0,2,4,6,8,1,3,5,7
F73E  08 01 03 05
F742  07
F743                    DS      17

0010                    DRAMD   EQU     16 ;defineste, in kiloocteti, spatiul din memoria
                                           ;RAMDISC rezervat componentei ShadowBios
05CB                    RO5CB   EQU     05CBH
0004                    USRDSK  EQU     4
0010                    RETRY   EQU     16 ;numarul maxim de incercari la scriere/citire
0004                    NRDSKS  EQU     4       ;numarul de discuri operationale
0003                    IOBYTE  EQU     3       ;adresa variabilei IOBYTE
00A9                    VIOBYT  EQU     0A9H    ;valoarea initiala a variabilei IOBYTE
0080                    BUFF    EQU     80H     ;bufferul de manevra sector logic
0100                    STACK   EQU     BUFF+80H;adresa initiala a pointerului stiva
0040                    CKSIZE  EQU     64      ;dimensiunea spatiului total de memorie
DC00                    BASE    EQU     (CKSIZE-9)*1024 ;dimensiunea spatiului utilizabil
DE00                    CCP     EQU     BASE+200H       ;debutul componentei CCP
E606                    BDOS    EQU     CCP+806H        ;debutul componentei BDOS
F400                    BIOS    EQU     CCP+1600H       ;debutul componentei BIOS
DD00                    ETPA    EQU     CCP-256
F754  DD00              TPALOC: DW      ETPA
F756                    DDSKNO: DS      2       ;variabila locala pentru drive-ul curent
F758                    RWBUF:  DS      512 ;buffer pentru citire/scriere sector fizic
F958                    DIRBUF: DS      128 ;buffer pentru citire/scriere sector logic
```

```
F9D8                    ALV0:   DS      45
FA05                    CSV0:   DS      33
FA26                    ALV1:   DS      45
FA53                    CSV1:   DS      33
FA74                    ALV2:   DS      17
FA85                    CSV2:   DS      17
FA96                    ALV3:   DS      65
FAD7                    CSV3:   DS      85
FB2C                    RTCNT:  DS      1       ;contor rezervat pentru numar de incercari
FB2D                    RWFLG:  DS      1       ;tipul operatiei: citire (01), scriere (00)
FB2E                    BUFWRT: DS      1       ;starea sectorului fizic: 00; citibil
                                                                ; 01; inregistrabil
FB2F                    SEKDSK: DS      1       ;memoreaza codul noului drive
FB30                    CURDPH: DS      2       ;memoreaza DPH-ul curent
FB32                    SEKDEN: DS      1       ;densitatea discului curent (TUI)
FB33                    DPBPNT: DS      2       ;memoreaza pointerul pentru DPB-ul curent
FB35                    LACSEC: DS      1       ;memoreaza codul noului sector
FB36                    SEKTRK: DS      2       ;memoreaza codul noii piste
FB38                    CURDEN: DS      1       ;densitatea curenta
FB39                    SEKSEC: DS      1       ;tine minte sectorul netranslatat
FB3A                    DMAADD: DS      2       ;adresa DMA de prelucrare a sectorului logic
FB3C                    DSKOP:  DS      1       ;tipul operatiei: citire (00), scriere (01)
FB3D                    WRTYPE: DS      1       ;tipul operatiei de scriere (00 sau 02)
FB3E                    UNACNT: DS      1
FB3F                    UNADSK: DS      1
FB40                    UNATRK: DS      2
FB42                    UNASEC: DS      1
FB43                    RSFLG:  DS      1       ;fanion citire sf: 00; sf deja citit
                                                                ; 01; sf trebuie citit
FB44                    RECFL:  DS      1       ;fanion RECALIBRARE, utilizat in cazul primei
                                                ;erori  la scriere/citire
FB45                    DSTS:   DS      1       ;contor numar de parametri rezultat
FB46                    RWSTBL: DS      7       ;zona parametrilor rezultat (ST0, ST1,...)
FB4D                    TRKTBL: DS      4       ;tine evidenta ultimei piste utilizate pentru
                                                ;fiecare disc din patru posibile
FB51                    RWTBL:                  ;aici se aranjeaza parametri inainte de a fi
                                                ;transmisi, in faza de comanda 8272
FB51                    DSKNO:  DS      1       ;numarul discului
FB52                    TRKNO:  DS      1       ;numarul pistei
FB53                    HEAD:   DS      1       ;numarul capului descriere/citire
FB54                    SECT:   DS      1       ;numarul sectorului
FB55                    N:      DS      1       ;tipul sectorului: 0; 256 octeti/sector
                                                                ; 1; 512 octeti/sector
FB56                    EOT:    DS      1       ;numarul ultimului sector
FB57                    GPL:    DS      1       ;lungime gap
FB58                    DTL:    DS      1       ;lungime date
FB59                    ATDLOC: DS      17
FB6A    0000            TRK:    DW      0
FB6C                    RDCFD:  DS      1

FB6D                    AMAN:   DS      1
FB6E                    XC0CFD: DS      1
FB6F                    XC7FFD: DS      1
                        ;
FB70                    ZKST0:
FB70    FF              KST0:   DB      0FFH;KST0-7 variabile pentru scanarea tastaturii
FB71    00                      DB      0
FB72    23                      DB      23H
FB73    0D                      DB      0DH
FB74    0D              KST4:   DB      0DH
FB75    05                      DB      5
```

```
FB76  23                    DB    23H
FB77  00                    DB    00H
FB78                LASTK:  DS    1   ;variabila pentru ultima tasta apasata
FB79  23            REPDEL: DB    23H ;contor pentru validare tasta noua
FB7A  05            REPPER: DB    5   ;contor pentru tasta in mod "repeat"
FB7B  08            FLAGS:  DB    8
FB7C  08            FLAGS2: DB    8
FB7D  00            MODE:   DB    0   ;modul de tratare al tastaturii
FB7E  01            PIP:    DB    1   ;durata sunetului produs la apasare tasta
FB7F  01            FRAME:  DB    1
FB80  A0            MOFL:   DB    0A0H;control stare motor discuri:
                                      ;   bit7: 0-motor neoperational
                                      ;         1-motor operational
                                      ;   bit6: 0-motor operational sau oprit
                                      ;         1-motor in curs de stabilizare
                                      ;   bit5: cel mai semnificativ bit (9) din
                                      ;         contorul de mentinere motor pornit
FB81  38            CDT:    DB    38H ;nr.rotatii stabilizare, dupa pornire motor
FB82  FF            CMON:   DB    0FFH;impreuna cu bit5 formeaza un contor pe 9
                                      ;biti, care reprezinta numarul de rotatii
                                      ;dupa care se opreste motorul, daca intre
                                      ;timp nu s-a realizat un acces la disc
FB83  FF            CMAN:   DB    0FFH;contor de manevra pentru numar de rotatii
FB84  0E            P1FFD:  DB    0EH ;imagine port #7FFD
FB85  19            PORTFE: DB    19H ;imagine port #**FE (conteaza numai  #FE)
FB86  00            BORDER: DB    0   ;culoare border
FB87  00            P7FFD:  DB    0   ;imagine port #7FFD
FB88                IP7FFD: DS    1   ;imagine port #7FFD inainte de tratare nINT
FB89                IOCFD:  DS    1   ;imagine port #0CFD inainte de tratare nINT
FB8A  31            C7FFD:  DB    31H ;imagine port #7FFD inainte de comutare MB-SB
FB8B  0F            COCFD:  DB    0FH ;imagine port #0CFD inainte de comutare MB-SB
                    ;unde: nINT=intrerupere mascabila
                    ;      comutare MB-SB=comutare MainBios-ShadowBios
FB8C  E0            PAPINK: DB    0E0H;variabila culoare PAPER si INK:
                                      ;   bit0: B (albastru) pentru INK
                                      ;   bit1: R (rosu) pentru INK
                                      ;   bit2: G (verde) pentru INK
                                      ;   bit3: B pentru PAPER
                                      ;   bit4: R pentru PAPER
                                      ;   bit5: G pentru PAPER
                                      ;   bit7: 0-culori actualizate
                                      ;         1-culori in curs de actualizare
FB8D  00            CPAPIN: DB    0
0018                CARR    EQU   24  ;numarul de rinduri alfanumerice afisate
FB8E  00            Y:      DB    0   ;numarul rindului curent (cu cursor)
FB8F  00            X:      DB    0   ;pozitia cursorului pe rind (numar coloana)
FB90  00            ROLL:   DB    0   ;numarul primului rind afisat pentru scroll
FB91  00            ESC:    DB    0   ;optiune ESCAPE
FB92  00            POZCUR: DB    0   ;
FB93  00            VIDEO:  DB    0   ;mod de afisare informatie: 0 - video normal
                                      ;                             1 - video invers
FB94                ZCAR:   DS    8   ;zona de manevra pentru datele cursorului

                    ;zona rezervata tastelor functionale
                    ;fiecarei taste functionale i se asociaza o tabela de 16 octeti
                    ;sfirsitul sirului de caractere asociat tastei functionale (daca

                    ;este definit) este marcat prin codul #00
                    ;sint posibile definitii de sir de maxim 15 caractere
                    ;Initial, tastelor functionale li se asociaza sirurile de mai jos
                    ;TTF0...TTF3 sint asociate respectiv tastelor F0...F3 fara SHIFT
```

```
                                ;TTF4...TTF7 sint asociate respectiv tastelor F0...F3 cu SHIFT
                                ;
FB9C  53 54 41 54       TTF0:   DB      "STAT A:*.*",0DH,0
FBA0  20 41 3A 2A
FBA4  2E 2A 0D 00
FBA8                            DS      4
FBAC  44 49 52 20       TTF1:   DB      "DIR A:",0DH,0
FBB0  41 3A 0D 00
FBB4                            DS      8
FBBC  45 52 41 20       TTF2:   DB      "ERA A:",0
FBC0  41 3A 00
FBC3                            DS      9
FBCC  52 45 4E 20       TTF3:   DB      "REN A:",0
FBD0  41 3A 00
FBD3                            DS      9
FBDC  53 54 41 54       TTF4:   DB      "STAT B:*.*",0DH,0
FBE0  20 42 3A 2A
FBE4  2E 2A 0D 00
FBE8                            DS      4
FBEC  44 49 52 20       TTF5:   DB      "DIR B:",0DH,0
FBF0  42 3A 0D 00
FBF4                            DS      8
FBFC  45 52 41 20       TTF6:   DB      "ERA B:",0
FC00  42 3A 00
FC03                            DS      9
FC0C  52 45 4E 20       TTF7:   DB      "REN B:",0
FC10  42 3A 00
FC13                            DS      9
FC1C  FB9C              ADRCOD: DW      TTF0    ;adresa tabelei tastelor functionale
FC1E                    NRCOD:  DS      1
FC1F  00                SIRTAS: DB      0
FC20  00                TP:     DB      0
FC21  42 48 59          TA:     DB      'B','H','Y'
FC24  36 35 54                  DB      '6','5','T'
FC27  47 56 4E                  DB      'G','V','N'
FC2A  4A 55 37                  DB      'J','U','7'
FC2D  34 52 46                  DB      '4','R','F'
FC30  43 4D 4B                  DB      'C','M','K'
FC33  49 38 33                  DB      'I','8','3'
FC36  45 44 58                  DB      'E','D','X'
FC39  80 4C 4F                  DB      80H,'L','O'
FC3C  39 32 57                  DB      '9','2','W'
FC3F  53 5A 20                  DB      'S','Z',' '
FC42  0D 50 30                  DB      0DH,'P','0'
FC45  31 51 41                  DB      '1','Q','A'
FC48  00 00 00                  DB      0,0,0
FC4B  F3 F1 08                  DB      0F3H,0F1H,8
FC4E  0A 7F 1B                  DB      0AH,7FH,1BH
FC51  00 00 00                  DB      0,0,0
FC54  F2 F0                     DB      0F2H,0F0H
FC56  81 09 00                  DB      81H,9,0
FC59  7E                TE:     DB      7EH
FC5A  2A 3F 5C                  DB      '*','?',5CH
FC5D  00 7B 7D                  DB      0,7BH,7DH
FC60  5E 00 2D                  DB      '^',0,'-'
FC63  2B 3D 2E                  DB      '+','=','.'
FC66  2C 3B 22                  DB      ',',';','"'
FC69  00 3C 7C                  DB      0,'<',7CH
FC6C  3E 5D 2F                  DB      '>',5DH,'/'
FC6F  00 60 5B                  DB      0,60H,5BH
FC72  3A                        DB      3AH
```

```
                        .DEPHASE

                        ;  ==================
                        ;  ! 7.3.2 ShadowBios !
                        ;  ==================

                        .PHASE  8000H

                        ;insemnatatea aceastei rutine nu trebuie judecata in nici un caz
                        ;dupa aparente, deoarece dimensiunile ei reduse nu o impiedica sa
                        ;joace rolul cheie pentru apelul - din MainBios - a rutinelor din
                        ;ShadowBios
                        ;Intrari: HL - adresa rutinei apelate
                        ;
8000    E9              COMUT1: JP      (HL)

                        ;tipareste mesaje
                        ;Intrari: HL-adresa primului caracter tiparibil
                        ;Executie: emite catre dispozivul asignat drept consola (CONOUT)
                        ;caractere pe care le preia din memorie, incepind cu cel de la
                        ;adresa data de continutul lui HL, pina la primul caracter cu
                        ;codul #24
                        ;
8001    E3              TYPE:   EX      (SP),HL         ;salveaza HL in stiva
8002    7E                      LD      A,(HL)          ;preia codul caracterului
8003    23                      INC     HL              ;pregateste adresa urmatoare
8004    FE 24                   CP      024H            ;este cod de sfirsit de sir
8006    E3                      EX      (SP),HL         ;reface HL
8007    C8                      RET     Z               ;revenire pentru cod de sfirsit
8008    CD 800D                 CALL    D4              ;emisie caracter
800B    18 F4                   JR      TYPE            ;mai sint?...

                        ;emite un caracter la dispozitivul (CONOUT), pastrind continutul
                        ;registrelor BC
                        ;Intrari: A-codul ASCII al caracterului
                        ;
800D    C5              D4:     PUSH    BC
800E    4F                      LD      C,A
800F    CD 8836                 CALL    DISPO
8012    C1                      POP     BC
8013    C9                      RET

                        ;verifica daca discul curent este operational
                        ;
8014    3A FB51         DRNRDY: LD      A,(DSKNO)
8017    C5                      PUSH    BC
8018    4F                      LD      C,A
8019    CD 8052                 CALL    DSKST
801C    E6 20                   AND     020H
801E    C1                      POP     BC
801F    C0                      RET     NZ
8020    3A FB51                 LD      A,(DSKNO)
8023    C6 41                   ADD     A,041H
8025    32 802D                 LD      (DVNAME),A
8028    CD 8001                 CALL    TYPE
802B    0D.0A                   DB      0DH,0AH
802D    41 24           DVNAME: DB      "A$"
802F    CD 803B                 CALL    NRDYTX
8032    20 E0                   JR      NZ,DRNRDY
8034    AF                      XOR     A
```

```
8035  32 0004          LD      (USRDSK),A
8038  C3 A457          JP      WBOOT1
                       ;
803B  CD 8001          NRDYTX: CALL    TYPE
803E  3A 20 6E 6F              DB      ": not ready"
8042  74 20 72 65
8046  61 64 79
8049  16 0D 24                DB      16H,0DH,24H
804C  CD 86C3                  CALL    KBINP
804F  FE 03                    CP      "C"-40H
8051  C9                       RET

                       ;citeste starea drive-ului curent
                       ;Intrari: C - numarul drive-ului
                       ;Iesiri : A - parametrul rezultat ST3
                       ;
8052  F3               DSKST:  DI
8053  3E 04                    LD      A,004H  ;codul comenzii citire stare drive
8055  CD F492                  CALL    LL400
8058  79                       LD      A,C     ;numarul drive-ului
8059  CD F439                  CALL    L400
805C  CD F4CE                  CALL    L427    ;preluare rezultat de la 8272
805F  7E                       LD      A,(HL)  ;ST3 in A
8060  C9                       RET
                       ;
8061  3A FB2D         PTRDWR: LD      A,(RWFLG)
8064  B7                       OR      A
8065  28 0D                    JR      Z,RDERR
8067  CD 8001                  CALL    TYPE
806A  0D 0A                    DB      0DH,0AH
806C  57 52 49 54              DB      "WRITE $"
8070  45 20 24
8073  C9                       RET
                       ;
8074  CD 8001          RDERR:  CALL    TYPE
8077  0D 0A                    DB      0DH,0AH
8079  52 45 41 44              DB      "READ $"
807D  20 24
807F  C9                       RET
                       ;
8080  CD 8001          PNPROT: CALL    TYPE
8083  70 72 6F 74              DB      "protect$"
8087  65 63 74 24
808B  C9                       RET
                       ;
808C  C5               L7B6:   PUSH    BC
808D  CD 842C                  CALL    RDID
8090  3A FB49                  LD      A,(RWSTBL+3)
8093  C1                       POP     BC
8094  C9                       RET

                       ;scrie sectorul curent
                       ;
8095  3E 01           WRITE1: LD      A,001H  ;optiune scriere sector
8097  18 02                    JR      D0

                       ;citeste sectorul curent
                       ;
8099  AF               READ1:  XOR     A       ;optiune citire sector
809A  4F                       LD      C,A     ;forteaza tip de scriere alocata
809B  32 FB3C          D0:     LD      (DSKOP),A       ;memoreaza operatia curenta
```

```
809E  79                    LD    A,C
809F  32 FB3D               LD    (WRTYPE),A      ;memoreaza tipul scrierii
80A2  3A FB32               LD    A,(SEKDEN)      ;tipul densitatii
80A5  FE 10                 CP    010H            ;este RAMDISC intern?
80A7  CA 857A               JP    Z,RDSKRW        ;da, salt
80AA  FE 20                 CP    020H            ;este RAMDISC extern?
80AC  CA 855C               JP    Z,ERDSKRW       ;da, salt
80AF  3A FB3C               LD    A,(DSKOP)       ;care-i tipul operatiei?
80B2  B7                    OR    A               ;citire?
80B3  28 74                 JR    Z,SPREAD
80B5  3A FB3D               LD    A,(WRTYPE)      ;Scriere; ce tip de scriere?
80B8  FE 02                 CP    002H            ;este un bloc nealocat?
80BA  20 1D                 JR    NZ,CHKUNA       ;daca nu, cauta alta stare
80BC  CD 811D               CALL  GETDPB  ;selecteaza blocul curent al parametrilor
80BF  23                    INC   HL
80C0  23                    INC   HL
80C1  23                    INC   HL      ;pozitionare pe masca blocului
80C2  7E                    LD    A,(HL)  ;o preia
80C3  3C                    INC   A       ;si-o incrementeaza
80C4  32 FB3E               LD    (UNACNT),A ;o retine drept contor inregistrare
80C7  3A FB2F               LD    A,(SEKDSK)
80CA  32 FB3F               LD    (UNADSK),A      ;actualizeaza numarul discului
80CD  2A FB36               LD    HL,(SEKTRK)
80D0  22 FB40               LD    (UNATRK),HL     ;actualizeaza numarul pistei
80D3  3A FB39               LD    A,(SEKSEC)
80D6  32 FB42               LD    (UNASEC),A      ;actualizeaza numarul sectorului
80D9  3A FB3E         CHKUNA: LD    A,(UNACNT)      ;aduce contorul de inregistrari
80DC  B7                    OR    A
80DD  28 4A                 JR    Z,SPREAD
80DF  3D                    DEC   A               ;il decrementeaza
80E0  32 FB3E               LD    (UNACNT),A
80E3  3A FB2F               LD    A,(SEKDSK)
80E6  21 FB3F               LD    HL,UNADSK
80E9  BE                    CP    (HL)            ;acelasi?
80EA  20 3D                 JR    NZ,SPREAD
80EC  2A FB36               LD    HL,(SEKTRK)     ;acelasi disc, caut aceiasi pista
80EF  EB                    EX    DE,HL
80F0  2A FB40               LD    HL,(UNATRK)
80F3  7A                    LD    A,D
80F4  BC                    CP    H
80F5  20 32                 JR    NZ,SPREAD
80F7  7B                    LD    A,E
80F8  BD                    CP    L
80F9  20 2E                 JR    NZ,SPREAD
80FB  3A FB39               LD    A,(SEKSEC)      ;acelasi disc, aceiasi pista
80FE  21 FB42               LD    HL,UNASEC
8101  BE                    CP    (HL)            ;este si sectorul acelasi?
8102  20 25                 JR    NZ,SPREAD
8104  34                    INC   (HL)    ;da, il inpachetam, dupa care incrementam
8105  7E                    LD    A,(HL)  ;numarul sectorului; il preluam
8106  CD 811D               CALL  GETDPB  ;pozitionare pe blocul parametrilor disc
8109  BE                    CP    (HL)    ;codul sectorului depaseste valoarea
810A  38 0B                 JR    C,NOOVF ;maxima admisibila? Nu, sare la NOOVF
810C  AF                    XOR   A       ;DA
810D  32 FB42               LD    (UNASEC),A      ;sector nou, primul pe noua pista
8110  2A FB40               LD    HL,(UNATRK)
8113  23                    INC   HL              ;incrementeaza numar pista
8114  22 FB40               LD    (UNATRK),HL
8117  AF                NOOVF: XOR  A
8118  32 FB43               LD    (RSFLG),A       ;marcheaza operatie precitire
811B  18 14                 JR    RWBUF0
```

```
                        ;returneaza in DE adresa DPB-ului curent
                        ;
811D  2A FB30           GETDPB: LD      HL,(CURDPH)
8120  11 000A                   LD      DE,000AH
8123  19                       ADD     HL,DE
8124  5E                       LD      E,(HL)
8125  23                       INC     HL
8126  56                       LD      D,(HL)
8127  EB                       EX      DE,HL
8128  C9                       RET
                        ;
8129  AF                SPREAD: XOR     A
812A  32 FB3E                  LD      (UNACNT),A      ;anuleaza contorul de nealocare
812D  3C                       INC     A
812E  32 FB43                  LD      (RSFLG),A       ;si forteaza optiunea precitire
8131  3A FB32           RWBUF0: LD      A,(SEKDEN)
8134  1F                       RRA
8135  3A FB35                  LD      A,(LACSEC)
8138  38 02                    JR      C,JR007
813A  B7                       OR      A
813B  1F                       RRA
813C  B7                 JR007: OR      A
813D  1F                       RRA
813E  06 00                    LD      B,000H
8140  3C                 RWBUF1: INC     A
8141  4F                       LD      C,A
8142  3A FB2F                  LD      A,(SEKDSK)      ;buffer activ
8145  6F                       LD      L,A
8146  3A FB51                  LD      A,(DSKNO)
8149  BD                       CP      L               ;cauta pentru discul curent
814A  20 0F                    JR      NZ,NEWSEC       ;daca nu-i, sector nou
814C  21 FB36                  LD      HL,SEKTRK
814F  3A FB52                  LD      A,(TRKNO)
8152  BE                       CP      (HL)            ;cauta pentru pista curenta
8153  20 06                    JR      NZ,NEWSEC       ;nu-i, sector nou
8155  3A FB54                  LD      A,(SECT)
8158  B9                       CP      C               ;cauta pentru sectorul curent
8159  28 34                    JR      Z,CRSEC         ;da, buffer-ul e valid
                        ;sectorul adresat este diferit, sector nou
                        ;
815B  C5                 NEWSEC: PUSH    BC      ;salveaza un nou sector de informatie
815C  CD 81C7                  CALL    WRSTAT  ;da o fuga pe la scriere buffer
815F  CD 83A0                  CALL    DSKSEL  ;cauta discul selectat
8162  CD 844D                  CALL    FLSEEK  ;cauta pista selectata
8165  C1                       POP     BC
8166  21 FB54                  LD      HL,SECT
8169  71                       LD      (HL),C  ;actualizeaza numarul sectorului
816A  2B                       DEC     HL
816B  70                       LD      (HL),B  ;actualizeaza numarul capului
816C  2B                       DEC     HL
816D  2B                       DEC     HL
816E  78                       LD      A,B
816F  07                       RLCA
8170  07                       RLCA
8171  47                       LD      B,A
8172  3E 03                    LD      A,003H
8174  A6                       AND     (HL)
8175  B0                       OR      B
8176  77                       LD      (HL),A  ;inca o actualizare a numarului capului
8177  3A FB43                  LD      A,(RSFLG)       ;adu fanionul de citire sector
```

```
817A  B7                  OR    A              ;este pre-citire?
817B  28 12               JR    Z,CRSEC        ;nu, salveaza
817D  2A FB3A             LD    HL,(DMAADD)
8180  E5                  PUSH  HL             ;pastrez adresa DMA utilizator
8181  21 F758             LD    HL,RWBUF       ;noua adresa DMA, temporara
8184  22 FB3A             LD    (DMAADD),HL
8187  CD 81F2             CALL  FLREAD         ;citire
818A  E1                  POP   HL
818B  22 FB3A             LD    (DMAADD),HL    ;refac adresa DMA utilizator
818E  C0                  RET   NZ             ;revenire pentru eroare
818F  3A FB32       CRSEC: LD   A,(SEKDEN) ;preia tipul densitatii, pentru a-i
8192  1F                  RRA                  ;stabili amplasamentul in cadrul sf
8193  3A FB35             LD    A,(LACSEC) ;preia codul sectorului logic
8196  38 49               JR    C,COMM2    ;salt pentru 256 octeti/sf
8198  E6 03               AND   003H       ;512 octeti/sf
819A  0F                  RRCA
819B  0F            COMM1: RRCA
819C  6F                  LD    L,A
819D  26 00               LD    H,000H
819F  29                  ADD   HL,HL
81A0  11 F758             LD    DE,RWBUF
81A3  19            COMMX: ADD   HL,DE
81A4  EB                  EX    DE,HL      ;adresa sectorului logic in DE
81A5  2A FB3A             LD    HL,(DMAADD);preiau si adresa DMA
81A8  0E 80               LD    C,080H     ;voi muta un bloc de 128 de octeti
81AA  3A FB3C             LD    A,(DSKOP)  ;adu tipul operatiei
81AD  B7                  OR    A          ;de care-i?
81AE  28 06               JR    Z,RWMOVE   ;daca sintem in citire
81B0  3E 01               LD    A,001H     ;mai tirziu vom avea de lucru cu
81B2  32 FB2E             LD    (BUFWRT),A ;bufferul de scriere, trebuie marcat
81B5  EB                  EX    DE,HL      ;inversez adresele bufferelor
81B6  C5            RWMOVE: PUSH  BC
81B7  EB                  EX    DE,HL
81B8  06 00               LD    B,000H
81BA  CD F671             CALL  PIR        ;si-acum chem rutina de mutare sector
81BD  EB                  EX    DE,HL
81BE  C1                  POP   BC
81BF  3A FB3D       RWDONE: LD    A,(WRTYPE) ;preiau tipul scrierii
81C2  FE 01               CP    001H       ;caut pentru scriere
81C4  3E 00               LD    A,0        ;pregatesc pentru revenire
81C6  C0                  RET   NZ         ;daca nu-i pentru scriere, revenire
                    ;cauta bufferul de scriere si-l reactualizeaza, daca-i nevoie
                    ;
81C7  21 FB2E       WRSTAT: LD    HL,BUFWRT
81CA  7E                  LD    A,(HL)
81CB  36 00               LD    (HL),000H  ;nu va mai fi buffer de scriere
81CD  B7                  OR    A          ;buffer ocupat?
81CE  C8                  RET   Z          ;nu, revenire
81CF  2A FB3A             LD    HL,(DMAADD);salveaza adresa DMA utilizator in
81D2  E5                  PUSH  HL         ;stiva
81D3  21 F758             LD    HL,RWBUF
81D6  22 FB3A             LD    (DMAADD),HL
81D9  CD 81EC             CALL  FLWRT      ;transfera bufferul
81DC  E1                  POP   HL
81DD  22 FB3A             LD    (DMAADD),HL;restaureaza adresa DMA utilizator
81E0  C9                  RET
                    ;
81E1  E6 01         COMM2: AND   001H
81E3  0F                  RRCA
81E4  5F                  LD    E,A
81E5  16 00               LD    D,000H
```

```
81E7  21 F758          LD    HL,RWBUF
81EA  18 B7            JR    COMMX

                       ;scrie sector descris in tabela de scriere/citire
                       ;Intrari: adresa DMA-de aici incepe transferul
                       ;
81EC  06 45      FLWRT: LD    B,045H  ;comanda de scriere
81EE  3E 01            LD    A,001H
81F0  18 03            JR    FLRDWR

                       ;citeste sectorul descris prin tabela de scriere/citire
                       ;Intrari: adresa DMA-de aici incepe transferul
                       ;
81F2  06 46      FLREAD: LD    B,046H  ;comanda de citire
81F4  AF               XOR   A
81F5  32 FB2D    FLRDWR: LD    (RWFLG),A
81F8  AF         JP001:  XOR   A               ;initializeaza cu zero RECFL
81F9  32 FB44            LD    (RECFL),A      ;urmeaza a fi utilizat in cazul
                       ;primului acces eronat, pentru a semnaliza buclei de reluare a
                       ;operatiei daca a trecut de primul pas, caz in care declanseaza
                       ;automat o manevra RECALIBRARE/SEEK pentru o posibila corectie de
                       ;pozitionare a capului; aici, cu 0, valideaza corectia
81FC  3E 10              LD    A,RETRY        ;adu numarul maxim de incercari
81FE  32 FB2C    FLRETR: LD    (RTCNT),A
8201  3A FB54    JR008:  LD    A,(SECT);citind un singur sector, fortez ultimul
8204  32 FB56            LD    (EOT),A ;sector la aceiasi valoare cu primul
8207  C5               PUSH  BC      ;salvez in stiva codul comenzii (B)
8208  0E 08              LD    C,008H  ;C va memora numarul de parametri
820A  21 F53F            LD    HL,RWRET;pregatesc viitoarea revenire, executia
820D  E5               PUSH  HL      ;rutinei de transfer sector
820E  CB 40              BIT   0,B     ;care-i tipul operatiei?
8210  21 F4F7            LD    HL,MINIWR ;probabil scriere! In HL adresa rutinei
8213  20 03              JR    NZ,JR009  ;Nu! De data asta-i citire
8215  21 F50D            LD    HL,MINIRD ;in HL adresa rutinei de citire sector
8218  E5       JR009:  PUSH  HL      ;pregateste in stiva adresa rutinei
8219  21 F4CE            LD    HL,L427 ;nu inainte de a salva adresa rutinei de
821C  E3               EX    (SP),HL ;preluare rezultat
821D  E5               PUSH  HL      ;acum pot pune si adresa rutinei transfer
821E  78               LD    A,B     ;aduc codul operatiei
821F  C3 F535          JP    RWP     ;execut comanda
8222  CD F4C5  RWRET1: CALL  BD13
8225  C1               POP   BC
8226  FE 40              CP    040H    ;asa ar trebui sa arate parametrul
8228  20 08              JR    NZ,JR010;rezultat ST0 (putin mascat) pentru
822A  23               INC   HL      ;operatie corecta
822B  7E               LD    A,(HL)  ;preiau ST1
822C  FE 80              CP    080H    ;daca si ST1=#80, atunci este OK
822E  20 02              JR    NZ,JR010
8230  AF               XOR   A       ;(de data asta) am reusit!
8231  C9               RET           ;nu mai am alta treaba
8232  C5       JR010:  PUSH  BC
8233  3A FB46            LD    A,(RWSTBL)
8236  E6 18              AND   018H
8238  28 06              JR    Z,DRDY
823A  CD 8014            CALL  DRNRDY
823D  C1               POP   BC
823E  18 C1              JR    JR008
                       ;
8240  3A FB48    DRDY:   LD    A,(RWSTBL+2)
8243  E6 10              AND   010H
8245  28 10              JR    Z,FLERR
```

```
8247  3A FB44                   LD    A,(RECFL)
824A  B7                        OR    A
824B  20 0A                     JR    NZ,FLERR
824D  2F                        CPL
824E  32 FB44                   LD    (RECFL),A
8251  CD 8440                   CALL  RCAL
8254  CD 844D                   CALL  FLSEEK
8257  C1                FLERR:  POP   BC
8258  3A FB2C                   LD    A,(RTCNT)
825B  3D                        DEC   A
825C  F2 81FE                   JP    P,FLRETR
                          ;Erorile au dreptul cel mult la corectii, dar in nici un caz la
                          ;comentarii! Asa ca, incepind din acest moment, orice comentariu
                          ;este de prisos...
                          ;
825F  CD 8061                   CALL  PTRDWR
8262  21 FB47                   LD    HL,RWSTBL+1
8265  7E                        LD    A,(HL)
8266  E6 80                     AND   080H
8268  28 10                     JR    Z,FLE1
826A  CD 8001                   CALL  TYPE
826D  65 6E 64 20               DB    "end of track$"
8271  6F 66 20 74
8275  72 61 63 6B
8279  24
827A  7E                  FLE1: LD    A,(HL)
827B  E6 20                     AND   020H
827D  28 1D                     JR    Z,FLE2
827F  23                        INC   HL
8280  7E                        LD    A,(HL)
8281  2B                        DEC   HL
8282  E6 20                     AND   020H
8284  28 0C                     JR    Z,IDCRC
8286  CD 8001                   CALL  TYPE
8289  64 61 74 61               DB    "data CRC$"
828D  20 43 52 43
8291  24
8292  CD 8001             IDCRC: CALL  TYPE
8295  49 44 20 43               DB    "ID CRC$"
8299  52 43 24
829C  7E                  FLE2: LD    A,(HL)
829D  E6 04                     AND   004H
829F  28 14                     JR    Z,FLE3
82A1  CD 8001                   CALL  TYPE
82A4  73 65 63 74               DB    "sector not found$"
82A8  6F 72 20 6E
82AC  6F 74 20 66
82B0  6F 75 6E 64
82B4  24
82B5  7E                  FLE3: LD    A,(HL)
82B6  E6 02                     AND   002H
82B8  C4 8080                   CALL  NZ,PNPROT
82BB  7E                        LD    A,(HL)
82BC  E6 01                     AND   001H
82BE  28 26                     JR    Z,EREXIT
82C0  23                        INC   HL
82C1  7E                        LD    A,(HL)
82C2  E6 01                     AND   001H
82C4  28 0A                     JR    Z,IDAM
82C6  CD 8001                   CALL  TYPE
82C9  64 61 74 61               DB    "data$"
```

```
82CD  24
82CE  18 06                    JR     IDERR
82D0  CD 8001          IDAM:   CALL   TYPE
82D3  49 44 24                 DB     "ID$"
82D6  CD 8001          IDERR:  CALL   TYPE
82D9  41 44 44 52              DB     "ADDRESS MARK$"
82DD  45 53 53 20
82E1  4D 41 52 4B
82E5  24
82E6  CD 8001          EREXIT: CALL   TYPE    ;Crezi ca mai sint sanse?...
82E9  45 52 52 4F              DB     "ERROR"
82ED  52
82EE  0D 0A                    DB     0DH,0AH
82F0  43 61 6E 63              DB     "Cancel,Ignore,Retry ?$"
82F4  65 6C 2C 49
82F8  67 6E 6F 72
82FC  65 2C 52 65
8300  74 72 79 20
8304  3F 24
8306  C5                      PUSH   BC
8307  CD 86C3                CALL   KBINP
830A  F5                      PUSH   AF
830B  4F                      LD     C,A
830C  CD 8836                 CALL   DISPO
830F  CD 8001                 CALL   TYPE
8312  0D 0A 24                DB     0DH,0AH,'$'
8315  F1                      POP    AF
8316  C1                      POP    BC
8317  E6 DF            JR011: AND    0DFH    ;...ce-ar fi sa mai incerc!?
8319  FE 49                   CP     "I"
831B  28 08                   JR     Z,ERRIGN
831D  FE 52                   CP     "R"     ;as putea relua operatia
831F  CA 81F8                 JP     Z,JP001 ;...mai stii, poate de data asta reusesc!
8322  AF               ERCONT: XOR    A       ;marcheaza eroare permanenta
8323  3D                      DEC    A
8324  C9                      RET
8325  AF               ERRIGN: XOR    A       ;ignora eroarea
8326  C9                      RET

                       ;translatare sector
                       ;Intrari: BC - codul sectorului logic
                       ;         DE - adresa tabelei de translatare
                       ;Iesiri : HL - codul sectorului translatat
                       ;
8327  79               SECTR1: LD     A,C
8328  32 FB39                 LD     (SEKSEC),A      ;memoreaza sectorul netranslatat
832B  3A FB32                 LD     A,(SEKDEN)
832E  FE 10                   CP     010H    ;RAMDISC-ul nu are nevoie de translatare
8330  28 04                   JR     Z,ALFA0 ;asa ca am sa-i las sectorul neschimbat
8332  FE 20                   CP     020H    ;RAMDISC extern (Micromext)?
8334  20 03                   JR     NZ,SCTTRN       ;nu, salt
8336  60               ALFA0:  LD     H,B
8337  69                       LD     L,C
8338  C9                       RET
8339  1F               SCTTRN: RRA            ;...treaba se complica
833A  38 15                   JR     C,D256  ;256 octeti/sf?
833C  21 F73A                 LD     HL,XLT1 ;Nu, 512. In HL am adresa tabelei de
833F  79                      LD     A,C     ;translatare
8340  B7                      OR     A
8341  F5                      PUSH   AF
8342  1F                      RRA
```

```
8343  B7                OR    A
8344  1F                RRA
8345  4F                LD    C,A
8346  09                ADD   HL,BC
8347  7E                LD    A,(HL)  ;o privire prin tabela sectoarelor de
8348  07                RLCA          ;translatare
8349  07                RLCA
834A  4F                LD    C,A
834B  F1                POP   AF
834C  E6 03             AND   003H
834E  B1                OR    C
834F  6F                LD    L,A
8350  C9                RET
                  ;256 octeti/sf. Aici problemele sint ceva mai simple...
                  ;
8351  1E 10       D256:  LD    E,010H
8353  79                LD    A,C
8354  B7                OR    A
8355  1F                RRA
8356  F5                PUSH  AF
8357  4F                LD    C,A
8358  AF                XOR   A
8359  0D           FLACEX: DEC  C
835A  FA 8366           JP    M,DDX2
835D  C6 08             ADD   A,008H
835F  BB                CP    E
8360  38 F7             JR    C,FLACEX
8362  93                SUB   E
8363  3C                INC   A
8364  18 F3             JR    FLACEX
8366  4F           DDX2:  LD    C,A
8367  F1                POP   AF
8368  79                LD    A,C
8369  17                RLA
836A  6F                LD    L,A
836B  C9                RET

                  ;selectie disc
                  ;Intrari: C  - numarul discului, obtinut prin scaderea, din
                  ;            valoarea asociata codului sau ASCII, a constantei
                  ;            #41 (i.e. pentru drive-ul A, 00 = #41-#41)
                  ;Iesiri : HL - rezultatul selectiei: 0 - densitate neidentificata
                  ;                                    altfel, densitate identificata
836C  79          SELDSK1:LD   A,C
836D  21 0000           LD     HL,0000H;pregatiri pentru un posibil rateu
8370  FE 0C              CP     00CH    ;este drive-ul M (SYSTEM RAMDISC)?
8372  20 03             JR     NZ,ABCD ;Nu, salt
8374  3E 02             LD     A,002H  ;Da, corectie locala, putin semnificativa
8376  4F                LD     C,A     ;codul corectat depus in C
8377  FE 04        ABCD:  CP     NRDSKS  ;este mai mare decit codul maxim?
8379  D0                RET    NC      ;Da, eroare. Nu cunoastem asemenea disc!
837A  32 FB2F           LD     (SEKDSK),A ;cod corect, sa-l tinem minte
837D  11 F6D8           LD     DE,DPBASE  ;tabela tabelelor de parametri; pe
8380  87                ADD    A,A        ;baza adresei ei de debut calculez mai
8381  87                ADD    A,A        ;intii un pointer de adresa pentru
8382  87                ADD    A,A        ;tabela de parametri a discului curent
8383  87                ADD    A,A        ;urmind ca mai tirziu sa ma pozitionez
8384  81                ADD    A,C        ;chiar pe cel de-al 17-lea parametru
8385  6F                LD     L,A        ;din tabela asociata drive-ului curent
8386  19                ADD    HL,DE      ;care reprezinta tipul densitatii
8387  22 FB30           LD     (CURDPH),HL;memoreaza pointerul de adresa al
```

```
838A  E5              PUSH  HL          ;al tabelei de parametri a discului
838B  11 0010         LD    DE,0010H    ;pozitionare pe parametrul tip
838E  19              ADD   HL,DE       ;densitate
838F  7E              LD    A,(HL)
8390  32 FB32         LD    (SEKDEN),A ;memoreaza densitatea utilizator
8393  3C              INC   A           ;o cunoastem?
8394  E1              POP   HL
8395  C0              RET   NZ          ;Da, revenire din selectie disc
8396  CD 81C7         CALL  WRSTAT      ;cauta buffer tiparibil, scrie-l daca
8399  CD 83A0         CALL  DSKSEL      ;este plin; selecteaza drive-ul si
839C  2A FB30         LD    HL,(CURDPH);cauta-i densitatea, dupa care se face
839F  C9              RET               ;revenire, cu HL adresind DPH-ul curent

                    ;compara discul utilizator selectat cu discul curent selectat
                    ;si realizeaza o actualizare a pistei, daca este nevoie
                    ;
83A0  3A FB2F       DSKSEL: LD    A,(SEKDSK) ;preia noul numar de disc
83A3  5F                    LD    E,A
83A4  16 00                 LD    D,000H
83A6  3A FB51               LD    A,(DSKNO)  ;il preia si pe cel vechi
83A9  BB                    CP    E          ;seamana?
83AA  28 17                 JR    Z,CKDEN    ;da, actualizeaza tabela pistelor
83AC  D5                    PUSH  DE
83AD  21 FB4D               LD    HL,TRKTBL
83B0  5F                    LD    E,A
83B1  19                    ADD   HL,DE
83B2  3A FB52               LD    A,(TRKNO)
83B5  77                    LD    (HL),A     ;memoreaza vechiul numar de pista,
83B6  21 FB4D               LD    HL,TRKTBL  ;care reprezinta ultima pista pe care
83B9  D1                    POP   DE         ;s-a pozitionat capul drive-ului de
83BA  19                    ADD   HL,DE      ;curind abandonat (vechi)
83BB  7E                    LD    A,(HL)     ;preia noul numar de pista
83BC  32 FB52               LD    (TRKNO),A
83BF  7B                    LD    A,E
83C0  32 FB51               LD    (DSKNO),A  ;memoreaza noul nume de disc
                    ;cauta densitatea discului curent
                    ;o schimba, daca nu este aceiasi, sau o cauta, daca n-o cunoaste
                    ;
83C3  3A FB32       CKDEN:  LD    A,(SEKDEN) ;preia noua densitate
83C6  FE FF                 CP    0FFH       ;necunoscuta?
83C8  28 1E                 JR    Z,FINDEN   ;da, cauta densitatea
83CA  21 FB38       SETDEN: LD    HL,CURDEN  ;pozitionare pe densitatea curenta
83CD  BE                    CP    (HL)       ;aceiasi cu cea noua?
83CE  C8                    RET   Z          ;da, revenire
83CF  77                    LD    (HL),A     ;actualizeaza indicator densitate
83D0  11 FB55               LD    DE,N       ;curenta; pozitionare pe tabela de
83D3  06 04                 LD    B,4        ;scriere/citire; determina densitatea
83D5  21 F6AE               LD    HL,D2VAL
83D8  1F                    RRA
83D9  38 03                 JR    C,SETD1    ;o fi dubla densitate, 256 b/sf?
83DB  21 F6B4               LD    HL,D5VAL   ;nu, este dubla densitate, dar 512b/sf
83DE  7E            SETD1:  LD    A,(HL)     ;actualizeaza tabela scriere/citire cu
83DF  12                    LD    (DE),A     ;parametri corespunzatori densitatii
83E0  23                    INC   HL         ;selectate
83E1  13                    INC   DE         ;la DPBPNT memoreaza adresa adresei
83E2  10 FA                 DJNZ  SETD1      ;tabelei de parametri ce defineste
83E4  22 FB33               LD    (DPBPNT),HL;densitatea selectata (DPBD2 pentru
83E7  C9                    RET              ;256b/sf si DPBD5 pentru 512b/sf)
                    ;identifica densitatea discului
                    ;
83E8  CD 843C       FINDEN: CALL  RECAL      ;trimiti capul acasa (recalibrare)
```

```
83EB  01 0004          LD    BC,4
83EE  CD F52B          CALL  SETTRK     ;dupa care-l duci pe pista 2
83F1  CD 844D          CALL  FLSEEK
83F4  3E 01            LD    A,001H     ;presupui ca este dubla densitate cu
83F6  CD 8429          CALL  READID     ;256b/sf; lansezi o citire a cimpului
83F9  28 0E            JR    Z,DFOUND;ID de pe disc. Aceiasi densitate? Da, OK
83FB  3E 02            LD    A,002H     ;nu, e posibil sa fie 512b/sf
83FD  CD 8429          CALL  READID     ;mai incerci o citire de ID
8400  28 07            JR    Z,DFOUND   ;aceiasi, OK
8402  21 0000          LD    HL,0000H   ;densitatea nu se poate determina
8405  22 FB30          LD    (CURDPH),HL;la CURDPH marchez cu 0 eroare de
8408  C9               RET    ;neidentificare densitate, dupa care revin
                       ;
8409  CD 843C    DFOUND: CALL  RECAL
840C  2A FB30          LD    HL,(CURDPH)     ;preia DPH-ul curent
840F  11 000A          LD    DE,000AH
8412  19               ADD   HL,DE
8413  EB               EX    DE,HL
8414  2A FB33          LD    HL,(DPBPNT)     ;preia pointerul DPB identificat
8417  7E               LD    A,(HL)
8418  12               LD    (DE),A
8419  23               INC   HL
841A  13               INC   DE
841B  7E               LD    A,(HL)
841C  12               LD    (DE),A  ;memoreaza adresa tabelei care defineste
841D  21 0005          LD    HL,0005H;densitatea curenta
8420  19               ADD   HL,DE   ;fanionul densitatii este repozitionat
8421  3A FB38          LD    A,(CURDEN)      ;densitatea curenta o memoreaza
8424  77               LD    (HL),A          ;in DPH
8425  32 FB32          LD    (SEKDEN),A      ;si la SEKDEN
8428  C9               RET

                       ;citeste cimpul ID al unui sector, pentru a determina densitatea
                       ;Iesiri: fanionul Z: 0 pentru densitate identificata
                       ;                   1 pentru densitate neidentificata
                       ;
8429  CD 83CA    READID: CALL  SETDEN
842C  3E 4A        RDID:  LD    A,04AH  ;octet de comanda pentru operatia READID
842E  0E 01               LD    C,001   ;numarul parametrilor de comanda
8430  CD 849B             CALL  EX      ;comanda efectiva+executie+rezultat
8433  C0                  RET   NZ      ;revenire pentru operatie nereusita
8434  21 FB4C             LD    HL,RWSTBL+6
8437  3A FB38             LD    A,(CURDEN)
843A  BE                  CP    (HL)    ;compara parametrul densitate curenta cu
843B  C9                  RET           ;parametrul similar preluat de pe disc

                       ;pozitioneaza capul de scriere/citire pe pista 0
                       ;
843C  AF           RECAL: XOR   A
843D  32 FB36             LD    (SEKTRK),A
8440  AF            RCAL: XOR   A
8441  32 FB52             LD    (TRKNO),A
8444  CD 8014             CALL  DRNRDY
8447  06 07               LD    B,007H
8449  0E 01               LD    C,001H  ;un parametru
844B  18 0D               JR    EXEC

                       ;pozitioneaza capul de scriere/citire pe pista al carei numar
                       ;este dat de variabila SEKTRK
                       ;Iesiri: fanionul Z: 1, pozitionare reusita
                       ;                   0, pozitionare nereusita
```

```
                        ;
844D  3A FB36           FLSEEK: LD      A,(SEKTRK)
8450  21 FB52                   LD      HL,TRKNO
8453  BE                        CP      (HL)    ;sint cumva pe aceiasi pista?
8454  C8                        RET     Z       ;da, revenire
8455  77                        LD      (HL),A  ;actualizez TRKNO
8456  06 0F                     LD      B,00FH  ;octet de comanda pentru operatie
8458  0E 02                     LD      C,002H  ;numar de parametri
845A  CD 848B           EXEC:   CALL    OPENDMA ;ii spune lui 8272 ca trebuie, temporar,
845D  C5                        PUSH    BC      ;sa lucreze in mod DMA, astfel incit sa
845E  78                        LD      A,B     ;putem beneficia de semnalizarile INT72
845F  CD F498                   CALL    AD13    ;comanda+executie
8462  CD F4C5                   CALL    BD13
8465  01 5EFD                   LD      BC,5EFDH;comanda fiind in curs de executie,
8468  ED 78           D14:    IN      A,(C)   ;semnalul de intrerupere INT72 (furnizat
846A  E6 10                     AND     10H     ;de catre 8272) ne spune, pentru starea
846C  28 FA                     JR      Z,D14   ;unu logic, ca executia s-a finalizat
846E  3E 08                     LD      A,008H  ;codul comenzii CITIRE STARE INTRERUPERE,
8470  CD F439                   CALL    L400    ;emis in scopul
8473  CD F4CE                   CALL    L427    ;preluarii rezultatului privitor la
8476  C1                        POP     BC      ;executia operatiei
                        ;rutina care recomuta modul de lucru al lui8272 pe tratarea prin
                        ;program
                        ;
8477  F5                CLOSDMA:PUSH    AF
8478  C5                        PUSH    BC
8479  E5                        PUSH    HL
847A  3E 03                     LD      A,3     ;al treilea parametru al comenzii de
                                                ;specificare, cu cimpul ND (bitul b0) pe
                                                ;optiunea nonDMA (prin program)
847C  21 8494           CDMA:   LD      HL,PSPEC+2      ;adresa in memorie a parametrului
847F  77                        LD      (HL),A          ;de mai sus
8480  2B                        DEC     HL      ;se actualizeaza parametrul al trelea
8481  2B                        DEC     HL      ;pozitionare pe primul parametru
8482  3E 03                     LD      A,3     ;in A incarcam numarul de parametri
8484  CD F4BA                   CALL    L41C    ;emitem cei trei parametri
8487  E1                        POP     HL
8488  C1                        POP     BC
8489  F1                        POP     AF
848A  C9                        RET
                        ;rutina care trece 8272 in modul de lucru DMA
                        ;
848B  F5                OPENDMA:PUSH    AF
848C  C5                        PUSH    BC
848D  E5                        PUSH    HL
848E  3E 02                     LD      A,2     ;al treilea parametru pentru mod de lucru
8490  18 EA                     JR      CDMA    ;DMA, avind pe cimpul ND valoarea 0
                        ;tabela celor trei parametri ai comenzii de specificare
                        ;
8492  03                PSPEC:  DB      3       ;codul comenzii
8493  A1                        DB      0A1H    ;"SRT"= 10 si "HUT"=1
8494  03                        DB      3       ;"HLT"=1, iar ND configurabil functie de
                                                ;modul de lucru cu 8272
                        ;
8495  12                CYBR:   LD      (DE),A
8496  13                        INC     DE
8497  0D                        DEC     C
8498  20 FB                     JR      NZ,CYBR
849A  C9                        RET;
849B  CD F4B2           EX:     CALL    D13
849E  C3 F4CE                   JP      L427
```

```
                        ;emite un caracter catre imprimanta paralela
                        ;Intrari: C - codul ASCII al caracterului
                        ;
84A1  79                LPT:    LD      A,C
84A2  F3                        DI              ;e bine sa n-avem de lucru cu intreruperile
84A3  F5                        PUSH    AF
84A4  3E 0F             JR013:  LD      A,00FH  ;imprimanta este ocupata?
84A6  DB FD                     IN      A,(0FDH);preia in A starea lui nBUSY1
84A8  0F                        RRCA            ;duc bitul (D0), asociat lui BUSY1, in CY
84A9  38 F9             LPTB:   JR      C,JR013 ;daca-i 1, imprimanta ocupata, mai astept
84AB  F1                        POP     AF
84AC  C5                        PUSH    BC
84AD  01 0FFD                   LD      BC,0FFDH
84B0  CB BF                     RES     7,A
84B2  ED 79                     OUT     (C),A   ;incarc codul caracterului in portul date
84B4  06 1F                     LD      B,01FH
84B6  FD 21 FB70                LD      IY,ZXST0
84BA  FD 7E 14                  LD      A,(IY+14H);preiau imaginea portului #1FFD si-i
84BD  CB E7             LPTS1:  SET     4,A     ;modulez bitul (D3) asociat strobului
84BF  ED 79                     OUT     (C),A   ;mai intii il trec pe zero (spre
84C1  CB A7             LPTS2:  RES     4,A     ;imprimanta se transmite inversat)
84C3  ED 79                     OUT     (C),A   ;apoi pe unu
84C5  CB E7             LPTS3:  SET     4,A
84C7  ED 79                     OUT     (C),A   ;si iarasi pe zero
84C9  C1                        POP     BC
84CA  C9                        RET
                        ;
84CB  3E DF             SIOAST: LD      A,0DFH
84CD  DB FD                     IN      A,(0FDH)
84CF  CB 47                     BIT     0,A
84D1  3E 00                     LD      A,0
84D3  C0                        RET     NZ
84D4  3D                        DEC     A
84D5  C9                        RET
                        ;
84D6  C5                XMITA:  PUSH    BC
84D7  79                        LD      A,C
84D8  F5                        PUSH    AF
84D9  01 DFFD                   LD      BC,0DFFDH
84DC  ED 78             XMI1:   IN      A,(C)
84DE  17                        RLA
84DF  30 FB                     JR      NC,XMI1
84E1  ED 78             XMI2:   IN      A,(C)
84E3  1F                        RRA
84E4  30 FB                     JR      NC,XMI2
84E6  05                        DEC     B
84E7  F1                        POP     AF
84E8  ED 79                     OUT     (C),A
84EA  C1                        POP     BC
84EB  C9                        RET

84EC  C5                RECVA:  PUSH    BC
84ED  01 DFFD                   LD      BC,0DFFDH
84F0  ED 78             REC1:   IN      A,(C)
84F2  E6 02                     AND     002H
84F4  28 FA                     JR      Z,REC1
84F6  05                        DEC     B
84F7  ED 78                     IN      A,(C)
84F9  C1                        POP     BC
84FA  C9                        RET
```

```
84FB                    SIOBST:
84FB                    RECVB:
84FB  C9                XMITB:  RET

84FC  21 86AB           PUNCH1: LD      HL,PUNTAB
84FF  3A 0003                   LD      A,(3)
8502  0F                        RRCA
8503  0F                        RRCA
8504  0F                PUN1:   RRCA
8505  18 23                     JR      INDEX

8507  21 86A3           READER1:LD      HL,RDRTAB
850A  3A 0003                   LD      A,(3)
850D  18 F5                     JR      PUN1

850F  21 868B           RDRST1: LD      HL,RDSTAB
8512  18 F6                     JR      READER1+3


8514  3A 0003           LIST1:  LD      A,(3)
8517  21 8693           LIST1X: LD      HL,LISTAB
851A  07                        RLCA
851B  07                        RLCA
851C  18 0B                     JR      JR014

851E  21 86B3           CONIN1: LD      HL,CITAB
8521  18 03                     JR      JR015

8523  21 86BB           CONOUT1:LD      HL,COTAB
8526  3A 0003           JR015:  LD      A,(3)
8529  07                JR014:  RLCA
852A  E6 06             INDEX:  AND     006H
852C  CD 854E                   CALL    RAD
852F  D5                        PUSH    DE
8530  C9                        RET

8531  3A 0003           CONST1: LD      A,(3)
8534  4F                        LD      C,A
8535  E6 03                     AND     3
8537  FE 02                     CP      2
8539  20 0E                     JR      NZ,CONSTX+3
853B  79                        LD      A,C
853C  0F                        RRCA
853D  E6 06                     AND     6
853F  21 868B                   LD      HL,RDSTAB
8542  CD 854E                   CALL    RAD
8545  EB                        EX      DE,HL
8546  22 869F           CONSTX: LD      (STSTAB+4),HL
8549  21 869B                   LD      HL,STSTAB
854C  18 D3                     JR      CONIN1+3


854E  CD 8555           RAD:    CALL    AXH
8551  5E                        LD      E,(HL)
8552  23                        INC     HL
8553  56                        LD      D,(HL)
8554  C9                        RET

8555  85                AXH:    ADD     A,L
8556  6F                        LD      L,A
```

```
8557 7C          LD    A,H
8558 CE 00       ADC   A,0
855A 67          LD    H,A
855B C9          RET
                 ;
855C F3          ERDSKRW:DI
855D 2A FB35     LD    HL,(LACSEC)
8560 AF          XOR   A
8561 CB 3C       SRL   H
8563 CB 1D       RR    L
8565 1F          RRA
8566 D3 0F       OUT   (0FH),A
8568 7D          LD    A,L
8569 D3 1F       OUT   (1FH),A
856B 7C          LD    A,H
856C E6 0F       AND   0FH
856E D3 2F       OUT   (2FH),A
8570 2A FB3A     LD    HL,(DMAADD)
8573 3A FB3C     LD    A,(DSKOP)
8576 B7          OR    A
8577 C3 F57F     JP    MOWGLI
                 ;
857A F3          RDSKRW: DI
857B 2A FB35     LD    HL,(LACSEC)
857E 11 0080     LD    DE,8*DRAMD
8581 19          ADD   HL,DE
8582 22 FB35     LD    (LACSEC),HL
8585 AF          XOR   A
8586 3E 80       LD    A,80H
8588 08          RDS4:  EX    AF,AF'
8589 3A FB35     LD    A,(LACSEC)
858C CB 3F       SRL   A
858E 1E 00       LD    E,000H
8590 CB 1B       RR    E
8592 16 40       LD    D,040H
8594 E6 3F       AND   03FH
8596 82          ADD   A,D
8597 57          LD    D,A
8598 2A FB3A     LD    HL,(DMAADD)
859B 01 0B30     LD    BC,0B30H
859E 08          EX    AF,AF'
859F 20 1F       JR    NZ,RDS0
85A1 08          EX    AF,AF'
85A2 7C          LD    A,H
85A3 FE 80       CP    080H
85A5 30 2B       JR    NC,ARD0
85A7 CB A1       RES   4,C
85A9 CB 88       RES   1,B
85AB FE 7F       CP    7FH
85AD 20 0A       JR    NZ,RDS1
85AF 7D          LD    A,L
85B0 FE 81       CP    81H
85B2 38 05       JR    C,RDS1
85B4 08          EX    AF,AF'
85B5 7D          LD    A,L
85B6 2F          CPL
85B7 3C          INC   A
85B8 08          EX    AF,AF'
85B9 3E 40       RDS1:  LD    A,40H
85BB 82          ADD   A,D
85BC 57          LD    D,A
```

```
85BD  08              EX    AF,AF'
85BE  30 11           JR    NC,RDS3
85C0  08        RDS0: EX    AF,AF'
85C1  D9              EXX
85C2  01 0B30         LD    BC,0B30H
85C5  3A FB3C         LD    A,(DSKOP)
85C8  B7              OR    A
85C9  20 01           JR    NZ,RDS6
85CB  EB              EX    DE,HL
85CC  7A        RDS6: LD    A,D
85CD  D6 40           SUB   40H
85CF  57              LD    D,A
85D0  08              EX    AF,AF'
85D1  08        RDS3: EX    AF,AF'
85D2  3A FB3C   ARD0: LD    A,(DSKOP)
85D5  B7              OR    A
85D6  20 01           JR    NZ,RDSKW
85D8  EB              EX    DE,HL
85D9  78        RDSKW: LD   A,B
85DA  D9              EXX
85DB  47              LD    B,A
85DC  D9              EXX
85DD  79              LD    A,C
85DE  D9              EXX
85DF  4F              LD    C,A
85E0  3A FB35         LD    A,(LACSEC)
85E3  17              RLA
85E4  3A FB36         LD    A,(SEKTRK)
85E7  17              RLA
85E8  E6 07           AND   007H
85EA  B1              OR    C
85EB  58              LD    E,B
85EC  C3 F548         JP    KAPOOR
85EF  D9        KAP:  EXX
85F0  08              EX    AF,AF'
85F1  DA 8588         JP    C,RDS4
85F4  2A FB35         LD    HL,(LACSEC)
85F7  11 FF80         LD    DE,-8*DRAMD
85FA  19              ADD   HL,DE
85FB  22 FB35         LD    (LACSEC),HL
85FE  AF              XOR   A
85FF  C9              RET

                ;SUBRUTINA INTERFATA PERFORATOR DE BANDA "DT1055"
                ;
8600  F5        DT105S: PUSH  AF
8601  C5              PUSH  BC
8602  3E 5E     DT1:  LD    A,5EH
8604  DB FD           IN    A,(0FDH)
8606  CB 77           BIT   6,A          ;BUSY are nivel HI ?
8608  28 F8           JR    Z,DT1
860A  79              LD    A,C          ;DA
860B  B7              OR    A
860C  EA 8611         JP    PE,DT2
860F  F6 80           OR    80H          ;Creaza bitul de paritate para
8611  01 0FFD   DT2:  LD    BC,0FFDH
8614  ED 79           OUT   (C),A        ;Emite datele
8616  06 5F           LD    B,5FH
8618  3E 01           LD    A,1
861A  ED 79           OUT   (C),A        ;STROBE nivel HI
861C  05              DEC   B
```

```
861D ED 78          DT3:    IN      A,(C)
861F CB 77                  BIT     6,A             ;BUSY are nivel LO ?
8621 20 FA                  JR      NZ,DT3
8623 04                     INC     B               ;DA
8624 AF                     XOR     A
8625 ED 79                  OUT     (C),A           ;STROBE nivel LO
8627 C1                     POP     BC
8628 F1                     POP     AF
8629 C9                     RET

                    ;SUBRUTINA INTERFATA CITITOR DE BANDA PERFORATA "CONSUL"
                    ;
862A 00             CAP:    DB      0               ;Fanion ce indica daca a fost
                                                    ;citit un caracter nenul
862B C5             CONSUL: PUSH    BC
862C 01 5FFD                LD      BC,5FFDH
862F 3A 862A                LD      A,(CAP)         ;Se citeste primul caracter de pe
8632 B7                     OR      A               ;banda ?
8633 20 0A                  JR      NZ,ECO2
8635 3E 07                  LD      A,7             ;DA
8637 ED 79                  OUT     (C),A           ;START se pozitioneaza HI
8639 00                     NOP                     ;Temporizare
863A 00                     NOP
863B 3E 06                  LD      A,6
863D ED 79                  OUT     (C),A           ;START se pozitioneaza LO
863F 05             ECO2:   DEC     B
8640 ED 78          ECO1:   IN      A,(C)           ;A aparut perforatia de sincroni-
8642 CB 07                  RLC     A               ;zare ?
8644 30 FA                  JR      NC,ECO1
8646 00                     NOP                     ;DA, temporizare
8647 00                     NOP
8648 00                     NOP
8649 00                     NOP
864A 05                     DEC     B
864B ED 78                  IN      A,(C)           ;Se citeste DATA
864D E6 7F                  AND     7FH             ;Se elimina bitul de paritate
864F 20 28                  JR      NZ,ECO3         ;Este caracter nul ?
8651 3A 862A                LD      A,(CAP)         ;DA
8654 B7                     OR      A               ;A fost citit un caracter nenul
8655 20 09                  JR      NZ,ECO4         ;in acest fisier ?
8657 04                     INC     B               ;NU, sintem la inceputul benzii
8658 ED 78          ECO5:   IN      A,(C)
865A CB 07                  RLC     A               ;Mai exista perforatia de sincro-
865C 30 FA                  JR      NC,ECO5         ;nizare ?
865E 18 E0                  JR      ECO1            ;NU
8660 04             ECO4:   INC     B               ;Ne aflam la sfirsitul benzii
8661 ED 78          ECO7:   IN      A,(C)
8663 CB 07                  RLC     A               ;Mai exista perforatia de sincro-
8665 38 FA                  JR      C,ECO7          ;nizare ?
8667 3E 07                  LD      A,7             ;NU
8669 06 0D                  LD      B,0DH
866B ED 79                  OUT     (C),A           ;STOP se pozitioneaza HI
866D 3E 06                  LD      A,6
866F ED 79                  OUT     (C),A           ;STOP se pozitioneaza LO
8671 AF                     XOR     A
8672 32 862A                LD      (CAP),A         ;Se initializeaza fanionul
8675 3E 1A                  LD      A,1AH           ;Se emite terminatorul de fisier
8677 C1                     POP     BC
8678 C9                     RET
8679 FE 1A          ECO3:   CP      1AH             ;Este terminator de fisier ?
867B 28 E3                  JR      Z,ECO4
```

```
867D  32 862A           LD    (CAP),A     ;NU, se pozitioneaza fanionul in-
8680  F5                PUSH  AF          ;dicind citirea unui caracter din
8681  04                INC   B           ;fisierul curent
8682  ED 78     ECO6:   IN    A,(C)
8684  CB 07             RLC   A           ;Mai exista perforatia de sincro-
8686  38 FA             JR    C,ECO6      ;nizare ?
8688  F1                POP   AF          ;NU, DATA citita o avem in regis-
8689  C1                POP   BC          ;trul A
868A  C9                RET
                ;
868B  84CB      RDSTAB: DW    SIOAST
868D  8705              DW    KBSTS
868F  84CB              DW    SIOAST
8691  84FB              DW    SIOBST
8693  84D6      LISTAB: DW    XMITA
8695  8836              DW    DISPO
8697  84A1              DW    LPT
8699  84FB              DW    XMITB
869B  84CB      STSTAB: DW    SIOAST
869D  8705              DW    KBSTS
869F  84CB              DW    SIOAST
86A1  84FB              DW    SIOBST
86A3  84EC      RDRTAB: DW    RECVA
86A5  862B              DW    CONSUL
86A7  86C3              DW    KBINP
86A9  84FB              DW    RECVB
86AB  84D6      PUNTAB: DW    XMITA
86AD  8600              DW    DT105S
86AF  8836              DW    DISPO
86B1  84FB              DW    XMITB
86B3  84EC      CITAB:  DW    RECVA
86B5  86C3              DW    KBINP
86B7  F5B4              DW    READER
86B9  84FB              DW    RECVB
86BB  84D6      COTAB:  DW    XMITA
86BD  8836              DW    DISPO
86BF  F5C0              DW    LIST
86C1  84FB              DW    XMITB

86C3  3A 94FE   KBINP:  LD    A,(CON4)
86C6  B7                OR    A
86C7  28 28             JR    Z,CIN1
86C9  3D                DEC   A
86CA  32 94FE           LD    (CON4),A
86CD  FE 04             CP    4
86CF  20 04             JR    NZ,CI1
86D1  3A FB78           LD    A,(LASTK)
86D4  C9                RET
86D5  FE 03     CI1:    CP    3
86D7  20 04             JR    NZ,CI2
86D9  3A 9505           LD    A,(XCM+1)
86DC  C9                RET
86DD  FE 02     CI2:    CP    2
86DF  20 04             JR    NZ,CI3
86E1  3A 9504           LD    A,(XCM)
86E4  C9                RET
86E5  FE 01     CI3:    CP    1
86E7  20 04             JR    NZ,CI4
86E9  3A 9507           LD    A,(YCM+1)
86EC  C9                RET
86ED  3A 9506   CI4:    LD    A,(YCM)
```

```
86F0  C9                  RET
86F1  FB          CIN1:   EI
86F2  FD 21 FB70          LD      IY,ZKST0
86F6  FD CB 0B 6E TFLAGS: BIT     5,(IY+0BH)      ;Asteapta apasarea unei taste
86FA  28 FA               JR      Z,TFLAGS
86FC  3A FB78             LD      A,(LASTK)       ;Preia tasta
86FF  FD CB 0B AE         RES     5,(IY+0BH)      ;Semnalizeaza ca a preluat tasta
8703  F3                  DI
8704  C9                  RET

8705  3A 94FE     KBSTS:  LD      A,(CON4)
8708  B7                  OR      A
8709  28 03               JR      Z,KBSTS2
870B  3E FF               LD      A,0FFH
870D  C9                  RET
870E  AF          KBSTS2: XOR     A
870F  FD 21 FB70          LD      IY,ZKST0
8713  FD CB 0B 6E          BIT     5,(IY+0BH)
8717  C8                  RET     Z
8718  3D                  DEC     A
8719  C9                  RET
                  ;*************************
                  ;* Scanarea tastaturii   *
                  ;*************************
                  ;
871A  2E 3F       L028E:  LD      L,03FH
871C  11 FFFF             LD      DE,0FFFFH
871F  01 FEFE             LD      BC,0FEFEH
8722  ED 78       L0296:  IN      A,(C)
8724  07                  RLCA
8725  07                  RLCA
8726  CB 2F               SRA     A
8728  07                  RLCA
8729  2F                  CPL
872A  E6 7F               AND     07FH
872C  28 0E               JR      Z,L02AB         ;salt pentru nici o tasta apasata
872E  67                  LD      H,A
872F  7D                  LD      A,L
8730  14          L029F:  INC     D
8731  C0                  RET     NZ
8732  D6 08       L02A1:  SUB     008H
8734  CB 3C               SRL     H
8736  30 FA               JR      NC,L02A1
8738  53                  LD      D,E
8739  5F                  LD      E,A
873A  20 F4               JR      NZ,L029F
873C  2D          L02AB:  DEC     L
873D  CB 00               RLC     B
873F  38 E1               JR      C,L0296
8741  7A                  LD      A,D
8742  3C                  INC     A
8743  C8                  RET     Z               ;revine pentru nici o tasta
                                                  ;apasata sau o tasta apasata
                  ;Analiza pentru doua taste actionate
8744  FE 38               CP      038H            ;una din cele 2 taste este CTRL?
8746  20 0A               JR      NZ,NUCTRL
8748  08          TESTE:  EX      AF,AF'
8749  3E 26               LD      A,026H
874B  BB                  CP      E
874C  38 02               JR      C,KEYSWR
874E  08                  EX      AF,AF'
```

```
874F  C9                  RET
8750  B7          KEYSWR: OR      A             ;inacceptabil 2 taste din blocul
8751  C9                  RET                   ;extins, revenire cu Z=0
8752  FE 28       NUCTRL: CP      028H          ;una din cele doua taste este CS?
8754  28 F2               JR      Z,TESTE
8756  FE 19               CP      019H
8758  C8                  RET     Z             ;revine pentru SS+ M sau N sau B
8759  7B                  LD      A,E
875A  5A                  LD      E,D
875B  57                  LD      D,A
875C  FE 18               CP      018H
875E  C9                  RET                   ;Z=1 daca s-a apasat si SS

875F  CD 871A       L02BF:  CALL    L028E
8762  C0                  RET     NZ            ;revenire pt caz inacceptabil
                    ;Se examineaza in continuare doua seturi de cite patru octeti
                    ;denumite KSTATE0-3 si KSTATE4-7 care permit detectia unei taste
                    ;noi apasate precum si lucrul in mod "repeat"
8763  21 FB70             LD      HL,ZKST0
8766  7E          L02C6:  LD      A,(HL)
8767  3C                  INC     A
8768  28 07               JR      Z,L02D1
876A  23                  INC     HL
876B  35                  DEC     (HL)
876C  2B                  DEC     HL
876D  20 02               JR      NZ,L02D1
876F  36 FF               LD      (HL),0FFH
8771  7D          L02D1:  LD      A,L
8772  21 FB74             LD      HL,KST4
8775  BD                  CP      L
8776  20 EE               JR      NZ,L02C6
8778  CD 87BE             CALL    L031E         ;pozitioneaza CY si aduce in A
877B  D0                  RET     NC            ;codul tastei daca este un caz
877C  21 FB70             LD      HL,ZKST0      ;acceptabil
877F  BE                  CP      (HL)
8780  28 2E               JR      Z,L0310       ;salt daca este o tasta in mod
8782  EB                  EX      DE,HL         ;"repeat"
8783  21 FB74             LD      HL,KST4
8786  BE                  CP      (HL)
8787  28 27               JR      Z,L0310       ;salt daca este o tasta in mod
8789  34                  INC     (HL)          ;"repeat"
878A  28 07               JR      Z,KNEW        ;salt pentru tasta noua
878C  35                  DEC     (HL)
878D  EB                  EX      DE,HL
878E  34                  INC     (HL)
878F  28 02               JR      Z,KNEW        ;salt pentru tasta noua
8791  35                  DEC     (HL)
8792  C9                  RET
8793  35          KNEW:   DEC     (HL)
8794  5F                  LD      E,A
8795  77                  LD      (HL),A
8796  23                  INC     HL
8797  36 05               LD      (HL),005H
8799  23                  INC     HL
879A  3A FB79             LD      A,(REPDEL)
879D  77                  LD      (HL),A
879E  23                  INC     HL
879F  FD 56 0B            LD      D,(IY+0BH)    ;D<--Flags
87A2  E5                  PUSH    HL
87A3  CD 87D7             CALL    L0333         ;subrutina de decodificare a
87A6  E1                  POP     HL            ;tastaturii
```

```
87A7  77                  LD    (HL),A
87A8  32 FB78       L0308: LD    (LASTK),A      ;depune la LASTK codul tastei
87AB  FD CB 0B EE         SET   5,(IY+0BH)     ;disponibile si semnalizeaza
87AF  C9                  RET                  ;tasta disponibila

                    ;Subrutina de "repeat"
                    ;
87B0  23            L0310: INC   HL
87B1  36 05               LD    (HL),005H
87B3  23                  INC   HL
87B4  35                  DEC   (HL)
87B5  C0                  RET   NZ
87B6  3A FB7A             LD    A,(REPPER)
87B9  77                  LD    (HL),A
87BA  23                  INC   HL
87BB  7E                  LD    A,(HL)
87BC  18 EA               JR    L0308

                    ;Subrutina L031E elimina cazurile: nici o cheie apasata, CTRL+SS
                    ;sau apasarea singulara a uneia din tastele CTRL, CS, SS. La
                    ;revenire in A se gaseste codul tastei extras din tabelul TA
                    ;
87BE  42            L031E: LD    B,D
87BF  16 00               LD    D,000H
87C1  7B                  LD    A,E
87C2  FE 37               CP    037H
87C4  D0                  RET   NC             ;cheie neapasata, CTRL singular
87C5  FE 27               CP    027H           ;=> CY=0
87C7  C8                  RET   Z              ;CS singular => CY=0
87C8  FE 18               CP    018H
87CA  20 04               JR    NZ,L032C
87CC  78                  LD    A,B
87CD  FE 28               CP    028H
87CF  D0                  RET   NC             ;SS singular sau CTRL+SS => CY=0
87D0  21 FC21       L032C: LD    HL,TA
87D3  19                  ADD   HL,DE
87D4  7E                  LD    A,(HL)         ;A <- codul extras din tabelul TA
87D5  37                  SCF                  ;TA=Main code table
87D6  C9                  RET
                    ;Intrari: E=Main code(extras din tabelul TA)
                    ;         D=Flags
                    ;         B=Shift byte: 18H pentru SS
                    ;                       27H pentru CS
                    ;                       37H pentru CTRL
                    ;                       FFH in rest
                    ;
87D7  7B            L0333: LD    A,E
87D8  FE 7F               CP    07FH ;DEL?(7FH), CS+SS?(80H), CSL?(81H), F0H-F7H?
87DA  38 09               JR    C,CP3A
87DC  FE F0               CP    0F0H           ;tasta programabila?
87DE  D8                  RET   C              ;revine pentru DEL, CS+SS, CSL
87DF  CB 40               BIT   0,B            ;SS+tasta programabila (tp)?
87E1  C0                  RET   NZ             ;revine daca nu s-a apasat SS+tp
87E2  C6 04               ADD   A,004H         ;cod F4H...F7H
87E4  C9                  RET
87E5  FE 3A         CP3A: CP    03AH
87E7  38 26               JR    C,L0367        ;salt daca s-a apasat o cifra
87E9  21 FC18       L034F: LD    HL,TE-041H
87EC  CB 40               BIT   0,B
87EE  28 1A               JR    Z,L034A        ;salt pentru litera +SS
87F0  08                  EX    AF,AF'
```

```
87F1  3E 37              LD    A,037H
87F3  B8                 CP    B              ;litera+CTRL?
87F4  20 04              JR    NZ,NCTRL
87F6  08                 EX    AF,AF'
87F7  D6 40              SUB   040H           ;litera+CTRL ==>01H...1AH
87F9  C9                 RET

87FA  08           NCTRL: EX    AF,AF'
87FB  FD CB 0C 5E        BIT   3,(IY+0CH)     ;Caps Lock setat?
87FF  20 05              JR    NZ,CSLOCK
8801  04                 INC   B
8802  C0                 RET   NZ
8803  C6 20        A20:  ADD   A,020H         ;litere mari ->litere mici
8805  C9                 RET

8806  04           CSLOCK: INC   B
8807  C8                 RET   Z
8808  18 F9              JR    A20

880A  16 00        L034A: LD    D,000H
880C  19                 ADD   HL,DE  ;formeaza adresa din tabelul referit de H
880D  7E                 LD    A,(HL) ;aduce codul tastei
880E  C9                 RET

880F  FE 30        L0367: CP    030H
8811  D8                 RET   C      ;revine pentru BS, TAB, LF, CR, ESC, SPACE
8812  04                 INC   B      ;B=00 =no SHIFT, 19H=SS, 28H=CS, 38H=CTRL
8813  C8                 RET   Z      ;revine pentru cifra fara Shift
8814  CB 40              BIT   0,B
8816  20 0F              JR    NZ,SSDGT
8818  CB 60              BIT   4,B
881A  28 09              JR    Z,INVAL        ;nu se accepta CS+cifra
881C  D6 16        CTRDGT: SUB   016H         ;CTRL+1...5=#1B...#1F
881E  FE 1A              CP    01AH
8820  28 03              JR    Z,INVAL
8822  FE 20              CP    020H
8824  D8                 RET   C
8825  AF           INVAL: XOR   A              ;CY=0 pentru invalidare
8826  C9                 RET
8827  D6 10        SSDGT: SUB   010H          ;30H...39H -->20H..,29H
8829  FE 22              CP    022H           ;salt pt SS+2 (coada de maimuta)
882B  28 06              JR    Z,L03B2
882D  FE 20              CP    020H
882F  C0                 RET   NZ
8830  3E 5F              LD    A,05FH
8832  C9                 RET
8833  3E 40        L03B2: LD    A,040H
8835  C9                 RET

8836  F3           DISP0: DI
8837  CD 89A3            CALL  D17
883A  CD 91AF      D15:  CALL  SALV
883D  3A FB91            LD    A,(ESC)
8840  B7                 OR    A
8841  C2 8919            JP    NZ,EXTEND
8844  3A 91D5      D15P4: LD    A,(ML)
8847  CB 47              BIT   0,A
8849  C2 A039            JP    NZ,CDSPEC
884C  CB 57              BIT   2,A
884E  C2 A0DB            JP    NZ,GR
8851  79                 LD    A,C
```

```
8852  E6 7F              AND   07FH
8854  FE 20              CP    020H
8856  38 36              JR    C,COM
8858  FE 7F              CP    07FH
885A  CA 8987            JP    Z,RETCO
885D  CD 8B42            CALL  PICT
8860  3A FB8F            LD    A,(X)
8863  3C                 INC   A
8864  32 FB8F            LD    (X),A
8867  FE 50        AD1:  CP    050H
8869  20 1D              JR    NZ,TIPCUR
886B  AF                 XOR   A
886C  32 FB8F            LD    (X),A
886F  3A FB8E      INCRY: LD    A,(Y)       ;revenire in coloana 0 (prima)
8872  3C                 INC   A             ;a rindului urmator, al carui
8873  32 FB8E            LD    (Y),A       ;numar nu poate fi mai mare ca
8876  FE 18              CP    CARR        ;valoarea maxima:
8878  20 04              JR    NZ,NUY0     ;       - 23 pentru ecran mic
887A  AF                 XOR   A           ;       - 35 pentru ecran mare
887B  32 FB8E            LD    (Y),A       ;revenire in rindul 0 daca este
887E  47           NUY0: LD    B,A         ;necesar
887F  3A FB90            LD    A,(ROLL)    ;este nevoie de SCROLL?
8882  B8                 CP    B
8883  20 03              JR    NZ,TIPCUR   ;nu, tipareste cursor
8885  CD 8B0C            CALL  SCROLL      ;da, SCROLL
8888  CD 8AED    TIPCUR: CALL  YCURS       ;memoreaza la ZCAR caracterul
888B  C3 8987              JP    RETCO      ;curent pentru eventuala refacere
                                            ;ulterioara cu NCURS salt pentru
                                            ;revenirea din subrutina
888E  FE 1D        COM:  CP    1DH
8890  20 0D              JR    NZ,COM1
8892  21 91D8            LD    HL,INDIC
8895  36 00              LD    (HL),0
8897  21 91D5            LD    HL,ML
889A  CB D6              SET   2,(HL)
889C  C3 8987            JP    RETCO
889F  FE 0D        COM1: CP    00DH        ;CR?
88A1  20 09              JR    NZ,CP0A
88A3  CD 8AFC      CR:   CALL  NCURS       ;reface caracterul de la pozitia
88A6  AF                 XOR   A           ;curenta
88A7  32 FB8F            LD    (X),A       ;cursorul pe prima pozitie a
88AA  18 DC              JR    TIPCUR      ;liniei curente
                                            ;revenire indirecta

88AC  FE 0A        CP0A: CP    00AH        ;LF?
88AE  20 05              JR    NZ,CP07
88B0  CD 8AFC            CALL  NCURS       ;reface caracterul curent
88B3  18 BA              JR    INCRY       ;revenire prin mutarea cursorului
                                            ;pe linia urmatoare
88B5  FE 07        CP07: CP    007H        ;BELL?
88B7  20 0C              JR    NZ,CP1B
88B9  11 00FF            LD    DE,00FFH    ;initializare parametri
88BC  21 00C8            LD    HL,00C8H    ;subrutina sunet
88BF  CD 8F35            CALL  BEEP        ;apel subrutina sunet
88C2  C3 8987            JP    RETCO

88C5  FE 1B        CP1B: CP    01BH        ;ESC?
88C7  20 08              JR    NZ,CP09
88C9  3E 01        CP1BP2: LD  A,001H
88CB  32 FB91            LD    (ESC),A     ;initializeaza secventa ESCAPE
88CE  C3 8987            JP    RETCO
```

```
88D1  FE 09       CP09:  CP    009H
88D3  20 16              JR    NZ,CP08      ;TAB?
88D5  CD 8AFC            CALL  NCURS        ;reface carac. din pozitia
88D8  3A FB8F            LD    A,(X)        ;curenta
88DB  C6 08              ADD   A,008H       ;incrementeaza cu 8 pozitia
88DD  32 FB8F            LD    (X),A        ;de coloana, verificind sa nu
88E0  FE 50       AD2:   CP    050H         ;treaca de 79
88E2  38 A4              JR    C,TIPCUR
88E4  3E 4F       AD3:   LD    A,04FH
88E6  32 FB8F            LD    (X),A
88E9  18 9D              JR    TIPCUR

88EB  FE 08       CP08:  CP    008H         ;Back Space?
88ED  C2 8987            JP    NZ,RETCO     ;nu, salt
88F0  3A FB8F            LD    A,(X)        ;cursorul pe coloana 0?
88F3  B7                 OR    A
88F4  C2 8A36            JP    NZ,BSNORM    ;nu => secventa normala de
88F7  3A FB90            LD    A,(ROLL)     ;Back Space
88FA  47                 LD    B,A          ;daca cursorul este in pozitia
88FB  3A FB8E            LD    A,(Y)        ;HOME nu se intreprinde nimic
88FE  B8                 CP    B            ;altfel se muta cursorul pe
88FF  CA 8987            JP    Z,RETCO      ;ultima pozitie a rindului
8902  CD 8AFC            CALL  NCURS        ;precedent
8905  3A FB8E            LD    A,(Y)
8908  3D                 DEC   A
8909  F2 890E            JP    P,YAY
890C  3E 17              LD    A,CARR-1
890E  32 FB8E     YAY:   LD    (Y),A
8911  3E 4F       AD4:   LD    A,04FH
8913  32 FB8F            LD    (X),A
8916  C3 8888            JP    TIPCUR

8919  21 91D5   EXTEND: LD    HL,ML
891C  79                 LD    A,C
891D  FE 0C              CP    0CH
891F  CA 93D4            JP    Z,ESCFF      ;Sterge ecranul
8922  FE 1A              CP    1AH
8924  CA 9508            JP    Z,ESCSUB     ;Introducere grafica
8927  FE 17              CP    17H
8929  CA 9711            JP    Z,ESCETB     ;Copie la imprimanta
892C  FE 06              CP    06H
892E  CA 98B0            JP    Z,ESCACK     ;Programare interfete
8931  FE 03              CP    03H
8933  CA A031            JP    Z,ESCETX     ;Comenzi speciale
8936  FE 18              CP    18H          ;18H => anulare secventa  ESCAPE
8938  CA 8980            JP    Z,STESC
893B  3A FB92            LD    A,(POZCUR)
893E  B7                 OR    A            ;test pentru pozitie cursor = 0
893F  CA 89C6            JP    Z,TESTC      ;este => secventa de pozitionare
8942  FE 01              CP    001H         ;cursor
8944  20 28              JR    NZ,PC2
8946  79                 LD    A,C
8947  FE 20              CP    020H         ;test pentru valoarea corecta a
8949  38 35              JR    C,STESC      ;argumumentului de pozitionare
894B  FE 38              CP    CARR+20H     ;cursor pe o linie
894D  30 31              JR    NC,STESC
894F  D6 20              SUB   020H         ;Modificare (Y) in functie de
8951  4F                 LD    C,A          ;argumument si starea variabilei
8952  3A FB90            LD    A,(ROLL)     ;(ROLL), ce reflecta starea de
8955  81                 ADD   A,C          ;SCROLL a ecranului
```

```
8956  FE 18          CP    CARR
8958  38 02          JR    C,OK
895A  D6 18          SUB   CARR
895C  F5       OK:   PUSH  AF
895D  CD 8AFC        CALL  NCURS
8960  F1             POP   AF
8961  32 FB8E        LD    (Y),A
8964  3E 02    A2:   LD    A,002H      ;(POZCUR)=2 pentru primirea celui
8966  32 FB92        LD    (POZCUR),A  ;de-al doilea argument de
8969  CD 8AED        CALL  YCURS       ;pozitionare cursor
896C  18 19          JR    RETCO       ;salt la revenire

896E  79       PC2:  LD    A,C
896F  D6 20          SUB   020H
8971  FE 50    AD5:  CP    050H
8973  30 0B          JR    NC,STESC    ;pentru numar de coloana >=80 se
8975  F5             PUSH  AF          ;iese din ESCAPE
8976  CD 8AFC        CALL  NCURS
8979  F1             POP   AF
897A  32 FB8F        LD    (X),A
897D  CD 8AED        CALL  YCURS
8980  AF       STESC: XOR   A           ;iesire din secventa ESCAPE
8981  32 FB91        LD    (ESC),A
8984  32 FB92        LD    (POZCUR),A
8987  CD 91C2  RETCO: CALL  REF         ;refacere context microprocesor
898A  F5       D18:  PUSH  AF
898B  C5             PUSH  BC
898C  01 7FFD        LD    BC,7FFDH
898F  3A FB87        LD    A,(P7FFD)   ;inchide RAM VIDEO
8992  E6 FB          AND   0FBH
8994  32 FB87        LD    (P7FFD),A
8997  ED 79          OUT   (C),A
8999  06 0C          LD    B,0CH
899B  ED 78          IN    A,(C)
899D  E6 F9          AND   0F9H
899F  F6 01          OR    1
89A1  18 1D          JR    D171
89A3  F5       D17:  PUSH  AF
89A4  3A 0003        LD    A,(3)
89A7  32 89C5        LD    (DIOBYT),A
89AA  C5             PUSH  BC
89AB  01 0CFD        LD    BC,0CFDH
89AE  ED 78          IN    A,(C)
89B0  F6 06          OR    6
89B2  E6 FE          AND   0FEH
89B4  ED 79          OUT   (C),A
89B6  06 7F          LD    B,7FH
89B8  3A FB87        LD    A,(P7FFD)
89BB  F6 04          OR    4
89BD  32 FB87        LD    (P7FFD),A
89C0  ED 79    D171: OUT   (C),A
89C2  C1             POP   BC
89C3  F1             POP   AF
89C4  C9             RET
89C5           DIOBYT: DS  1
               ;
89C6  79       TESTC: LD   A,C
89C7  E6 5F          AND   05FH
89C9  FE 41          CP    "A"
89CB  20 1C          JR    NZ,CPB
89CD  3A FB90        LD    A,(ROLL)    ;tratare ESC A (cursorul se
```

```
89D0  47                    LD    B,A          ;deplaseaza cu o linie in sus)
89D1  3A FB8E               LD    A,(Y)
89D4  B8                    CP    B
89D5  CA 8980               JP    Z,STESC
89D8  CD 8AFC               CALL  NCURS
89DB  3A FB8E               LD    A,(Y)
89DE  3D                    DEC   A
89DF  F2 89E4               JP    P,CSUS
89E2  3E 17                 LD    A,CARR-1
89E4  32 FB8E         CSUS: LD    (Y),A
89E7  18 3C                 JR    DEPLC
                      ;
89E9  FE 42           CPB:  CP    "B"
89EB  20 22                 JR    NZ,CPC
89ED  3A FB90               LD    A,(RGLL)
89F0  3D                    DEC   A
89F1  F2 89F6               JP    P,CJOS
89F4  3E 17                 LD    A,CARR-1
89F6  47              CJOS: LD    B,A          ;ESC B <=) cursor in jos
89F7  3A FB8E               LD    A,(Y)
89FA  B8                    CP    B
89FB  CA 8980               JP    Z,STESC
89FE  CD 8AFC               CALL  NCURS
8A01  3A FB8E               LD    A,(Y)
8A04  3C                    INC   A
8A05  FE 18                 CP    CARR
8A07  20 01                 JR    NZ,AINY
8A09  AF                    XOR   A
8A0A  32 FB8E         AINY: LD    (Y),A
8A0D  18 16                 JR    DEPLC

8A0F  FE 43           CPC:  CP    "C"
8A11  20 18                 JR    NZ,CPD       ;ESC C <=) cursor la dreapta
8A13  3A FB8F               LD    A,(X)
8A16  FE 4F           AD6:  CP    04FH
8A18  CA 8980               JP    Z,STESC
8A1B  CD 8AFC               CALL  NCURS
8A1E  3A FB8F               LD    A,(X)
8A21  3C                    INC   A
8A22  32 FB8F               LD    (X),A
8A25  CD 8AED         DEPLC: CALL YCURS
8A28  C3 8980               JP    STESC

8A2B  FE 44           CPD:  CP    "D"
8A2D  20 13                 JR    NZ,CPE       ;ESC D <=) cursor la stinga
8A2F  3A FB8F               LD    A,(X)
8A32  B7                    OR    A
8A33  CA 8980               JP    Z,STESC
8A36  CD 8AFC         BSNORM: CALL NCURS
8A39  3A FB8F               LD    A,(X)
8A3C  3D                    DEC   A
8A3D  32 FB8F               LD    (X),A
8A40  18 E3                 JR    DEPLC

8A42  FE 45           CPE:  CP    "E"
8A44  20 05                 JR    NZ,CPH       ;ESC E <=) sterge ecran
8A46  CD 9462               CALL  STEREC
8A49  18 DA                 JR    DEPLC

8A4B  FE 48           CPH:  CP    "H"
8A4D  20 0F                 JR    NZ,CPK
```

```
8A4F  CD 8AFC            CALL  NCURS      ;ESC H <=> cursor in coltul
8A52  AF                 XOR   A          ;din stinga sus
8A53  32 FB8F            LD    (X),A
8A56  3A FB90            LD    A,(ROLL)
8A59  32 FB8E            LD    (Y),A
8A5C  18 C7              JR    DEPLC

8A5E  FE 4B       CPK:   CP    "K"
8A60  20 1E              JR    NZ,CPJ
8A62  CD 8AFC     BETA:  CALL  NCURS      ;ESC K sterge caracterele de
8A65  3A FB8F            LD    A,(X)      ;la cursor pina la sfirsitul
8A68  F5                 PUSH  AF         ;liniei curente
8A69  FE 50       CP80:  CP    050H
8A6B  28 0D              JR    Z,OLDX
8A6D  F5                 PUSH  AF
8A6E  3E 20              LD    A," "
8A70  CD 8B42            CALL  PICT
8A73  F1                 POP   AF
8A74  3C                 INC   A
8A75  32 FB8F            LD    (X),A
8A78  18 EF              JR    CP80

8A7A  F1          OLDX:  POP   AF
8A7B  32 FB8F            LD    (X),A
8A7E  18 A5              JR    DEPLC

8A80  FE 4A       CPJ:   CP    "J"
8A82  20 49              JR    NZ,CPN
8A84  CD 8AFC            CALL  NCURS      ;ESC J sterge ecranul de la
8A87  3A FB8F            LD    A,(X)      ;cursor pina la sfirsitul
8A8A  F5                 PUSH  AF         ;ecranului
8A8B  FE 50       ESCJ80: CP   050H       ;(cursorul nu se deplaseaza)
8A8D  28 0D              JR    Z,OLDXJ
8A8F  F5                 PUSH  AF
8A90  3E 20              LD    A," "
8A92  CD 8B42            CALL  PICT
8A95  F1                 POP   AF
8A96  3C                 INC   A
8A97  32 FB8F            LD    (X),A
8A9A  18 EF              JR    ESCJ80

8A9C  F1          OLDXJ: POP   AF
8A9D  32 FB8F            LD    (X),A
8AA0  3A FB8E            LD    A,(Y)
8AA3  F5                 PUSH  AF
8AA4  3A FB90            LD    A,(ROLL)
8AA7  3D                 DEC   A
8AA8  F2 8AAD            JP    P,BA
8AAB  3E 17              LD    A,CARR-1
8AAD  47          BA:    LD    B,A
8AAE  C5                 PUSH  BC
8AAF  3A FB8E     YA:    LD    A,(Y)
8AB2  C1                 POP   BC
8AB3  B8                 CP    B
8AB4  C5                 PUSH  BC
8AB5  28 0E              JR    Z,REFBC
8AB7  3C                 INC   A
8AB8  FE 18              CP    CARR
8ABA  20 01              JR    NZ,AY
8ABC  AF                 XOR   A
8ABD  32 FB8E     AY:    LD    (Y),A
```

```
8AC0  CD 8BF4           CALL   STERLI
8AC3  18 EA             JR     YA

8AC5  C1          REFBC: POP    BC
8AC6  F1                 POP    AF
8AC7  32 FB8E            LD     (Y),A
8ACA  C3 8A25            JP     DEPLC

8ACD  FE 4E       CPN:   CP     "N"
8ACF  20 08              JR     NZ,CPO
8AD1  3E FF              LD     A,0FFH        ;ESC N <=> trece in VIDEO
8AD3  32 FB93     VIDEOA: LD    (VIDEO),A     ;INVERS
8AD6  C3 8980            JP     STESC

8AD9  FE 4F       CPO:   CP     "O"
8ADB  20 03              JR     NZ,CPY        ;ESC O <=> VIDEO NORMAL
8ADD  AF                 XOR    A
8ADE  18 F3              JR     VIDEOA

8AE0  FE 59       CPY:   CP     "Y"
8AE2  C2 8980            JP     NZ,STESC
8AE5  3E 01              LD     A,001H        ;ESC Y <=> 1 -> (POZCUR)
8AE7  32 FB92            LD     (POZCUR),A    ;<=>validareare pozitionare
8AEA  C3 8987            JP     RETCO         ;cursor

8AED              YCURS:
8AED  CD 8B60     MEMCAR: CALL  FHL           ;f(X,Y) -> HL
8AF0  11 FB94             LD    DE,ZCAR
8AF3  06 08               LD    B,008H
8AF5  7E          MEMZC:  LD    A,(HL)        ;memoreaza la ZCAR caracterul
8AF6  12                  LD    (DE),A        ;referit de (X,Y)
8AF7  24                  INC   H
8AF8  13                  INC   DE
8AF9  10 FA               DJNZ  MEMZC
8AFB  C9                  RET

8AFC  F3          NCURS:  DI
8AFD  CD 8B60     REFCAR: CALL  FHL           ;reface, la adresa referita de
8B00  11 FB94             LD    DE,ZCAR       ;(X,Y) caracterul de la ZCAR
8B03  06 08               LD    B,008H
8B05  1A          ZCMEM:  LD    A,(DE)
8B06  77                  LD    (HL),A
8B07  24                  INC   H
8B08  13                  INC   DE
8B09  10 FA               DJNZ  ZCMEM
8B0B  C9                  RET

8B0C  3A 91D6     SCROLL: LD    A,(NCR)       ;SCROLL ecran
8B0F  FE 00               CP    0
8B11  C2 929C             JP    NZ,SCROLI
8B14  3A FB8E             LD    A,(Y)
8B17  3C                  INC   A
8B18  FE 18               CP    CARR
8B1A  20 01               JR    NZ,SARI1
8B1C  AF                  XOR   A
8B1D  CD 8B24     SARI1:  CALL  SARI
8B20  CD 8BF4             CALL  STERLI
8B23  C9                  RET
8B24  32 FB90     SARI:   LD    (ROLL),A
8B27  CB 27               SLA   A
8B29  CB 27               SLA   A
```

```
8B2B  CB 27                   SLA   A
8B2D  01 5CFD                 LD    BC,5CFDH
8B30  ED 79                   OUT   (C),A
8B32  01 0EFD                 LD    BC,0EFDH
8B35  ED 78                   IN    A,(C)
8B37  38 04                   JR    C,S2
8B39  E6 EF                   AND   0EFH
8B3B  18 02                   JR    S3
8B3D  F6 10             S2:   OR    10H
8B3F  ED 79             S3:   OUT   (C),A
8B41  C9                      RET

8B42  01 8B35           PICT: LD    BC,BASE1-100H  ;PICteaza caracterul al carui cod
8B45  26 00                   LD    H,000H         ;este primit in registrul A
8B47  6F                      LD    L,A
8B48  29                      ADD   HL,HL
8B49  29                      ADD   HL,HL
8B4A  29                      ADD   HL,HL
8B4B  09                      ADD   HL,BC
8B4C  EB                      EX    DE,HL ;DE-adresa matricii caracterului de generat
8B4D  D5                      PUSH  DE    ;in generatorul de caractere
8B4E  CD 8B60                 CALL  FHL   ;HL-adresa din memoria video unde se
8B51  D1                      POP   DE    ;PICTeaza
8B52  06 08                   LD    B,008H;octeti
8B54  1A                DRAW: LD    A,(DE)
8B55  4F                      LD    C,A
8B56  3A FB93                 LD    A,(VIDEO)      ;pictarea se face in acord cu
8B59  A9                      XOR   C              ;optiunea de afisare pe video
8B5A  77                      LD    (HL),A         ;(VIDEO DIRECT sau INVERS)
8B5B  13                      INC   DE
8B5C  24                      INC   H
8B5D  10 F5                   DJNZ  DRAW
8B5F  C9                      RET
8B60  ED 5B FB8E        FHL:  LD    DE,(Y)         ;in functie de pozitia cursorului
8B64  CB 3A             FHLP1: SRL  D              ;(X,Y) se deschide pagina video
8B66  D4 8BD0                 CALL  NC,PAR         ;corespunzatoare si se
8B69  DC 8BC6                 CALL  C,IMPAR        ;calculeaza adresa
8B6C  D5                GHL:  PUSH  DE             ;de depunere in memoria video a
8B6D  7B                      LD    A,E            ;primului octet din cei opt care
8B6E  D6 18                   SUB   24             ;apartin matricii formei
8B70  38 07                   JR    C,GH1          ;caracterului
8B72  FE 08                   CP    8
8B74  38 02                   JR    C,GH2
8B76  D6 08                   SUB   8
8B78  5F                GH2:  LD    E,A
8B79  CD 8BB6           GH1:  CALL  ABCY
8B7C  CD 8BA3                 CALL  GHLP1
8B7F  D1                      POP   DE
8B80  7B                      LD    A,E
8B81  FE 18                   CP    24
8B83  30 03                   JR    NC,GH3
8B85  C3 91E2                 JP    ZONA1
8B88  FE 20             GH3:  CP    32
8B8A  D2 91F8                 JP    NC,ZONA3
8B8D  CD 91EC                 CALL  ZONA2
8B90  7A                      LD    A,D
8B91  FE 08                   CP    8
8B93  30 09                   JR    NC,GH5
8B95  7C                      LD    A,H
8B96  E6 EF                   AND   0EFH
8B98  67                      LD    H,A
```

```
8B99  7D              LD    A,L
8B9A  C6 18           ADD   A,18H
8B9C  6F              LD    L,A          ;adresa este returnata in HL
8B9D  C9              RET
8B9E  7C        GH5:  LD    A,H
8B9F  C6 18           ADD   A,18H
8BA1  67              LD    H,A
8BA2  C9              RET
8BA3  30 08     GHLP1:JR    NC,ZB
8BA5  B0              OR    B
8BA6  26 58           LD    H,058H
8BA8  6F              LD    L,A
8BA9  7A              LD    A,D
8BAA  B5              OR    L
8BAB  6F        LA:   LD    L,A
8BAC  C9              RET
8BAD  F6 40     ZB:   OR    040H
8BAF  67              LD    H,A
8BB0  7A              LD    A,D
8BB1  D6 08           SUB   008H
8BB3  B0              OR    B
8BB4  18 F5           JR    LA

8BB6  7B        ABCY: LD    A,E
8BB7  0F              RRCA
8BB8  0F              RRCA
8BB9  0F              RRCA
8BBA  E6 E0           AND   0E0H
8BBC  47              LD    B,A
8BBD  7B              LD    A,E
8BBE  E6 18           AND   018H
8BC0  4F              LD    C,A
8BC1  7A              LD    A,D
8BC2  FE 08           CP    008H
8BC4  79              LD    A,C
8BC5  C9              RET

8BC6  01 7FFD   IMPAR:LD    BC,7FFDH     ;deschide, in zona de adrese
8BC9  3A FB87         LD    A,(P7FFD)    ;#4000-#5FFF a memoriei video
8BCC  F6 08           OR    008H         ;pagina video impara pentru
8BCE  18 08           JR    PAR00        ;coloanele 1, 3,... 79

8BD0  01 7FFD   PAR:  LD    BC,7FFDH  ;deschide, in zona de adrese
8BD3  3A FB87         LD    A,(P7FFD) ;#4000-#5FFF a memoriei video pagina
8BD6  E6 F7           AND   0F7H ;pagina video para pentru coloanele 0, 2,
8BD8  32 FB87   PAR00:LD    (P7FFD),A ;4,... 78
8BDB  ED 79           OUT   (C),A
8BDD  C9              RET

8BDE  3E 08     STER: LD    A,008H
8BE0  E5        SALHL:PUSH  HL
8BE1  54              LD    D,H
8BE2  5D              LD    E,L
8BE3  13              INC   DE
8BE4  F5              PUSH  AF
8BE5  3A FB93         LD    A,(VIDEO)
8BE8  77              LD    (HL),A
8BE9  F1              POP   AF
8BEA  C5              PUSH  BC
8BEB  ED B0           LDIR
8BED  C1              POP   BC
```

```
8BEE  E1                POP   HL
8BEF  24                INC   H
8BF0  3D                DEC   A
8BF1  20 ED             JR    NZ,SALHL
8BF3  C9                RET

8BF4  01 0007           STERLI: LD    BC,0007H
8BF7  CD 8BFD                   CALL  LIZOST
8BFA  01 001F                   LD    BC,001FH
8BFD  ED 5B FB8E        LIZOST: LD    DE,(Y)
8C01  CB 40                     BIT   0,B
8C03  20 03                     JR    NZ,LI1
8C05  C5                        PUSH  BC
8C06  18 11                     JR    LIZOP2
8C08  CB 80             LI1:    RES   0,B
8C0A  CB 48                     BIT   1,B
8C0C  20 04                     JR    NZ,LI2
8C0E  16 00                     LD    D,0
8C10  18 04                     JR    LI3
8C12  CB 88             LI2:    RES   1,B
8C14  16 08                     LD    D,8
8C16  C5                LI3:    PUSH  BC
8C17  18 09                     JR    JR016
8C19  51                LIZOP2: LD    D,C
8C1A  CB 62                     BIT   4,D
8C1C  16 08                     LD    D,008H
8C1E  20 02                     JR    NZ,JR016
8C20  16 00                     LD    D,000H
8C22  CD 8BD0           JR016:  CALL  PAR
8C25  CD 8B6C                   CALL  GHL
8C28  C1                        POP   BC
8C29  E5                        PUSH  HL
8C2A  CD 8BDE                   CALL  STER
8C2D  E1                        POP   HL
8C2E  C5                        PUSH  BC
8C2F  CD 8BC6                   CALL  IMPAR
8C32  C1                        POP   BC
8C33  18 A9                     JR    STER

                        ;generatorul de caractere
                        ;
8C35  00 00 00 00       BASE1:        DB    000H,000H,000H,000H,000H,000H,000H,000H
8C39  00 00 00
8C3C  00 30 78 78             DB    000H,'0xx00',000H,'0',000H,'11'
8C40  30 30 00 30
8C44  00 6C 6C
8C47  6C 00 00 00             DB    '1',000H,000H,000H,000H,000H,'1'
8C4B  00 00 6C
8C4E  6C FE 6C FE             DB    '1',0FEH,'1',0FEH,'11',000H,'0'
8C52  6C 6C 00 30
8C56  7C C0 78 0C             DB    '¦',0C0H,'x',00CH,0F8H,'0',000H
8C5A  F8 30 00
8C5D  00 C6 CC 18             DB    000H,0C6H,0CCH,018H,'0f',0C6H,000H
8C61  30 66 C6 00
8C65  38 6C 38 76             DB    '818v',0DCH,0CCH,'v',000H,'```',0C0H
8C69  DC CC 76 00
8C6D  60 60 C0
8C70  00 00 00 00             DB    000H,000H,000H,000H,000H,018H,'0'
8C74  00 18 30
8C77  60 60 60 30             DB    '```0',018H,000H,'`0',018H,018H
8C7B  18 00 60 30
```

```
8C7F  18 18
8C81  18 30 60 00                DB    018H,'0`',000H,000H,'f<',0FFH,'<'
8C85  00 66 3C FF
8C89  3C
8C8A  66 00 00 00                DB    'f',000H,000H,000H,'00',0FCH,'0'
8C8E  30 30 FC 30
8C92  30 00 00 00                DB    '0',000H,000H,000H,000H,000H,000H
8C96  00 00 00
8C99  00 30 30 60                DB    000H,'00`',000H,000H,000H,0FCH
8C9D  00 00 00 FC
8CA1  00 00 00 00                DB    000H,000H,000H,000H,000H,000H,000H
8CA5  00 00 00
8CA8  00 00 30 30                DB    000H,000H,'00',000H,006H,00CH,018H
8CAC  00 06 0C 18
8CB0  30 60 C0 80                DB    '0`',0C0H,080H,000H,'!',0C6H,0CEH
8CB4  00 7C C6 CE
8CB8  DE F6 E6 7C                DB    0DEH,0F6H,0E6H,'!',000H,'0p000'
8CBC  00 30 70 30
8CC0  30 30
8CC2  30 FC 00 78                DB    '0',0FCH,000H,'x',0CCH,00CH,'8'
8CC6  CC 0C 38
8CC9  60 CC FC 00                DB    '`',0CCH,0FCH,000H,'x',0CCH,00CH
8CCD  78 CC 0C
8CD0  38 0C CC 78                DB    '8',00CH,0CCH,'x',000H,01CH,'<'
8CD4  00 1C 3C
8CD7  6C CC FE 0C                DB    'l',0CCH,0FEH,00CH,01EH,000H,0FCH
8CDB  1E 00 FC
8CDE  C0 F8 0C 0C                DB    0C0H,0F8H,00CH,00CH,0CCH,'x',000H
8CE2  CC 78 00
8CE5  38 60 C0 F8                DB    '8`',0C0H,0F8H,0CCH,0CCH,'x',000H
8CE9  CC CC 78 00
8CED  FC CC 0C 18                DB    0FCH,0CCH,00CH,018H,'000',000H
8CF1  30 30 30 00
8CF5  78 CC CC 78                DB    'x',0CCH,0CCH,'x',0CCH,0CCH,'x'
8CF9  CC CC 78
8CFC  00 78 CC CC                DB    000H,'x',0CCH,0CCH,'!',00CH,018H
8D00  7C 0C 18
8D03  70 00 00 30                DB    'p',000H,000H,'00',000H,000H,'0'
8D07  30 00 00 30
8D0B  30 00 00 30                DB    '0',000H,000H,'00',000H,000H,'0'
8D0F  30 00 00 30
8D13  30 60 18 30                DB    '0`',018H,'0`',0C0H,'`0',018H,000H
8D17  60 C0 60 30
8D1B  18 00
8D1D  00 00 FC 00                DB    000H,000H,0FCH,000H,000H,0FCH,000H
8D21  00 FC 00
8D24  00 60 30 18                DB    000H,'`0',018H,00CH,018H,'0`',000H
8D28  0C 18 30 60
8D2C  00
8D2D  78 CC 0C 18                DB    'x',0CCH,00CH,018H,'0',000H,'0'
8D31  30 00 30
8D34  00 7C C6 DE                DB    000H,'!',0C6H,0DEH,0DEH,0DEH,0C0H
8D38  DE DE C0
8D3B  78 00 30 78                DB    'x',000H,'0x',0CCH,0CCH,0FCH,0CCH
8D3F  CC CC FC CC
8D43  CC 00 FC 66                DB    0CCH,000H,0FCH,'ff!ff',0FCH,000H
8D47  66 7C 66 66
8D4B  FC 00
8D4D  3C 66 C0 C0                DB    '<f',0C0H,0C0H,0C0H,'f<',000H,0F8H
8D51  C0 66 3C 00
8D55  F8
```

```
8D56  6C 66 66 66                DB    'lfffl',0F8H,000H,0FEH,'bhxhb',0FEH
8D5A  6C F8 00 FE
8D5E  62 68 78 68
8D62  62 FE
8D64  00 FE 62 68                DB    000H,0FEH,'bhxh`',0F0H,000H,'<'
8D68  78 68 60 F0
8D6C  00 3C
8D6E  66 C0 C0 CE                DB    'f',0C0H,0C0H,0CEH,'f>',000H,0CCH
8D72  66 3E 00 CC
8D76  CC CC FC CC                DB    0CCH,0CCH,0FCH,0CCH,0CCH,0CCH,000H
8D7A  CC CC 00
8D7D  78 30 30 30                DB    'x00000x',000H,01EH,00CH,00CH,00CH
8D81  30 30 78 00
8D85  1E 0C 0C 0C
8D89  CC CC 78 00                DB    0CCH,0CCH,'x',000H,0E6H,'flxlf'
8D8D  E6 66 6C 78
8D91  6C 66
8D93  E6 00 F0 60                DB    0E6H,000H,0F0H,'```bf',0FEH,000H
8D97  60 60 62 66
8D9B  FE 00
8D9D  C6 EE FE FE                DB    0C6H,0EEH,0FEH,0FEH,0D6H,0C6H,0C6H
8DA1  D6 C6 C6
8DA4  00 C6 E6 F6                DB    000H,0C6H,0E6H,0F6H,0DEH,0CEH,0C6H
8DA8  DE CE C6
8DAB  C6 00 38 6C                DB    0C6H,000H,'81',0C6H,0C6H,0C6H,'1'
8DAF  C6 C6 C6 6C
8DB3  38 00 FC 66                DB    '8',000H,0FCH,'ff!```',0F0H,000H
8DB7  66 7C 60 60
8DBB  F0 00
8DBD  78 CC CC CC                DB    'x',0CCH,0CCH,0CCH,0DCH,'x',01CH
8DC1  DC 78 1C
8DC4  00 FC 66 66                DB    000H,0FCH,'ff!lf',0E6H,000H,'x'
8DC8  7C 6C 66 E6
8DCC  00 78
8DCE  CC E0 70 1C                DB    0CCH,0E0H,'p',01CH,0CCH,'x',000H
8DD2  CC 78 00
8DD5  FC B4 30 30                DB    0FCH,0B4H,'0000x',000H,0CCH,0CCH
8DD9  30 30 78 00
8DDD  CC CC
8DDF  CC CC CC CC                DB    0CCH,0CCH,0CCH,0CCH,0FCH,000H,0CCH
8DE3  FC 00 CC
8DE6  CC CC CC CC                DB    0CCH,0CCH,0CCH,0CCH,'x0',000H,0C6H
8DEA  78 30 00 C6
8DEE  C6 C6 D6 FE                DB    0C6H,0C6H,0D6H,0FEH,0EEH,0C6H,000H
8DF2  EE C6 00
8DF5  C6 C6 6C 38                DB    0C6H,0C6H,'1881',0C6H,000H,0CCH
8DF9  38 6C C6 00
8DFD  CC
8DFE  CC CC 78 30                DB    0CCH,0CCH,'x00x',000H,0FEH,0C6H
8E02  30 78 00 FE
8E06  C6
8E07  8C 18 32 66                DB    08CH,018H,'2f',0FEH,000H,'x```'
8E0B  FE 00 78 60
8E0F  60 60
8E11  60 60 78 00                DB    '``x',000H,0C0H,'`0',018H,00CH
8E15  C0 60 30 18
8E19  0C
8E1A  06 02 00 78                DB    006H,002H,000H,'x',018H,018H,018H
8E1E  18 18 18
8E21  18 18 78 00                DB    018H,018H,'x',000H,010H,'81',0C6H
8E25  10 38 6C C6
```

```
8E29  00 00 00 00                  DB      000H,000H,000H,000H,000H,000H,000H
8E2D  00 00 00
8E30  00 00 00 00                  DB      000H,000H,000H,000H,0FFH,'00',018H
8E34  FF 30 30 18
8E38  00 00 00 00                  DB      000H,000H,000H,000H,000H,000H,000H
8E3C  00 00 00
8E3F  78 0C 7C CC                  DB      'x',00CH,'|',0CCH,'v',000H,0E0H
8E43  76 00 E0
8E46  60 60 7C 66                  DB      '``|ff',0DCH,000H,000H,000H,'x'
8E4A  66 DC 00 00
8E4E  00 78
8E50  CC C0 CC 78                  DB      0CCH,0C0H,0CCH,'x',000H,01CH,00CH
8E54  00 1C 0C
8E57  0C 7C CC CC                  DB      00CH,'|',0CCH,0CCH,'v',000H,000H
8E5B  76 00 00
8E5E  00 78 CC FC                  DB      000H,'x',0CCH,0FCH,0C0H,'x',000H
8E62  C0 78 00
8E65  38 6C 60 F0                  DB      '8l`',0F0H,'```',0F0H,000H,000H
8E69  60 60 F0 00
8E6D  00
8E6E  00 76 CC CC                  DB      000H,'v',0CCH,0CCH,'|',00CH,0F8H
8E72  7C 0C F8
8E75  E0 60 6C 76                  DB      0E0H,'`lvff',0E6H,000H,'0',000H
8E79  66 66 E6 00
8E7D  30 00
8E7F  70 30 30 30                  DB      'p000x',000H,00CH,000H,00CH,00CH
8E83  78 00 0C 00
8E87  0C 0C
8E89  0C CC CC 78                  DB      00CH,0CCH,0CCH,'x',0E0H,'`flxl'
8E8D  E0 60 66 6C
8E91  78 6C
8E93  E6 00 70 30                  DB      0E6H,000H,'p00000x',000H,000H,000H
8E97  30 30 30 30
8E9B  78 00 00 00
8E9F  CC FE FE D6                  DB      0CCH,0FEH,0FEH,0D6H,0C6H,000H,000H
8EA3  C6 00 00
8EA6  00 F8 CC CC                  DB      000H,0F8H,0CCH,0CCH,0CCH,0CCH,000H
8EAA  CC CC 00
8EAD  00 00 78 CC                  DB      000H,000H,'x',0CCH,0CCH,0CCH,'x'
8EB1  CC CC 78
8EB4  00 00 00 DC                  DB      000H,000H,000H,0DCH,'ff|`',0F0H
8EB8  66 66 7C 60
8EBC  F0
8EBD  00 00 76 CC                  DB      000H,000H,'v',0CCH,0CCH,'|',00CH
8EC1  CC 7C 0C
8EC4  1E 00 00 DC                  DB      01EH,000H,000H,0DCH,'vf`',0F0H
8EC8  76 66 60 F0
8ECC  00 00 00 7C                  DB      000H,000H,000H,'|',0C0H,'x',00CH
8ED0  C0 78 0C
8ED3  F8 00 10 30                  DB      0F8H,000H,010H,'0!004',018H,000H
8ED7  7C 30 30 34
8EDB  18 00
8EDD  00 00 CC CC                  DB      000H,000H,0CCH,0CCH,0CCH,0CCH,'v'
8EE1  CC CC 76
8EE4  00 00 00 CC                  DB      000H,000H,000H,0CCH,0CCH,0CCH,'x'
8EE8  CC CC 78
8EEB  30 00 00 00                  DB      '0',000H,000H,000H,0C6H,0D6H,0FEH
8EEF  C6 D6 FE
8EF2  FE 6C 00 00                  DB      0FEH,'l',000H,000H,000H,0C6H,'l'
8EF6  00 C6 6C
8EF9  38 6C C6 00                  DB      '8l',0C6H,000H,000H,000H,0CCH,0CCH
```

```
8EFD  00 00 CC CC
8F01  CC 7C 0C F8
8F05  00 00 FC
8F08  98 30 64 FC
8F0C  00 1C 30 30
8F10  E0
8F11  30 30 1C 00
8F15  18 18 18 00
8F19  18 18 18 00
8F1D  E0 30 30 1C
8F21  30 30 E0 00
8F25  76 DC 00 00
8F29  00 00 00 00
8F2D  00 10 38
8F30  6C C6 C6 FE
8F34  00
```

```
                        DB      0CCH,'|',00CH,0F8H,000H,000H,0FCH

                        DB      098H,'0d',0FCH,000H,01CH,'00',0E0H

                        DB      '00',01CH,000H,018H,018H,018H,000H

                        DB      018H,018H,018H,000H,0E0H,'00',01CH

                        DB      '00',0E0H,000H,'v',0DCH,000H,000H

                        DB      000H,000H,000H,000H,000H,010H,'8'

                        DB      'l',0C6H,0C6H,0FEH,000H
```

```
                ;genereaza sunet clasic
                ;
8F35  F3        BEEP:   DI
8F36  7D                LD      A,L             ;C=f(L)
8F37  CB 3D              SRL     L
8F39  CB 3D              SRL     L
8F3B  2F                 CPL
8F3C  E6 03              AND     003H
8F3E  4F                 LD      C,A
8F3F  06 00              LD      B,000H
8F41  DD 21 8F4E         LD      IX,L03D1
8F45  DD 09              ADD     IX,BC
8F47  3A FB85            LD      A,(PORTFE)
8F4A  E6 1F              AND     01FH
8F4C  F6 08              OR      008H
8F4E  00        L03D1:  NOP
8F4F  00                 NOP
8F50  00                 NOP
8F51  04                 INC     B
8F52  0C                 INC     C
8F53  0D        L03D6:  DEC     C
8F54  20 FD              JR      NZ,L03D6
8F56  0E 3F              LD      C,03FH
8F58  05                 DEC     B
8F59  C2 8F53            JP      NZ,L03D6
8F5C  EE 10              XOR     010H
8F5E  D3 FE              OUT     (0FEH),A
8F60  32 FB85            LD      (PORTFE),A
8F63  44                 LD      B,H
8F64  4F                 LD      C,A
8F65  CB 67              BIT     4,A
8F67  20 08              JR      NZ,L03F2
8F69  7A                 LD      A,D
8F6A  B3                 OR      E
8F6B  C8                 RET     Z
8F6C  79                 LD      A,C
8F6D  4D                 LD      C,L
8F6E  1B                 DEC     DE
8F6F  DD E9              JP      (IX)
8F71  4D        L03F2:  LD      C,L
8F72  0C                 INC     C
8F73  DD E9              JP      (IX)
8F75  C9        IRET:   RET
```

```
                        ;tratarea intreruperii mascabile
8F76  CD 91AF   ZINTR1: CALL  SALV          ;salveaza registre microprocesor
8F79  3A FB8C           LD    A,(PAPINK)    ;aduce variabila de control
8F7C  CB 7F             BIT   7,A           ;culoare ecran
8F7E  28 60             JR    Z,IYKSTO      ;trebuie redefinite culorile?
8F80  E6 7F     TRAT:   AND   07FH          ;da, mai intii achit cererea de
8F82  32 FB8C           LD    (PAPINK),A    ;redefinire si o reactualizez
8F85  01 5CFD           LD    BC,5CFDH      ;fortez vectorul de SCROLL pe
8F88  AF                XOR   A             ;starea initiala
8F89  ED 79             OUT   (C),A
8F8B  06 0E             LD    B,0EH
8F8D  ED 78             IN    A,(C)         ;retin continutul portului #0EFD
8F8F  F5                PUSH  AF            ;si-l salvez in stiva
8F90  3E 02             LD    A,2           ;simpla rezolutie
8F92  ED 79             OUT   (C),A         ;comut afisarea pe fereastra mica
8F94  CD 89A3           CALL  D17           ;deschide RAM VIDEO
8F97  CD 8BD0           CALL  PAR           ;ecran par
8F9A  21 5830           LD    HL,5830H      ;salveaza 8 octeti din memoria
8F9D  11 58D8           LD    DE,58D8H      ;video,...
8FA0  01 0008           LD    BC,8
8FA3  ED B0             LDIR
8FA5  2E 30             LD    L,30H         ;in vederea fortarii - in locul
8FA7  1E 31             LD    E,31H         ;lor - a atributului de culoare
8FA9  0E 0F             LD    C,0FH         ;pentru PAPER si INK, atribut
8FAB  3A FB8C           LD    A,(PAPINK)    ;memorat la PAPINK
8FAE  77                LD    (HL),A
8FAF  ED B0             LDIR
8FB1  21 8F75           LD    HL,IRET       ;pregatesc o adresa noua de
8FB4  22 FFFE           LD    (0FFFEH),HL   ;tratare a intreruperii, in scop
8FB7  FB                EI     ;de evitare a tratarii ei clasice (prin ZINTR1)
8FB8  76                HALT                ;astept prima intrerupere
8FB9  21 F5F8           LD    HL,ZINTR      ;refac calea clasica de tratare
8FBC  22 FFFE           LD    (0FFFEH),HL   ;intreruperi
8FBF  01 0051   ZZZZ:   LD    BC,51H        ;calibrare care permite alinierea
8FC2  0B        DECBC:  DEC   BC            ;momentului definirii noilor
8FC3  78                LD    A,B           ;culori la momentul in care
8FC4  B1                OR    C             ;spotul luminos exploreaza zona
8FC5  20 FB             JR    NZ,DECBC      ;de memorie cu noul atribut
8FC7  01 0EFD           LD    BC,0EFDH
8FCA  F1                POP   AF            ;refac continutul portului #0EFD
8FCB  ED 79             OUT   (C),A         ;a trecut momentul comutarii!
8FCD  21 58D8           LD    HL,58D8H      ;refac si zona de memorie video
8FD0  1E 30             LD    E,30H         ;distrusa putin mai'nainte
8FD2  01 0008           LD    BC,8
8FD5  ED B0             LDIR
8FD7  3A FB90           LD    A,(ROLL)      ;nu trebuie uitata nici refacerea
8FDA  CD 8B24           CALL  SARI          ;vectorului SCROLL
8FDD  CD 898A           CALL  D18           ;inchide RAM video
8FE0  FD 21 FB70 IYKSTO: LD   IY,KSTO
8FE4  21 FB83           LD    HL,KSTO+13H   ;preiau in A primii opt biti ai
8FE7  7E                LD    A,(HL)        ;contorului CMAN
8FE8  FD CB 10 7E       BIT   7,(IY+10H)    ;motorul este pornit?
8FEC  20 15             JR    NZ,MN0
8FEE  FD CB 10 76       BIT   6,(IY+10H)    ;motor oprit. Sint pe cale de a-l
8FF2  28 31             JR    Z,MN1         ;porni?
8FF4  3D                DEC   A             ;da, decrementez CMAN
8FF5  77                LD    (HL),A        ;reactualizez CMAN
8FF6  B7                OR    A             ;CMAN=0?
8FF7  20 2C             JR    NZ,MN1        ;nu, mai astept rotatii, salt
8FF9  FD CB 10 B6       RES   6,(IY+10H)    ;voi porni motorul
8FFD  FD CB 10 FE       SET   7,(IY+10H)
```

```
9001  18 22              JR    MN1
                  ;motor pornit, imi pun problema opririi lui
                  ;
9003  3D          MN0:  DEC   A           ;decrementez CMAN
9004  77                LD    (HL),A      ;il reactualizez
9005  B7                OR    A           ;este epuizat?
9006  20 1D             JR    NZ,MN1      ;nu, trebuie sa am rabdare
9008  FD CB 10 6E       BIT   5,(IY+10H)  ;da, cel mai semnificativ bit e
900C  28 06             JR    Z,MN2       ;zero? Daca da, salt
900E  FD CB 10 AE       RES   5,(IY+10H)  ;nu, il anulez si mai astept 256
9012  18 11             JR    MN1         ;rotatii
                  ;oprire motor, dupa aproximativ 10 secunde de somaj, datorat
                  ;neutilizarii discurilor
                  ;
9014  FD CB 10 BE   MN2:  RES   7,(IY+10H)  ;marchez motor oprit...
9018  01 1FFD            LD    BC,1FFDH
901B  3A FB84           LD    A,(P1FFD)
901E  E6 F7             AND   0F7H
9020  32 FB84           LD    (P1FFD),A
9023  ED 79              OUT   (C),A       ;si-l opresc
                  ;abia acum imi pun  problema tratarii tastaturii
                  ;
9025  3A FC1F       MN1:  LD    A,(SIRTAS)  ;= 00H => caracter provenit
9028  B7                OR    A           ;din tasta functionala?
9029  20 3B             JR    NZ,CARVAL
902B  CD 875F           CALL  L02BF       ;testeaza daca s-a apasat tasta
902E  FD CB 0B 6E       BIT   5,(IY+0BH)  ;daca nu, salt la TFLASH
9032  CA 918B           JP    Z,TFLASH
9035  16 00             LD    D,000H      ;se produce "click"-ul pentru
9037  FD 5E 0E          LD    E,(IY+0EH)  ;apasarea tastei
903A  21 00C8           LD    HL,00C8H
903D  CD 8F35           CALL  BEEP
9040  3A FB78           LD    A,(LASTK)
9043  FD CB 0D 46       BIT   0,(IY+0DH)  ;test pentru modul de tratare al
9047  20 4D             JR    NZ,MODE1    ;codului
9049  FE 80         MODE0: CP   080H
904B  20 0B             JR    NZ,CP81
904D  FD CB 0D C6       SET   0,(IY+0DH)  ;pentru codul 80H se trece din
9051  3E 08             LD    A,008H      ;MODE=0 in MODE=1 si
9053  FD B6 0C          OR    (IY+0CH)    ;se seteaza CAPS LOCK
9056  18 09             JR    INCSL

9058  FE 81         CP81: CP    081H
905A  20 17             JR    NZ,CPF0
905C  3E 08         CSL:  LD    A,008H      ;complementare CAPS LOCK
905E  FD AE 0C          XOR   (IY+0CH)
9061  FD 77 0C     INCSL: LD   (IY+0CH),A
9064  18 29             JR    NOTAST      ;salt pentru caz de tasta care
                                          ;nu trebuie preluata sub CP/M
9066  FD CB 0B 6E   CARVAL: BIT  5,(IY+0BH)
906A  C2 918B           JP    NZ,TFLASH   ;salt daca nu a fost preluat
906D  FD CB 0B EE       SET   5,(IY+0BH)  ;caracterul precedent
9071  18 0A             JR    HLADRC

9073  FE F0         CPF0: CP    0F0H        ;pentru F0...F7 cazul
9075  38 11             JR    C,CP0       ;tastei functionale
9077  32 FC1F           LD    (SIRTAS),A
907A  CD 917A           CALL  FADCOD      ;depune la (ADRCOD) adresa tastei
907D  2A FC1C     HLADRC: LD    HL,(ADRCOD) ;functionale apasate
9080  7E                LD    A,(HL)      ;(LAST K) <-- codul din sirul
9081  32 FB78           LD    (LASTK),A   ;apelat prin apasarea tastei
```

```
9084  23              INC   HL              ;functionale
9085  22 FC1C         LD    (ADRCOD),HL     ;pointerul (ADRCOD) se
9088  B7        CP0:  OR    A               ;incrementeaza pentru urmatorul
9089  C2 918B         JP    NZ,TFLASH       ;cod de tasta din sirul apelat
908C  32 FC1F         LD    (SIRTAS),A      ;codul 00H este terminator pentru
908F  FD CB 0B AE NOTAST: RES 5,(IY+0BH)    ;sir de taste apelat printr-o
                                            ;tasta functionala
9093  C3 918B         JP    TFLASH          ;salt la analiza functiei FLASH
                                            ;pentru cursor.

                    ;tratare tastatura in MODE=1 (MODE1)
                    ;
9096  FE 80       MODE1: CP    080H
9098  20 06              JR    NZ,TBORD
909A  FD CB 0D 86  GOMODO: RES  0,(IY+0DH)   ;pentru codul 80H (CS+SS) in
909E  18 EF              JR    NOTAST        ;MODE1 se revine in MODE=0
                                             ;(MODE0)

90A0  F5           TBORD: PUSH  AF
90A1  3A FB86             LD    A,(BORDER)   ;tratare comanda pentru schimbare
90A4  FE 01               CP    001H         ;BORDER?
90A6  28 33               JR    Z,IPOPAF
90A8  3A FB8D             LD    A,(CPAPIN)
90AB  CB 47               BIT   0,A
90AD  21 FB8C             LD    HL,PAPINK    ;PAPINK variabila de sistem ce
90B0  20 1C               JR    NZ,COMINK    ;defineste culoarea de PAPER si
90B2  CB 4F               BIT   1,A          ;INK pentru ecran
90B4  28 3E               JR    Z,TTP
90B6  F1           COMPAP: POP  AF
90B7  D6 30               SUB   030H         ;argument PAPER in intervalul
90B9  FE 08               CP    008H         ;[0;7]? Pentru argument incorect
90BB  30 0B               JR    NC,ILDA0     ;nu este afectat PAPINK
90BD  07                  RLCA
90BE  07                  RLCA
90BF  07                  RLCA
90C0  47                  LD    B,A          ;in B argumentul pentru schimbare
90C1  7E                  LD    A,(HL)       ;PAPER
90C2  E6 C7               AND   0C7H
90C4  B0           ILDBA: OR    B
90C5  F6 C0               OR    0C0H
90C7  77                  LD    (HL),A       ;inscrie in PAPINK noua valoare
90C8  AF           ILDA0: XOR   A            ;CPAPIN=0 <=>s-a executat
90C9  32 FB8D             LD    (CPAPIN),A   ;comanda de schimbare PAPER
90CC  18 CC               JR    GOMODO       ;sau INK, revenire in MODE=0

90CE  F1           COMINK: POP  AF
90CF  D6 30               SUB   030H         ;argument INK in intervalul
90D1  FE 08               CP    008H         ;[0,7]? Pentru argument incorect
90D3  30 F3               JR    NC,ILDA0     ;nu se modifica PAPINK
90D5  47                  LD    B,A          ;in B argumentul pentru
90D6  7E                  LD    A,(HL)       ;schimbare INK
90D7  E6 F8               AND   0F8H
90D9  18 E9               JR    ILDBA

90DB  F1           IPOPAF: POP  AF
90DC  D6 30               SUB   030H
90DE  FE 08               CP    008H         ;argument BORDER in intervalul
90E0  30 0C               JR    NC,XORA      ;[0;7]?
90E2  47                  LD    B,A
90E3  3A FB85             LD    A,(PORTFE)   ;modifica BORDER
90E6  E6 F8               AND   0F8H         ;ceilalti biti din portul #**FE
```

```
90E8  B0                    OR    B              ;ramin nemodificati
90E9  D3 FE                 OUT   (0FEH),A
90EB  32 FB85               LD    (PORTFE),A
90EE  AF              XORA: XOR   A              ;iesire din comanda BORDER si
90EF  32 FB86               LD    (BORDER),A     ;revenire in MODE=0
90F2  18 A6                 JR    GOMODO

90F4  3A FC20         TTP:  LD    A,(TP)
90F7  FE F0                 CP    0F0H
90F9  38 4B                 JR    C,T42          ;salt pentru TP=0
90FB  F1                    POP   AF
90FC  FE 7F                 CP    07FH
90FE  30 3F                 JR    NC,GT7F        ;salt daca in modul de definire
9100  FE 20                 CP    020H           ;taste se apasa CS+SS/CSL/tasta
9102  30 15                 JR    NC,GT20        ;functionala
9104  0E 5E                 LD    C,"^"          ;daca la programarea unei taste
9106  CD 8836               CALL  DISPO          ;functionale se apasa CTRL+
9109  F5                    PUSH  AF             ;caracer, se tipareste pe ecran
910A  C6 16                 ADD   A,016H         ;caracterul pecedat de simbolul ^
910C  FE 31                 CP    031H
910E  30 02                 JR    NC,CARINC
9110  C6 2A                 ADD   A,02AH
9112  4F              CARINC: LD    C,A
9113  CD 8836               CALL  DISPO
9116  F1                    POP   AF
9117  18 04                 JR    TIPBL

9119  4F              GT20: LD    C,A            ;afiseaza caracterul programat
911A  CD 8836               CALL  DISPO
911D  0E 20           TIPBL: LD   C," "          ;afiseaza BLANC (spatiu)
911F  CD 8836               CALL  DISPO
9122  2A FC1C               LD    HL,(ADRCOD)
9125  77                    LD    (HL),A         ;incarca codul programat
9126  23                    INC   HL             ;incrementeaza pointerul din
9127  22 FC1C               LD    (ADRCOD),HL    ;generatorul programabil
912A  3A FC1E               LD    A,(NRCOD)      ;decrementeaza contorul
912D  3D                    DEC   A
912E  32 FC1E               LD    (NRCOD),A
9131  C2 908F               JP    NZ,NOTAST      ;daca nu-i depasita limita maxima
9134  AF              TERM: XOR   A              ;de 15 coduri se revine pe calea
9135  2A FC1C               LD    HL,(ADRCOD)    ;de tasta transparenta pentru
9138  77                    LD    (HL),A         ;CP/M; se inscrie codul 00H = cod
9139  32 FC20               LD    (TP),A         ;terminator pentru un sir de
913C  C3 909A               JP    GOMODO         ;coduri programate
                                                 ;se indica terminarea secventei
                                                 ;de programare
                                                 ;se revine in MODE=0

913F  FE 81           GT7F: CP    081H           ;daca se apasa CSL se schimba
9141  CA 905C               JP    Z,CSL          ;starea de CAPSLOCK, altfel se
9144  18 EE                 JR    TERM           ;trece la secventa de incheiere a
                                                 ;programului tastelor
                                                 ;programabile

9146  F1              T42:  POP   AF
9147  FE 42                 CP    042H
9149  20 08                 JR    NZ,T43
914B  3E 01                 LD    A,001H         ;in MODE=1 s-a apasat BORDER =>
914D  32 FB86               LD    (BORDER),A     ;se doreste modificarea BORDER
9150  C3 908F               JP    NOTAST         ;=> BORDER=1
```

```
9153  FE 43          T43:   CP    043H
9155  20 08                 JR    NZ,T58
9157  3E 02                 LD    A,002H          ;in MODE=1 s-a apasat PAPER =>
9159  32 FB8D        ICPI:  LD    (CPAPIN),A      ;se doreste modificare PAPER =>
915C  C3 908F                JP    NOTAST          ;CPAPIN=2

915F  FE 58          T58:   CP    058H
9161  20 04                 JR    NZ,TPF
9163  3E 01                 LD    A,001H          ;in MODE=1 s-a apasat INK =>
9165  18 F2                 JR    ICPI            ;se doreste modificare INK =>
                                                   ;CPAPIN=1

9167  FE F0          TPF:   CP    0F0H
9169  DA 909A               JP    C,GOMODO
916C  32 FC20                LD    (TP),A          ;in MODE=1 s-a apasat o tasta
916F  CD 917A                CALL  FADCOD          ;programabila => TP -->[F0;F7]
9172  3E 0F                 LD    A,00FH          ;in ADRCOD adresa de depunere a
9174  32 FC1E                LD    (NRCOD),A       ;codurilor programabile
9177  C3 908F                JP    NOTAST          ;NRCOD=15 (numar maxim de coduri
                                                   ;definite pentru o tasta
                                                   ;programabila

917A  D6 F0          FADCOD: SUB   0F0H
917C  07                     RLCA
917D  07                     RLCA
917E  07                     RLCA
917F  07                     RLCA
9180  4F                     LD    C,A
9181  06 00                 LD    B,000H
9183  21 FB9C                LD    HL,TTF0
9186  09                     ADD   HL,BC
9187  22 FC1C                LD    (ADRCOD),HL
918A  C9                     RET

918B  FD 35 0F       TFLASH: DEC   (IY+0FH)        ;se decrementeaza FRAME si cind
918E  20 1C                  JR    NZ,D19          ;ajunge la 0 se complementeaza
9190  CD 89A3        ELCUR:  CALL  D17             ;cursorul
9193  CD 8B60                CALL  FHL
9196  06 08                 LD    B,008H
9198  7E             FLASH:  LD    A,(HL)
9199  2F                     CPL
919A  77                     LD    (HL),A
919B  24                     INC   H
919C  10 FA                  DJNZ  FLASH
919E  FD 36 0F 19            LD    (IY+0FH),019H   ;se seteaza FRAME in functie de
91A2  FD CB 0D 46            BIT   0,(IY+0DH)      ;MODE = 0/1 => frecventa de
91A6  20 04                  JR    NZ,D19          ;clipire este in functie de MODE
91A8  FD 36 0F 0C            LD    (IY+0FH),00CH   ;pentru model cursorul clipeste
91AC  C3 8987        D19:    JP    RETCO           ;mai rar

91AF  FD E3          SALV:   EX    (SP),IY
91B1  DD E5                  PUSH  IX
91B3  E5                     PUSH  HL
91B4  D5                     PUSH  DE
91B5  C5                     PUSH  BC
91B6  F5                     PUSH  AF              ;salveaza registrele din
91B7  D9                     EXX                   ;microprocesor
91B8  E5                     PUSH  HL
91B9  D5                     PUSH  DE
91BA  C5                     PUSH  BC
91BB  D9                     EXX
```

```
91BC  08              EX    AF,AF'
91BD  F5              PUSH  AF
91BE  08              EX    AF,AF'
91BF  FD E5           PUSH  IY
91C1  C9              RET

91C2  FD E1     REF:  POP   IY          ;reface registrele din micropr.
91C4  08              EX    AF,AF'
91C5  F1              POP   AF
91C6  08              EX    AF,AF'
91C7  D9              EXX
91C8  C1              POP   BC
91C9  D1              POP   DE
91CA  E1              POP   HL
91CB  D9              EXX
91CC  F1              POP   AF
91CD  C1              POP   BC
91CE  D1              POP   DE
91CF  E1              POP   HL
91D0  DD E1           POP   IX
91D2  FD E3           EX    (SP),IY
91D4  C9              RET


91D5  00        ML:   DB    0         ;Indicatori mod de lucru
91D6  00        NCR:  DB    0         ;Numar de caractere in regim grafic
91D7  FF        TDR:  DB    0FFH      ;Tipul dreptei
91D8  00        INDIC:DB    0         ;Indicatori folositi in regim grafic
91D9  00        LOYV: DB    0         ;Coordonate vechi
91DA  00        HIYV: DB    0
91DB  00        LOXV: DB    0
91DC  00        HIXV: DB    0
91DD  00        LOYN: DB    0         ;Coordonate noi
91DE  00        HIYN: DB    0
91DF  00        LOXN: DB    0
91E0  00        HIXH: DB    0
91E1  00        LEFD: DB    0

                ;SEMNIFICATIA INDICATORILOR:
                ;       -ML0-   0
                ;               1 Comenzi speciale
                ;       -ML1-
                ;       -ML2-   0
                ;               1 Mod grafic
                ;       -ML3-   0
                ;               1 Tip dreapta
                ;       -ML4-   0
                ;               1 Numar caractere alfanumerice in regim grafic
                ;       -ML5-   0 imprimanta SCAMP
                ;               1 imprimanta ROBOTRON
                ;       -ML6-   0 copie ecran 1/1
                ;               1 copie ecran 2/1
                ;       -ML7-   0 pimii 4 biti din tipul dreptei
                ;               1 urmatorii 4 biti din tipul dreptei
                ;       -INDIC0-0 HIY
                ;               1 HIX
                ;       -INDIC1-0 NU se traseaza dreapta
                ;               1 DA se traseaza dreapta
                ;       -INDIC2-0 NU se traseaza dreapta
                ;               1 DA se traseaza dreapta
                ;       -INDIC3-
```

```
                        ;       -INDIC4-
                        ;       -INDIC5-0 se traseaza o dreapta
                        ;                 1 se traseaza un punct
                        ;       -INDIC6-0 YN <> YV
                        ;                 1 YN =  YV
                        ;       -INDIC7-0 XN <> XV
                        ;                 1 XN =  XV
                        ;       -INDICA0-
                        ;       -INDICA1-
                        ;       -INDICA2-
                        ;       -INDICA3-0 sgnXABS= -
                        ;                 1 sgnXABS= +
                        ;       -INDICA4-0 sgnYABS= -
                        ;                 1 sgnYABS= +
                        ;       -INDICA5-0 YABS <= XABS deplasare mai mare pe orizontala
                        ;                 1 YABS > XABS deplasare mai mare pe verticala
                        ;       -INDICA6-0 deplasare pe orizontala sau pe verticala
                        ;                 1 deplasare pe una din diagonale
                        ;       -INDICA7-

                        ;DESCHIDERE PAGINI VIDEO

91E2  01 0EFD           ZONA1:  LD      BC,0EFDH
91E5  ED 78                     IN      A,(C)
91E7  E6 9F                     AND     9FH
91E9  ED 79                     OUT     (C),A
91EB  C9                        RET
91EC  01 0EFD           ZONA2:  LD      BC,0EFDH
91EF  ED 78                     IN      A,(C)
91F1  F6 20                     OR      20H
91F3  E6 BF                     AND     0BFH
91F5  ED 79                     OUT     (C),A
91F7  C9                        RET
91F8  01 0EFD           ZONA3:  LD      BC,0EFDH
91FB  ED 78                     IN      A,(C)
91FD  F6 40                     OR      40H
91FF  E6 DF                     AND     0DFH
9201  ED 79                     OUT     (C),A
9203  C9                        RET

                        ;SCROLL IN REGIM MIXT ALFANUMERIC SI GRAFIC

9204  17                C36:    DB      CARR-1
9205  00                SCR0:   DB      0
9206  00                NCRR1:  DB      0
9207  00 58 20 58       TAB1:   DB      0,58H,20H,58H,40H,58H,60H,58H,80H,58H,0A0H,58H
920B  40 58 60 58
920F  80 58 A0 58
9213  C0 58 E0 58              DB      0C0H,58H,0E0H,58H,8,58H,28H,58H,48H,58H,68H,58H
9217  08 58 28 58
921B  48 58 68 58
921F  88 58 A8 58              DB      88H,58H,0A8H,58H,0C8H,58H,0E8H,58H,10H,58H,30H
9223  C8 58 E8 58
9227  10 58 30
922A  58 50 58 70              DB      58H,50H,58H,70H,58H,90H,58H,0B0H,58H,0D0H,58H
922E  58 90 58 B0
9232  58 D0 58
9235  F0 58 18 48              DB      0F0H,58H,18H,48H,38H,48H,58H,48H,78H,48H,98H
9239  38 48 58 48
923D  78 48 98
```

```
9240  48 B8 48 D8                DB    48H,0B8H,48H,0D8H,48H,0F8H,48H,0,58H,20H,58H
9244  48 F8 48 00
9248  58 20 58
924B  40 58 60 58                DB    40H,58H,60H,58H,0,58H
924F  00 58
9251  00                 NCRR2:  DB    0
9252  00 40 20 40        TAB2:   DB    0,40H,20H,40H,40H,40H,60H,40H,80H,40H,0A0H,40H
9256  40 40 60 40
925A  80 40 A0 40
925E  C0 40 E0 40                DB    0C0H,40H,0E0H,40H,0,48H,20H,48H,40H,48H,60H,48H
9262  00 48 20 48
9266  40 48 60 48
926A  80 48 A0 48                DB    80H,48H,0A0H,48H,0C0H,48H,0E0H,48H,0,50H,20H,50H
926E  C0 48 E0 48
9272  00 50 20 50
9276  40 50 60 50                DB    40H,50H,60H,50H,80H,50H,0A0H,50H,0C0H,50H,0E0H
927A  80 50 A0 50
927E  C0 50 E0
9281  50 00 58 20                DB    50H,0,58H,20H,58H,40H,58H,60H,58H,80H,58H,0A0H
9285  58 40 58 60
9289  58 80 58 A0
928D  58 C0 58 E0                DB    58H,0C0H,58H,0E0H,58H,0,40H,20H,40H,40H,40H,60H
9291  58 00 40 20
9295  40 40 40 60
9299  40 00 40                   DB    40H,0,40H
929C  CD 8BD0      SCROL1: CALL  PAR
929F  CD 92D3              CALL  SCEXE
92A2  CD 8BC6              CALL  IMPAR
92A5  CD 92D3              CALL  SCEXE
92A8  3A FB8E              LD    A,(Y)          ;Trece la linia urmatoare
92AB  3C                   INC   A
92AC  3D                   DEC   A
92AD  28 03                JR    Z,S40
92AF  3D                   DEC   A
92B0  18 02                JR    S41
92B2  3E 17        S40:    LD    A,CARR-1
92B4  32 FB8E      S41:    LD    (Y),A
92B7  3A 91D6              LD    A,(NCR)
92BA  CB 3F                SRL   A
92BC  FE 09                CP    9              ;Exista mai mult de 16 caractere
92BE  38 0B                JR    C,S42          ;pe un rind ?
92C0  D6 08                SUB   8              ;DA
92C2  3D                   DEC   A
92C3  4F                   LD    C,A
92C4  06 03                LD    B,3
92C6  CD 8BFD              CALL  LIZOST
92C9  3E 08                LD    A,8
92CB  3D           S42:    DEC   A              ;NU
92CC  4F                   LD    C,A
92CD  06 01                LD    B,1
92CF  CD 8BFD              CALL  LIZOST
92D2  C9                   RET
92D3  3A FB90      SCEXE:  LD    A,(ROLL)       ;Preluare pozitie SCROLL HARD
92D6  32 9205              LD    (SCR0),A       ;de pe ecran
92D9  3A 91D6              LD    A,(NCR)
92DC  CB 3F                SRL   A
92DE  FE 09                CP    9              ;Exista mai mult de 16 caractere
92E0  38 0B                JR    C,S4           ;pe un rind ?
92E2  D6 08                SUB   8              ;DA
92E4  DD 21 9252           LD    IX,TAB2
92E8  32 9251              LD    (NCRR2),A
```

```
92EB  18 0D           JR    S5
92ED  32 9206   S4:   LD    (NCRR1),A     ;NU
92F0  DD 21 9207      LD    IX,TAB1
92F4  3A FB90         LD    A,(ROLL)
92F7  32 9205         LD    (SCR0),A
92FA  3A 9205   S5:   LD    A,(SCR0)      ;Preluare pozitie SCROLL SOFT
92FD  CB 27           SLA   A             ;de pe ecran
92FF  32 9310         LD    (S6+2),A
9302  3C              INC   A
9303  32 9313         LD    (S7+2),A
9306  3C              INC   A
9307  32 9316         LD    (S8+2),A
930A  3C              INC   A
930B  32 9319         LD    (S9+2),A
930E  DD 5E 01  S6:   LD    E,(IX+1)      ;Incarca DE cu adresa liniei TV
9311  DD 56 01  S7:   LD    D,(IX+1)      ;unde se muta informatia
9314  DD 6E 01  S8:   LD    L,(IX+1)      ;Incarca HL cu adresa liniei TV
9317  DD 66 01  S9:   LD    H,(IX+1)      ;de unde se muta informatia
931A  F5              PUSH  AF
931B  3E 18           LD    A,CARR        ;Exista doar 24 de rinduri
931D  FE 18           CP    24            ;alfanumerice ?
931F  20 16           JR    NZ,S50
9321  F1              POP   AF            ;DA
9322  FE 31           CP    49            ;Sintem in rindul 24 ?
9324  20 18           JR    NZ,S51
9326  DD 7E 01        LD    A,(IX+1)      ;DA
9329  FE 58           CP    58H           ;Sintem in zona primelor 16
932B  28 05           JR    Z,S52         ;caractere ?
932D  21 4000         LD    HL,4000H      ;NU
9330  18 0C           JR    S51
9332  21 5800   S52:   LD    HL,5800H
9335  18 07           JR    S51
9337  F1        S50:   POP   AF
9338  FE 31           CP    49
933A  28 52           JR    Z,TRZ21       ;Se deschid zonele video necesare
933C  30 05           JR    NC,S20        ;efectuarii transferului de
933E  CD 91E2   S51:  CALL  ZONA1         ;informatie sau se face apel la
9341  18 12           JR    S22           ;subrutine ce transfera
9343  FE 41     S20:  CP    65            ;informatie dintr-o pagina video in
9345  28 51           JR    Z,TRZ32       ;alta pagina video
9347  30 05           JR    NC,S21
9349  CD 91EC         CALL  ZONA2
934C  18 07           JR    S22
934E  FE 49     S21:  CP    73
9350  28 50           JR    Z,TRZ13
9352  CD 91F8         CALL  ZONA3
9355  3E 08     S22:  LD    A,8           ;Numar de rinduri TV dintr-un
9357  06 00           LD    B,0           ;rind alfanumeric
9359  DD 4E FF  S10:  LD    C,(IX-1)      ;Numar de caractere pe un rind
935C  D5              PUSH  DE
935D  E5              PUSH  HL
935E  ED B0           LDIR                ;Se efectueaza transferul in
9360  E1              POP   HL            ;cadrul aceleiasi pagini video
9361  D1              POP   DE

9362  24              INC   H
9363  14              INC   D
9364  3D              DEC   A
9365  20 F2           JR    NZ,S10
9367  3A 9205   S14:  LD    A,(SCR0)      ;Se trece la rindul alfanumeric
936A  3C              INC   A             ;urmator
```

```
936B  FE 18              CP    CARR
936D  20 02              JR    NZ,S11
936F  3E 00              LD    A,0
9371  32 9205      S11:  LD    (SCRO),A
9374  3A 9204            LD    A,(C36)
9377  3D                 DEC   A              ;A fost efectuat scroll-ul pe 36
9378  32 9204            LD    (C36),A        ;de rinduri ?
937B  C2 92FA            JP    NZ,S5
937E  3E 17              LD    A,CARR-1
9380  32 9204            LD    (C36),A
9383  DD 7E 01           LD    A,(IX+1)
9386  FE 58              CP    58H
9388  C8                 RET   Z              ;Sfirsit daca exista mai putin
9389  3E 08              LD    A,8            ;de 16 caractere pe rind
938B  C3 92ED            JP    S4
938E  E5          TRZ21: PUSH  HL             ;Se realizeaza transferul dintr-o
938F  D5                 PUSH  DE             ;zona video in  alta zona video
9390  21 91EC            LD    HL,ZONA2
9393  11 91E2            LD    DE,ZONA1
9396  18 12              JR    S15
9398  E5          TRZ32: PUSH  HL
9399  D5                 PUSH  DE
939A  21 91F8            LD    HL,ZONA3
939D  11 91EC            LD    DE,ZONA2
93A0  18 08              JR    S15
93A2  E5          TRZ13: PUSH  HL
93A3  D5                 PUSH  DE
93A4  21 91E2            LD    HL,ZONA1
93A7  11 91F8            LD    DE,ZONA3
93AA  22 93BD      S15:  LD    (ZN1+1),HL
93AD  ED 53 93C2         LD    (ZN2+1),DE
93B1  D1                 POP   DE
93B2  E1                 POP   HL
93B3  06 08              LD    B,8            ;Numar de rinduri TV dintr-un rind
93B5  C5           S13:  PUSH  BC             ;alfanumeric
93B6  DD 46 FF           LD    B,(IX-1)       ;Numar de caractere pe rind
93B9  D5                 PUSH  DE             ;Adresa unde se depune informatia
93BA  E5                 PUSH  HL             ;Adresa de unde se ia informatia
93BB  C5           S12:  PUSH  BC
93BC  CD 0000      ZN1:  CALL  0              ;Se deschide o zona video
93BF  7E                 LD    A,(HL)
93C0  F5                 PUSH  AF
93C1  CD 0000      ZN2:  CALL  0              ;Se deschide alta zona video
93C4  F1                 POP   AF
93C5  12                 LD    (DE),A
93C6  2C                 INC   L
93C7  1C                 INC   E
93C8  C1                 POP   BC
93C9  10 F0              DJNZ  S12
93CB  E1                 POP   HL
93CC  D1                 POP   DE
93CD  24                 INC   H
93CE  14                 INC   D
93CF  C1                 POP   BC
93D0  10 E3              DJNZ  S13
93D2  18 93              JR    S14

                   ;STERGEREA ECRANULUI

93D4  11 91D6      ESCFF: LD   DE,NCR
93D7  1A                  LD   A,(DE)
```

```
93D8  FE 00              CP   0
93DA  20 0E              JR   NZ,FF6          ;Salt pt modul de lucru mixt
93DC  CB 96              RES  2,(HL)          ;alfanumeric si grafic
93DE  CD 9462            CALL STEREC
93E1  3A FB90      FF20: LD   A,(ROLL)        ;Cursor in HOME
93E4  32 FB8E            LD   (Y),A
93E7  C3 8980            JP   STESC
93EA  CB 3F        FF6:  SRL  A               ;Sterge zona alfanumerica sau
93EC  57                 LD   D,A            ;zona grafica, functie de regimul
93ED  FE 08              CP   8              ;de lucru in care se afla
93EF  20 06              JR   NZ,FF8         ;Salt pentru un numar diferit de
93F1  CB 56              BIT  2,(HL)         ;16 caractere pe rind
93F3  28 04              JR   Z,FF81         ;Salt pentru regim alfanumeric
93F5  18 10              JR   FF9            ;Salt pentru regim grafic
93F7  30 0E        FF8:  JR   NC,FF9         ;Salt pentru mai putin de 16
                                             ;caractere
93F9  CD 8BD0      FF81: CALL PAR             ;Se executa stergerea pentru mai
93FC  CD 94A6            CALL ECON16         ;putin de 16 caractere pe rind
93FF  CD 8BC6            CALL IMPAR
9402  CD 94A6            CALL ECON16
9405  18 0C              JR   FF10
9407  CD 8BD0      FF9:  CALL PAR             ;Se executa stergerea pentru mai
940A  CD 9472            CALL ECOM16         ;mult de 16 caractere pe ri i
940D  CD 8BC6            CALL IMPAR
9410  CD 9472            CALL ECOM16
9413  7A          FF10: LD   A,D
9414  FE 08              CP   8
9416  21 91D5            LD   HL,ML
9419  20 06              JR   NZ,FF11
941B  CB 56              BIT  2,(HL)         ;Daca sint 16 caractere pe rind,
941D  28 2A              JR   Z,SFRa         ;se iese din subrutina
941F  18 2B              JR   SFRGR
9421  30 14        FF11: JR   NC,FF12
9423  CB 56              BIT  2,(HL)         ;Daca sint mai putin de 16
9425  28 22              JR   Z,SFRa         ;caractere:
9427  16 08              LD   D,8            ;-in reg alfanumeric se iese din
9429  CD 8BD0            CALL PAR            ; subrutina;
942C  CD 9472            CALL ECOM16         ;-in regim grafic se sterge zona de
942F  CD 8BC6            CALL IMPAR          ; dupa primele 16 caractere
9432  CD 9472            CALL ECOM16
9435  18 15              JR   SFRGR
9437  CB 56        FF12: BIT  2,(HL)         ;Daca-s mai mult de 16 caractere:
9439  20 11              JR   NZ,SFRGR       ;-in reg grafic se iese din
943B  16 08              LD   D,8            ; subrutina
943D  CD 8BD0            CALL PAR            ;-in regim alfanumeric se sterge
9440  CD 94A6            CALL ECON16         ; zona primelor 16 caractere
9443  CD 8BC6            CALL IMPAR
9446  CD 94A6            CALL ECON16
9449  C3 93E1      SFRa: JP   FF20           ;Revenire din subrutina in regim
                                             ;alfanumeric
944C  21 91D9      SFRGR: LD  HL,LOYV        ;Revenire din subrutina in regim
944F  06 04              LD   B,4            ;grafic
9451  36 00        FF13: LD   (HL),0
9453  23                 INC  HL
9454  10 FB              DJNZ FF13
9456  C3 8980            JP   STESC

                   ;STERGEREA INTREGULUI ECRAN

9459  CD 89A3      ERASE1: CALL D17           ;Deschide RAM VIDEO
945C  CD 9462              CALL STEREC
```

```
945F  C3 898A            JP    D18          ;Inchide RAM VIDEO
9462  CD 8BD0  STEREC:  CALL  PAR
9465  CD 946B            CALL  STERA
9468  CD 8BC6            CALL  IMPAR
946B  16 08      STERA:  LD    D,8
946D  CD 94A6            CALL  ECON16
9470  16 28              LD    D,40
9472  7A         ECOM16: LD    A,D          ;Se sterge zona de dupa primele
9473  D6 08              SUB   8            ;16 caractere
9475  57                 LD    D,A
9476  3E 20              LD    A,32
9478  92                 SUB   D
9479  5F                 LD    E,A
947A  CD 91E2            CALL  ZONA1        ;Se deschide prima zona video
947D  21 4000            LD    HL,4000H     ;Adresa de inceput a acestei zone
9480  0E C0              LD    C,0C0H       ;Lungimea acestei zone
9482  CD 94CD            CALL  EXEaGR
9485  3E 18              LD    A,CARR       ;Exista doar 24 de rinduri
9487  FE 18              CP    24           ;alfanumerice ?
9489  28 16              JR    Z,ECOM16E
948B  CD 91EC            CALL  ZONA2        ;NU
948E  21 5800            LD    HL,5800H
9491  0E 40              LD    C,40H
9493  CD 94CD            CALL  EXEaGR
9496  CD 91F8            CALL  ZONA3
9499  21 4000            LD    HL,4000H
949C  0E 40              LD    C,40H
949E  CD 94CD            CALL  EXEaGR
94A1  3E 08      ECOM16E:LD    A,8
94A3  82                 ADD   A,D
94A4  57                 LD    D,A
94A5  C9                 RET

94A6  3E 08      ECON16: LD    A,8          ;Se sterge in zona primelor
94A8  92                 SUB   D            ;16 caractere
94A9  5F                 LD    E,A
94AA  CD 91E2            CALL  ZONA1        ;Se deschide prima zona video
94AD  21 5800            LD    HL,5800H     ;Adresa de inceput a acestei zone
94B0  0E FF              LD    C,0FFH       ;Lungimea acestei zone
94B2  CD 94CD            CALL  EXEaGR
94B5  3E 18              LD    A,CARR       ;Exista doar 24 de rinduri
94B7  FE 18              CP    24           ;alfanumerice ?
94B9  C8                 RET   Z
94BA  CD 91EC            CALL  ZONA2        ;NU
94BD  21 4818            LD    HL,4818H
94C0  0E FD              LD    C,0FDH
94C2  CD 94CD            CALL  EXEaGR
94C5  CD 91F8            CALL  ZONA3
94C8  21 5800            LD    HL,5800H
94CB  0E ED              LD    C,0EDH
94CD  E5         EXEaGR: PUSH  HL           ;Se executa stergerea atit in
94CE  21 91D5            LD    HL,ML        ;regim grafic cit si in regim
94D1  CB 56              BIT   2,(HL)       ;alfanumeric
94D3  E1                 POP   HL
94D4  20 14              JR    NZ,EXEGR     ;Salt pentru regim grafic
94D6  3A FB93    EXEa:   LD    A,(VIDEO)    ;Stergere in regim alfanumeric in
94D9  42         Ea2:    LD    B,D          ;VIDEO normal sau invers
94DA  77         Ea4:    LD    (HL),A       ;In B numarul de caractere/rind
94DB  23                 INC   HL
94DC  10 FC              DJNZ  Ea4
94DE  43                 LD    B,E          ;In B avem 16 respectiv 64 minus
```

```
94DF  04              INC   B          ;numarul de caractere pe rind
94E0  05              DEC   B
94E1  28 03           JR    Z,Ea3
94E3  23        Ea1:  INC   HL
94E4  10 FD           DJNZ  Ea1
94E6  0D        Ea3:  DEC   C          ;In C numarul de rinduri
94E7  20 F0           JR    NZ,Ea2
94E9  C9              RET
94EA  3A FB93   EXEGR: LD   A,(VIDEO)  ;Sterge in regim grafic in
94ED  42        EGR2: LD    B,D        ;VIDEO normal sau invers
94EE  04              INC   B          ;In B numarul caracterelor pe rind
94EF  05              DEC   B
94F0  28 03           JR    Z,EGR3
94F2  23        EGR1: INC   HL
94F3  10 FD           DJNZ  EGR1
94F5  43        EGR3: LD    B,E        ;In B avem 16 respectiv 64 minus
94F6  77        EGR4: LD    (HL),A     ;numarul de caractere pe rind
94F7  23              INC   HL
94F8  10 FC           DJNZ  EGR4
94FA  0D              DEC   C          ;In C numar de rinduri
94FB  20 F0           JR    NZ,EGR2
94FD  C9              RET

                ;INTRODUCERE GRAFICA

94FE  00        CON4: DB    0
94FF  00 00     XC:   DB    0,0
9501  00 00     YC:   DB    0,0
9503  00        TDRM: DB    0
9504  00 00     XCM:  DB    0,0
9506  00 00     YCM:  DB    0,0
9508  3A 91D7   ESCSUB: LD   A,(TDR)
950B  32 9503         LD    (TDRM),A   ;Memoreaza tipul dreptei
950E  3E 55           LD    A,55H
9510  32 91D7         LD    (TDR),A    ;Tipul dreptei linie punctata
9513  21 9190         LD    HL,ELCUR   ;Elimina afisarea cursorului in
9516  11 91AC         LD    DE,D19     ;timpul modului de lucru IG
9519  36 C3           LD    (HL),0C3H
951B  23              INC   HL
951C  73              LD    (HL),E
951D  23              INC   HL
951E  72              LD    (HL),D
951F  3E AE           LD    A,0AEH     ;Modificarea subrutina de trasare
9521  32 A34D         LD    (PLOTOA),A ;a dreptelor pe orizontala si pe
9524  3E A9           LD    A,0A9H     ;verticala, pentru a fi utilizate
9526  32 A3C2         LD    (PLOTVA),A ;la trasarea cursorului in regim
9529  21 240D         LD    HL,240DH   ;IG
952C  22 A3C6         LD    (PLOTVB),HL
952F  21 0707         LD    HL,0707H
9532  22 A3AB         LD    (PLOTVC),HL
9535  AF              XOR   A
9536  2A A299         LD    HL,(XMAX)  ;Stabilirea limitelor pe
9539  E5              PUSH  HL         ;orizontala
953A  ED 5B 94FF      LD    DE,(XC)
953E  ED 52           SBC   HL,DE      ;Incadrarea cursorului in
9540  E1              POP   HL         ;limitele existente
9541  30 03           JR    NC,SU30
9543  22 94FF         LD    (XC),HL
9546  CD 9646   SU30: CALL  CURO       ;Trasarea cursorului
9549  CD 965C         CALL  CURV
954C  4F        SU2:  LD    C,A
```

```
954D  FB          SU3:  EI              ;Citirea tastaturii pentru a
954E  3E F7       SU32: LD    A,0F7H    ;sesiza daca una din tastele 5,
9550  DB FE             IN    A,(0FEH)  ; 6, 7 sau 8 a fost actionata in
9552  CB 67             BIT   4,A       ;mod continuu sau intermitent
9554  28 0D             JR    Z,SU31
9556  3E EF             LD    A,0EFH
9558  DB FE             IN    A,(0FEH)
955A  F6 E3             OR    0E3H
955C  EE FF             XOR   0FFH
955E  20 03             JR    NZ,SU31
9560  4F                LD    C,A
9561  18 07             JR    SU33
9563  CD 8705     SU31:  CALL  KBSTS
9566  FE 00             CP    0
9568  28 E4             JR    Z,SU32
956A  CD 86C3     SU33:  CALL  KBINP
956D  F3                DI
956E  06 01             LD    B,1       ;Deplasare cu 1 rind (coloana) TV
9570  FE 35             CP    35H       ;Este actionata tasta 5 ?
9572  20 16             JR    NZ,SU1
9574  CD 96BA     SU20:  CALL  CURVS;DA, deplaseaza cursor o pozitie spre stinga
9577  B9                CP    C         ;E actionata continuu tasta 5 ?
9578  20 D2             JR    NZ,SU2
957A  3E F7       SU4:   LD    A,0F7H    ;DA, se citeste direct tasta 5 si
957C  DB FE              IN    A,(0FEH)  ;cit timp e actionata se
957E  CB 67              BIT   4,A       ;realizeaza deplasari ale
9580  20 CA              JR    NZ,SU2    ;cursorului spre stinga
9582  CD 96BA            CALL  CURVS
9585  CD 96D9            CALL  TAS67     ;Daca simultan cu tasta 5 e
9588  18 F0              JR    SU4       ;actionata una din tastele 6 sau
                                         ;7, cursorul e deplasat simultan
                                         ;stinga si jos, respectiv stinga
                                         ;si sus
958A  FE 38       SU1:   CP    38H       ;E actionata tasta 8 ?
958C  20 16              JR    NZ,SU5
958E  CD 96A4     SU21:  CALL  CURVD     ;DA, idem tasta 5 dar cu
9591  B9                CP    C         ;deplasare spre dreapta
9592  20 B8              JR    NZ,SU2
9594  3E EF       SU6:   LD    A,0EFH
9596  DB FE              IN    A,(0FEH)
9598  CB 57              BIT   2,A
959A  20 B0              JR    NZ,SU2
959C  CD 96A4            CALL  CURVD
959F  CD 96D9            CALL  TAS67
95A2  18 F0              JR    SU6
95A4  FE 36       SU5:   CP    36H       ;E actionata tasta 6 ?
95A6  20 16              JR    NZ,SU7
95A8  CD 9685     SU22:  CALL  CUROJ     ;DA, idem tasta 5, dar deplasarea
95AB  B9                 CP    C         ;se face in jos, combinat cu
95AC  20 9E              JR    NZ,SU2    ;deplasari stinga sau dreapta
95AE  3E EF       SU8:   LD    A,0EFH
95B0  DB FE              IN    A,(0FEH)
95B2  CB 67              BIT   4,A
95B4  20 96              JR    NZ,SU2
95B6  CD 9685            CALL  CUROJ
95B9  CD 96F0            CALL  TAS58
95BC  18 F0              JR    SU8
95BE  FE 37       SU7:   CP    37H       ;Este actionata tasta 7 ?
95C0  20 17              JR    NZ,SU9
95C2  CD 9670     SU23:  CALL  CUROS     ;DA, idem tasta 6 dar cu
95C5  B9                 CP    C         ;deplasare in sus
```

```
95C6  20 84              JR    NZ,SU2
95C8  3E EF      SU10:   LD    A,0EFH
95CA  DB FE              IN    A,(0FEH)
95CC  CB 5F              BIT   3,A
95CE  C2 954C            JP    NZ,SU2
95D1  CD 9670            CALL  CUROS
95D4  CD 96F0            CALL  TAS58
95D7  18 EF              JR    SU10
95D9  06 08      SU9:    LD    B,8         ;In continuare, deplasarile se
                                           ;fac cu cite 8 pozitii in
                                           ;fiecare directie
95DB  FE 25              CP    25H         ;Tasta % (SS+5)
95DD  28 95              JR    Z,SU20
95DF  FE 28              CP    28H         ;Tasta ( (SS+8)
95E1  28 AB              JR    Z,SU21
95E3  FE 26              CP    26H         ;Tasta & (SS+6)
95E5  28 C1              JR    Z,SU22
95E7  FE 27              CP    27H         ;Tasta ' (SS+7)
95E9  28 D7              JR    Z,SU23
95EB  2A 94FF            LD    HL,(XC)     ;Daca a fost actionata alta
95EE  CD 9630            CALL  XCXCM       ;tasta, coordonatele cursorului
95F1  22 9504            LD    (XCM),HL    ;sint depuse intr-o zona de
95F4  2A 9501            LD    HL,(YC)     ;memorie pentru a fi preluate de
95F7  CD 9630            CALL  XCXCM       ;catre CONIN
95FA  22 9506            LD    (YCM),HL
95FD  21 94FE            LD    HL,CON4     ;Indica iesirea din regimul IG
9600  36 05              LD    (HL),5      ;Primele 5 apeluri la CONIN vor
                                           ;avea ca rezultat extragerea
                                           ;coordonatelor cursorului si a
                                           ;tastei care a fost actionata
9602  CD 9646            CALL  CURO        ;Stergere cursor de tip IG
9605  CD 965C            CALL  CURV
9608  3A 9503            LD    A,(TDRM)    ;Se reface tipul dreptei
960B  32 91D7            LD    (TDR),A
960E  21 9190            LD    HL,ELCUR    ;Se revine la cursorul normal
9611  11 89A3            LD    DE,D17
9614  36 CD              LD    (HL),0CDH
9616  23                 INC   HL
9617  73                 LD    (HL),E
9618  23                 INC   HL
9619  72                 LD    (HL),D
961A  3E 00              LD    A,0
961C  32 A34D            LD    (PLOTOA),A  ;Se refac subrutinele grafice
961F  3E B1              LD    A,0B1H      ;pentru a putea fi utilizate in
9621  32 A3C2            LD    (PLOTVA),A  ;regim grafic
9624  21 0000            LD    HL,0
9627  22 A3C6            LD    (PLOTVB),HL
962A  21 0F00            LD    HL,0F00H
962D  22 A3AB            LD    (PLOTVC),HL
9630  C3 8980            JP    STESC
9633  06 03      XCXCM:  LD    B,3         ;Se prelucreaza coordonatele
9635  CB 25      XCX1:   SLA   L           ;pentru a fi preluate de catre
9637  CB 14              RL    H           ;CONIN
9639  10 FA              DJNZ  XCX1
963B  06 03              LD    B,3
963D  CB 3D      XCX2:   SRL   L
963F  10 FC              DJNZ  XCX2
9641  CB EC              SET   5,H
9643  CB ED              SET   5,L
9645  C9                 RET
9646  D5         CURO:   PUSH  DE          ;Trasare segment orizontal din
```

```
9647  CB DF              SET   3,A          ;cursor IG
9649  ED 5B A299         LD    DE,(XMAX)
964D  13                 INC   DE
964E  D9                 EXX
964F  ED 4B 9501         LD    BC,(YC)
9653  11 0000           LD    DE,0
9656  CD A2F6           CALL  ORIZ
9659  D9                 EXX
965A  D1                 POP   DE
965B  C9                 RET
965C  C5           CURV: PUSH  BC           ;Trasare segment vertical din
965D  CB E7              SET   4,A          ;cursor IG
965F  01 00C0            LD    BC,CARR*8
9662  D9                 EXX
9663  01 0000           LD    BC,0
9666  ED 5B 94FF         LD    DE,(XC)
966A  CD A353            CALL  VERTIC
966D  D9                 EXX
966E  C1                 POP   BC
966F  C9                 RET
9670  C5          CUROS: PUSH  BC           ;Deplasare sgment orizontal din
9671  ED 5B 9501         LD    DE,(YC)      ;cursor IG in sus
9675  13          CUROS2: INC  DE
9676  10 FD              DJNZ  CUROS2
9678  B7                 OR    A
9679  21 00BF            LD    HL,CARR*8-1
967C  ED 52              SBC   HL,DE
967E  30 18              JR    NC,CUROJ1
9680  11 00BF            LD    DE;CARR*8-1
9683  18 13              JR    CUROJ1
9685  C5          CUROJ: PUSH  BC           ;Deplasare segment orizontal din
9686  ED 5B 9501         LD    DE,(YC)      ;cursor IG in jos
968A  1B          CUROJ2: DEC  DE
968B  10 FD              DJNZ  CUROJ2
968D  B7                 OR    A
968E  21 00BF            LD    HL,CARR*8-1
9691  ED 52              SBC   HL,DE
9693  30 03              JR    NC,CUROJ1
9695  11 0000           LD    DE,0
9698  CD 9646    CUROJ1: CALL  CURO
969B  ED 53 9501         LD    (YC),DE
969F  CD 9646            CALL  CURO
96A2  C1                 POP   BC
96A3  C9                 RET
96A4  C5           CURVD: PUSH  BC           ;Deplasare segment vertical din
96A5  ED 5B 94FF         LD    DE,(XC)      ;cursor IG spre dreapta
96A9  13          CURVD2: INC  DE
96AA  10 FD              DJNZ  CURVD2
96AC  B7                 OR    A
96AD  2A A299            LD    HL,(XMAX)
96B0  ED 52              SBC   HL,DE
96B2  30 19              JR    NC,CURVS1
96B4  ED 5B A299         LD    DE,(XMAX)
96B8  18 13              JR    CURVS1
96BA  C5           CURVS: PUSH  BC           ;Deplasare segment vertical din
96BB  ED 5B 94FF         LD    DE,(XC)      ;cursor IG spre stinga
96BF  1B          CURVS2: DEC  DE
96C0  10 FD              DJNZ  CURVS2
96C2  B7                 OR    A
96C3  2A A299            LD    HL,(XMAX)
96C6  ED 52              SBC   HL,DE
```

```
96C8  30 03                    JR    NC,CURVS1
96CA  11 0000                  LD    DE,0
96CD  CD 965C          CURVS1: CALL  CURV
96D0  ED 53 94FF               LD    (XC),DE
96D4  CD 965C                  CALL  CURV
96D7  C1                       POP   BC
96D8  C9                       RET
96D9  3E EF            TAS67:  LD    A,0EFH          ;Citirea tastelor 6 si 7 si
96DB  DB FE                    IN    A,(0FEH)        ;deplasarea corespunzatoare a
96DD  CB 67                    BIT   4,A             ;cursorului
96DF  20 03                    JR    NZ,TAS1
96E1  CD 9685                  CALL  CUROJ
96E4  3E EF            TAS1:   LD    A,0EFH
96E6  DB FE                    IN    A,(0FEH)
96E8  CB 5F                    BIT   3,A
96EA  20 03                    JR    NZ,TAS2
96EC  CD 9670                  CALL  CUROS
96EF  C9               TAS2:   RET
96F0  3E F7            TAS58:  LD    A,0F7H          ;Citirea tastelor 5 si 8 si
96F2  DB FE                    IN    A,(0FEH)        ;deplasarea corespunzatoare a
96F4  CB 67                    BIT   4,A             ;cursorului
96F6  20 03                    JR    NZ,TA1
96F8  CD 968A                  CALL  CURVS
96FB  3E EF            TA1:    LD    A,0EFH
96FD  DB FE                    IN    A,(0FEH)
96FF  CB 57                    BIT   2,A
9701  20 03                    JR    NZ,TA2
9703  CD 96A4                  CALL  CURVD
9706  C9               TA2:    RET


9707  1B 4C 80 02      AROB:   DB    1BH,"L",128,2
970B  00               AMARG:  DB    0
970C  20               C48:    DB    CARR/3*4
970D  18               C37:    DB    CARR
970E  80               BIT:    DB    80H
970F  01               NROCT:  DB    1
9710  00               SETSC:  DB    0

                       ;COPIE GRAFICA A ECRANULUI LA IMPRIMANTA

9711  3A FB90          ESCETB: LD    A,(ROLL)
9714  32 9205                 LD    (SCRO),A        ;Stabileste pozitia scroll hard
9717  CD 97F3                 CALL  ROSC11          ;Imprimanta SCAMP sau ROBOTRON ?
971A  28 0A                    JR    Z,CY11          ;salt pentru SCAMP
971C  0E 1B                    LD    C,1BH           ;ROBOTRON distanta intre rinduri
971E  CD 98A3                  CALL  LISTS1          ;de 7/72"
9721  0E 31                    LD    C,"1"
9723  CD 98A3                  CALL  LISTS1
9726  0E 20            CY11:   LD    C,20H
9728  3A 970B                  LD    A,(AMARG)       ;Numarul de spatii libere lasate
972B  47                       LD    B,A             ;in partea stinga la imprimare
972C  04                       INC   B
972D  05                       DEC   B
972E  28 0F                    JR    Z,CY13
9730  CD 98A3          CY15:   CALL  LISTS1
9733  10 FB                    DJNZ  CY15
9735  CD 97F3                  CALL  ROSC11
9738  20 05                    JR    NZ,CY13
973A  0E 0D                    LD    C,0DH           ;La SCAMP se emite un CR dupa
973C  CD 98A3                  CALL  LISTS1          ;terminarea caracterelor ASCII
```

```
973F  CD 97F3      CY13:  CALL  ROSC11      ;SCAMP sau ROBOTRON ?
9742  28 0E              JR    Z,CY18        ;salt pentru SCAMP
9744  21 9707            LD    HL,AROB       ;Comanda pentru intrare in regim
9747  06 04              LD    B,4           ;grafic la ROBOTRON
9749  4E           CY12:  LD    C,(HL)
974A  CD 98A3            CALL  LISTS1
974D  23                 INC   HL
974E  10 F9              DJNZ  CY12
9750  18 0A              JR    CY14
9752  0E 1B         CY18:  LD    C,1BH         ;Comanda pentru intrare in regim
9754  CD 98A3            CALL  LISTS1        ;grafic la SCAMP
9757  0E 48         CSCA:  LD    C,"H"
9759  CD 98A3            CALL  LISTS1
975C  DD 21 9207   CY14:  LD    IX,TAB1       ;Copierea zonei primelor 16
9760  06 08              LD    B,8           ;caractere alfanumerice
9762  CD 97DB            CALL  PARTE
9765  DD 21 9252         LD    IX,TAB2       ;Copierea zonei ultimelor 64
9769  06 20              LD    B,32          ;caractere alfanumerice
976B  CD 97DB            CALL  PARTE
976E  3E 01              LD    A,1           ;Initializeaza numarul
9770  32 970F            LD    (NROCT),A     ;octetului din linia TV
9773  3A 9710            LD    A,(SETSC)     ;Imprimarea s-a facut de pe doua
9776  B7                OR    A             ;rinduri alfanumerice succesive ?
9777  28 10              JR    Z,CY19
9779  AF                 XOR   A             ;DA, initializeaza indicatorul
977A  32 9710            LD    (SETSC),A
977D  3A 9205            LD    A,(SCRO)      ;Treci la rindul urmator
9780  3C                 INC   A
9781  FE 18              CP    CARR
9783  20 01              JR    NZ,CY16
9785  AF                 XOR   A
9786  32 9205    CY16:  LD    (SCRO),A
9789  CD 97F3    CY19:  CALL  ROSC11        ;SCAMP sau ROBOTRON ?
978C  F5                 PUSH  AF
978D  3A 970E            LD    A,(BIT)       ;Reface indicatorul bitului
9790  0F         CY120:  RRCA                ;ce urmeaza a fi copiat
9791  10 FD              DJNZ  CY120
9793  32 970E            LD    (BIT),A
9796  F1                 POP   AF
9797  28 26              JR    Z,CY17
9799  0E 0A              LD    C,0AH         ;Indicator sfirsit de linie
979B  CD 98A3            CALL  LISTS1        ;pentru ROBOTRON
979E  3A 970D            LD    A,(C37)       ;Treci la rindul urmator
97A1  3D                 DEC   A
97A2  32 970D            LD    (C37),A
97A5  C2 9726            JP    NZ,CY11
97A8  0E 1B              LD    C,1BH         ;Reface pentru ROBOTRON distanta
97AA  CD 98A3            CALL  LISTS1        ;intre rinduri de 1/8"
97AD  0E 30              LD    C,"0"
97AF  CD 98A3            CALL  LISTS1
97B2  3E 80              LD    A,80H
97B4  32 970E            LD    (BIT),A       ;Initializeaza indicatori pentru
97B7  3E 18              LD    A,CARR        ;o viitoare copie
97B9  32 970D            LD    (C37),A
97BC  C3 8980            JP    STESC
97BF  0E 2D        CY17:  LD    C,2DH         ;Comanda sirsit de rind pentru
97C1  CD 98A3            CALL  LISTS1        ;SCAMP
97C4  3A 970C            LD    A,(C48)       ;Treci la rindul urmator
97C7  3D                 DEC   A
97C8  32 970C            LD    (C48),A
97CB  C2 9726            JP    NZ,CY11
```

```
97CE  3E 80          LD    A,80H          ;Initializeaza indicatori pentru
97D0  32 970E        LD    (BIT),A        ;o noua copie
97D3  3E 20          LD    A,CARR/3*4
97D5  32 970C        LD    (C48),A
97D8  C3 8980        JP    STESC
                   ;Copierea primelor 16 sau a ultimelor 64 caractere alfanumerice
97DB  C5           PARTE: PUSH  BC
97DC  CD 8BD0        CALL  PAR
97DF  CD 9800        CALL  OCTET
97E2  CD 8BC6        CALL  IMPAR
97E5  CD 9800        CALL  OCTET
97E8  C1             POP   BC
97E9  3A 970F        LD    A,(NROCT)      ;Treci la octetul urmator
97EC  3C             INC   A
97ED  32 970F        LD    (NROCT),A
97F0  10 E9          DJNZ  PARTE
97F2  C9             RET
                   ;SCAMP sau ROBOTRON ?
97F3  3A 91D5      ROSC11: LD   A,(ML)
97F6  CB 6F          BIT   5,A
97F8  28 03          JR    Z,RS1
97FA  06 08          LD    B,8            ;ROBOTRON
97FC  C9             RET
97FD  06 06        RS1:  LD    B,6            ;SCAMP
97FF  C9             RET
                   ;Copierea unui cimp corespunzator unui caracter alfanumeric
9800  16 80        OCTET: LD   D,80H          ;Indicator octet
9802  CD 97F3      OC10:  CALL ROSC11         ;SCAMP sau ROBOTRON ?
9805  0E 00             LD    C,0            ;Octet trimis la imprimanta
9807  28 04             JR    Z,OC20
9809  1E 80             LD    E,80H          ;Indicator bit pentru ROBOTRON
980B  18 02             JR    OC21
980D  1E 01        OC20:  LD    E,01H          ;Indicator bit pentru SCAMP
980F  3A 9205      OC21:  LD    A,(SCR0)
9812  CB 27        OC7:   SLA   A              ;Determinarea adresei rindului
9814  32 981D             LD    (OC1+2),A      ;din care face parte bitul
9817  3C                  INC   A              ;imprimat
9818  32 9820             LD    (OC2+2),A
981B  DD 6E 01     OC1:   LD    L,(IX+1)       ;In HL adresa rindului
981E  DD 66 01     OC2:   LD    H,(IX+1)       ;alfanumeric din care face parte
9821  F5                   PUSH  AF             ;bitul imprimat
9822  3A 970F              LD    A,(NROCT)      ;Determinarea caracterului in
9825  FE 09                CP    9              ;cadrul rindului
9827  38 02                JR    C,OC9
9829  D6 08                SUB   8
982B  3D           OC9:    DEC   A
982C  28 03                JR    Z,OC8
982E  23                   INC   HL             ;In HL adresa caracterului
982F  18 FA                JR    OC9            ;alfanumeric din care face parte
9831  3A 970E      OC8:    LD    A,(BIT)        ;bitul imprimat
9834  07           OC16:   RLCA                 ;Determinarea adresei rindului TV
9835  38 03                JR    C,OC15         ;in cadrul caracterului
9837  24                   INC   H              ;In HL adresa octetului din care
9838  18 FA                JR    OC16           ;face parte bitul imprimat
983A  F1           OC15:   POP   AF
983B  C5                   PUSH  BC             ;Deschide zona RAM VIDEO necesara
983C  FE 2F                CP    47
983E  30 05                JR    NC,OC3
9840  CD 91E2              CALL  ZONA1          ;Zona rindurilor 1-24
9843  18 0C                JR    OC4
9845  FE 3F        OC3:    CP    63
```

```
9847  30 05              JR    NC,0C5
9849  CD 91EC            CALL  ZONA2       ;Zona rindurilor 25-32
984C  18 03              JR    0C4
984E  CD 91F8      0C5:  CALL  ZONA3       ;Zona rindurilor 33-36
9851  C1          0C4:  POP   BC
9852  7A          0C11: LD    A,D         ;Stabileste pozitia bitului in
                                          ;cadrul octetului
9853  A6                AND   (HL)        ;Se testeaza bitul dorit
9854  28 03             JR    Z,0C12
9856  7B                LD    A,E         ;Se pozitioneaza bitul ce urmeza
9857  B1                OR    C           ;sa fie trimis la imprimanta
9858  4F                LD    C,A
9859  C5          0C12: PUSH  BC
985A  CD 97F3           CALL  ROSC11
985D  28 04             JR    Z,0C22
985F  CB 08             RRC   E           ;Treci la bitul urmator pentru
9861  18 02             JR    0C23        ;ROBOTRON
9863  CB 03       0C22: RLC   E           ;Treci la bitul urmator pentru
9865  C1          0C23: POP   BC          ;SCAMP
9866  3A 970E           LD    A,(BIT)     ;Treci la rindul TV urmator
9869  0F                RRCA
986A  32 970E           LD    (BIT),A
986D  30 15             JR    NC,0C6
986F  3A 9205           LD    A,(SCRO)    ;Daca s-au terminat rindurile
9872  3C                INC   A           ;TV dintr-un caracter, se trece
9873  FE 18             CP    CARR        ;la urmatorul rind  alfanumeric
9875  20 01             JR    NZ,0C18
9877  AF                XOR   A
9878  F5          0C18: PUSH  AF
9879  AF                XOR   A
987A  3D                DEC   A
987B  32 9710           LD    (SETSC),A   ;Se incarca 0FFH in indicatorul
987E  F1                POP   AF          ;SETSC in cazul trecerii la
987F  05                DEC   B           ;urmatorul rind alfanumeric
9880  28 05             JR    Z,0C17
9882  18 8E             JR    0C7
9884  24          0C6:  INC   H           ;Se tece la urmatorul rind TV
9885  10 CB             DJNZ  0C11        ;pentru a prelua urmatorul bit
9887  CD 97F3     0C17: CALL  ROSC11      ;Au fost cititi 6 biti pentru
988A  20 02             JR    NZ,0C19     ;SCAMP sau 8 biti pentru ROBOTRON
988C  CB F1             SET   6,C         ;Pentru SCAMP se seteaza bitul 6
988E  CD 98A3     0C19: CALL  LISTS1      ;Caracterul este emis catre
9891  CD 97F3           CALL  ROSC11      ;imprimanta
9894  3A 970E           LD    A,(BIT)     ;Se reface indicatorul BIT
9897  07          0C13: RLCA
9898  10 FD             DJNZ  0C13
989A  32 970E           LD    (BIT),A
989D  CB 0A             RRC   D           ;Se trece la urmatorul octet in
                                          ;cadrul caracterului
989F  D8                RET   C           ;Iesire din subrutina dupa 8
98A0  C3 9802           JP    0C10        ;cuvinte emise la imprimanta
                ;Subrutina pentru salvarea registrelor in cazul emisiei
                ;spre imprimanta
98A3  C5          LISTS1: PUSH  BC
98A4  D5                  PUSH  DE
98A5  E5                  PUSH  HL
98A6  3A 89C5             LD    A,(DIOBYT)
98A9  CD 8517             CALL  LIST1X
98AC  E1                  POP   HL
98AD  D1                  POP   DE
98AE  C1                  POP   BC
```

```
98AF   C9                        RET

                                 ;PROGRAMAREA INTERFETELOR SERIALA SI PARALELA DE IESIRE

98B0   CD 9001                   ESCACK: CALL    TYPE
98B3   20 20 53 74                       DB      "  Stabilirea tipului interfetei:",0AH,0DH
98B7   61 62 69 6C
98BB   69 72 65 61
98BF   20 74 69 70
98C3   75 6C 75 69
98C7   20 69 6E 74
98CB   65 72 66 65
98CF   74 65 69 3A
98D3   0A 0D
98D5   09 09 31 2D                       DB                      1- interfata serie asincrona "
98D9   20 69 6E 74
98DD   65 72 66 61
98E1   74 61 20 73
98E5   65 72 69 65
98E9   20 61 73 69
98ED   6E 63 72 6F
98F1   6E 61 20
98F4   70 72 6F 74                       DB      "protocol DTR",0AH,0DH
98F8   6F 63 6F 6C
98FC   20 44 54 52
9900   0A 0D
9902   09 09 32 2D                       DB                      2- interfata paralela de iesire"
9906   20 69 6E 74
990A   65 72 66 61
990E   74 61 20 70
9912   61 72 61 6C
9916   65 6C 61 20
991A   64 65 20 69
991E   65 73 69 72
9922   65
9923   70 72 6F 74                       DB      "protocol BUSY, STROB",0AH,0DH,24H
9927   6F 63 6F 6C
992B   20 42 55 53
992F   59 2C 20 53
9933   54 52 4F 42
9937   0A 0D 24
993A   CD 851E                   EQ3:    CALL    CONIN1
993D   FE 31                             CP      31H
993F   28 07                             JR      Z,EQ1
9941   FE 32                             CP      32H
9943   CA 9C0D                           JP      Z,EQ2
9946   18 F2                             JR      EQ3
9948   21 0003                 EQ1:      LD      HL,3
994B   CB BE                             RES     7,(HL)
994D   CB B6                             RES     6,(HL)
994F   CD 8001                           CALL    TYPE
9952   20 20 53 65                       DB      "  Serie- rata de transfer:1-  300 bauds",0AH,0DH
9956   72 69 65 2D
995A   20 72 61 74
995E   61 20 64 65
9962   20 74 72 61
9966   6E 73 66 65
996A   72 3A 31 2D
996E   20 20 33 30
9972   30 20 62 61
9976   75 64 73 0A
```

```
997A 0D
997B 09 09 09 20                DB      "                       2-  600 bauds",0AH,0DH
997F 20 32 2D 20
9983 20 36 30 30
9987 20 62 61 75
998B 64 73 0A 0D
998F 09 09 09 20                DB      "                       3- 1200 bauds",0AH,0DH
9993 20 33 2D 20
9997 31 32 30 30
999B 20 62 61 75
999F 64 73 0A 0D
99A3 09 09 09 20                DB      "                       4- 2400 bauds",0AH,0DH
99A7 20 34 2D 20
99AB 32 34 30 30
99AF 20 62 61 75
99B3 64 73 0A 0D
99B7 09 09 09 20                DB      "                       5- 4800 bauds",0AH,0DH
99BB 20 35 2D 20
99BF 34 38 30 30
99C3 20 62 61 75
99C7 64 73 0A 0D
99CB 09 09 09 20                DB      "                       6- 9600 bauds",0AH,0DH
99CF 20 36 2D 20
99D3 39 36 30 30
99D7 20 62 61 75
99DB 64 73 0A 0D
99DF 24                         DB      24H
99E0 CD 851E            EQ10:   CALL    CONIN1
99E3 1E 0D                      LD      E,0DH
99E5 FE 31                      CP      31H
99E7 28 16                      JR      Z,EQ4
99E9 FE 32                      CP      32H
99EB 28 14                      JR      Z,EQ5
99ED FE 33                      CP      33H
99EF 28 12                      JR      Z,EQ6
99F1 FE 34                      CP      34H
99F3 28 10                      JR      Z,EQ7
99F5 FE 35                      CP      35H
99F7 28 0E                      JR      Z,EQ8
99F9 FE 36                      CP      36H
99FB 28 0C                      JR      Z,EQ9
99FD 18 E1                      JR      EQ10
99FF CB 23              EQ4:    SLA     E
9A01 CB 23              EQ5:    SLA     E
9A03 CB 23              EQ6:    SLA     E
9A05 CB 23              EQ7:    SLA     E
9A07 CB 23              EQ8:    SLA     E
9A09 CD A016            EQ9:    CALL    PR8253
9A0C 11 FE35                    LD      DE,0FE35H
9A0F D5                         PUSH    DE
9A10 CD 8001            EQ11:   CALL    TYPE
9A13 20 20 53 65                DB      "  Serie- control de paritate:",0AH,0DH
9A17 72 69 65 2D
9A1B 20 63 6F 6E
9A1F 74 72 6F 6C
9A23 20 64 65 20
9A27 70 61 72 69
9A2B 74 61 74 65
9A2F 3A 0A 0D
9A32 09 09 31 2D                DB                      1- cu control de paritate"
9A36 20 63 75 20
```

```
9A3A  63 6F 6E 74
9A3E  72 6F 6C 20
9A42  64 65 20 70
9A46  61 72 69 74
9A4A  61 74 65
9A4D  0D 0A                 DB    0DH,0AH
9A4F  09 09 32 2D           DB    "       2- fara control de paritate"
9A53  20 66 61 72
9A57  61 20 63 6F
9A5B  6E 74 72 6F
9A5F  6C 20 64 65
9A63  20 70 61 72
9A67  69 74 61 74
9A6B  65
9A6C  0A 0D 24              DB    0AH,0DH,24H
9A6F  CD 851E         EQ14: CALL  CONIN1
9A72  FE 31                 CP    31H
9A74  28 06                 JR    Z,EQ12
9A76  FE 32                 CP    32H
9A78  28 5B                 JR    Z,EQ13
9A7A  18 F3                 JR    EQ14
9A7C  CD 8001         EQ12: CALL  TYPE
9A7F  20 20 53 65           DB    "  Serie- cu control de paritate:",0AH,0DH
9A83  72 69 65 2D
9A87  20 63 75 20
9A8B  63 6F 6E 74
9A8F  72 6F 6C 20
9A93  64 65 20 70
9A97  61 72 69 74
9A9B  61 74 65 3A
9A9F  0A 0D
9AA1  09 09 31 2D           DB    "       1- para",0AH,0DH
9AA5  20 70 61 72
9AA9  61 0A 0D
9AAC  09 09 32 2D           DB    "       2- impara",0AH,0DH,24H
9AB0  20 69 6D 70
9AB4  61 72 61 0A
9AB8  0D 24
9ABA  CD 851E         EQ17: CALL  CONIN1
9ABD  FE 31                 CP    31H
9ABF  28 06                 JR    Z,EQ15
9AC1  FE 32                 CP    32H
9AC3  28 09                 JR    Z,EQ16
9AC5  18 F3                 JR    EQ17
9AC7  D1              EQ15: POP   DE
9AC8  CB E2                 SET   4,D
9ACA  CB AA                 RES   5,D
9ACC  18 0A                 JR    EQ18
9ACE  D1              EQ16: POP   DE
9ACF  CB E2                 SET   4,D
9AD1  CB EA                 SET   5,D
9AD3  18 03                 JR    EQ18
9AD5  D1              EQ13: POP   DE
9AD6  CB A2                 RES   4,D
9AD8  D5              EQ18: PUSH  DE
9AD9  CD 8001               CALL  TYPE
9ADC  20 20 4E 75           DB    "  Numarul de biti pe caracter:",0AH,0DH
9AE0  6D 61 72 75
9AE4  6C 20 64 65
9AE8  20 62 69 74
9AEC  69 20 70 65
```

```
9AF0  20 63 61 72
9AF4  61 63 74 65
9AF8  72 3A 0A 0D
9AFC  09 09 31 2D                DB    "              1- 5 biti",0AH,0DH
9B00  20 35 20 62
9B04  69 74 69 0A
9B08  0D
9B09  09 09 32 2D                DB    "              2- 6 biti",0AH,0DH
9B0D  20 36 20 62
9B11  69 74 69 0A
9B15  0D
9B16  09 09 33 2D                DB    "              3- 7 biti",0AH,0DH
9B1A  20 37 20 62
9B1E  69 74 69 0A
9B22  0D
9B23  09 09 34 2D                DB    "              4- 8 biti",0AH,0DH,24H
9B27  20 38 20 62
9B2B  69 74 69 0A
9B2F  0D 24
9B31  CD 851E          EQ23:     CALL  CONIN1
9B34  FE 31                      CP    31H
9B36  28 0E                      JR    Z,EQ19
9B38  FE 32                      CP    32H
9B3A  28 11                      JR    Z,EQ20
9B3C  FE 33                      CP    33H
9B3E  28 14                      JR    Z,EQ21
9B40  FE 34                      CP    34H
9B42  28 17                      JR    Z,EQ22
9B44  18 EB                      JR    EQ23
9B46  D1               EQ19:     POP   DE
9B47  CB 92                      RES   2,D
9B49  CB 9A                      RES   3,D
9B4B  18 13                      JR    EQ24
9B4D  D1               EQ20:     POP   DE
9B4E  CB D2                      SET   2,D
9B50  CB 9A                      RES   3,D
9B52  18 0C                      JR    EQ24
9B54  D1               EQ21:     POP   DE
9B55  CB 92                      RES   2,D
9B57  CB DA                      SET   3,D
9B59  18 05                      JR    EQ24
9B5B  D1               EQ22:     POP   DE
9B5C  CB D2                      SET   2,D
9B5E  CB DA                      SET   3,D
9B60  D5               EQ24:     PUSH  DE
9B61  CD 8001                    CALL  TYPE
9B64  20 20 53 65                DB    "  Serie- numarul bitilor de stop:",0AH,0DH
9B68  72 69 65 2D
9B6C  20 6E 75 6D
9B70  61 72 75 6C
9B74  20 62 69 74
9B78  69 6C 6F 72
9B7C  20 64 65 20
9B80  73 74 6F 70
9B84  3A 0A 0D
9B87  09 09 31 2D                DB    "              1- invalid",0AH,0DH
9B8B  20 69 6E 76
9B8F  61 6C 69 64
9B93  0A 0D
9B95  09 09 32 2D                DB    "              2- i bit de stop",0AH,0DH
9B99  20 31 20 62
```

```
9B9D  69 74 20 64
9BA1  65 20 73 74
9BA5  6F 70 0A 0D
9BA9  09 09 33 2D          DB    "          3- 1+1/2 biti de stop",0AH,0DH
9BAD  20 31 2B 31
9BB1  2F 32 20 62
9BB5  69 74 69 20
9BB9  64 65 20 73
9BBD  74 6F 70 0A
9BC1  0D
9BC2  09 09 34 2D          DB    "          4- 2 biti de stop",0AH,0DH,24H
9BC6  20 32 20 62
9BCA  69 74 69 20
9BCE  64 65 20 73
9BD2  74 6F 70 0A
9BD6  0D 24
9BD8  CD 851E        EQ29: CALL  CONIN1
9BDB  FE 31                CP    31H
9BDD  28 0E                JR    Z,EQ25
9BDF  FE 32                CP    32H
9BE1  28 11                JR    Z,EQ26
9BE3  FE 33                CP    33H
9BE5  28 14                JR    Z,EQ27
9BE7  FE 34                CP    34H
9BE9  28 17                JR    Z,EQ28
9BEB  18 EB                JR    EQ29
9BED  D1             EQ25: POP   DE
9BEE  CB B2                RES   6,D
9BF0  CB BA                RES   7,D
9BF2  18 13                JR    EQ30
9BF4  D1             EQ26: POP   DE
9BF5  CB F2                SET   6,D
9BF7  CB BA                RES   7,D
9BF9  18 0C                JR    EQ30
9BFB  D1             EQ27: POP   DE
9BFC  CB B2                RES   6,D
9BFE  CB FA                SET   7,D
9C00  18 05                JR    EQ30
9C02  D1             EQ28: POP   DE
9C03  CB F2                SET   6,D
9C05  CB FA                SET   7,D
9C07  CD A022        EQ30: CALL  PR8251
9C0A  C3 9CBC              JP    EQ31
9C0D  21 0003        EQ2:  LD    HL,3
9C10  CB FE                SET   7,(HL)
9C12  CB B6                RES   6,(HL)
9C14  CD 8001              CALL  TYPE
9C17  20 20 50 61          DB    "  Paralel- BUSY activ: 1- Hi",0AH,0DH
9C1B  72 61 6C 65
9C1F  6C 2D 20 42
9C23  55 53 59 20
9C27  61 63 74 69
9C2B  76 3A 20 31
9C2F  2D 20 48 69
9C33  0A 0D
9C35  09 09 20 20          DB    "          2- Lo",0AH,0DH,24H
9C39  20 20 20 20
9C3D  20 32 2D 20
9C41  4C 6F 0A 0D
9C45  24
9C46  CD 851E        EQ34: CALL  CONIN1
```

```
9C49  FE 31              CP    31H
9C4B  28 06              JR    Z,EQ32
9C4D  FE 32              CP    32H
9C4F  28 09              JR    Z,EQ33
9C51  18 F3              JR    EQ34
9C53  3E 38       EQ32:  LD    A,38H
9C55  32 84A9            LD    (LPTB),A
9C58  18 05              JR    EQ35
9C5A  3E 30       EQ33:  LD    A,30H
9C5C  32 84A9            LD    (LPTB),A
9C5F  CD 8001     EQ35:  CALL  TYPE
9C62  20 20 50 61        DB    "  Paralel- STROB activ: 1- Hi",0AH,0DH
9C66  72 61 6C 65
9C6A  6C 2D 20 53
9C6E  54 52 4F 42
9C72  20 61 63 74
9C76  69 76 3A 20
9C7A  31 2D 20 48
9C7E  69 0A 0D
9C81  09 09 20 20        DB    "                    2- Lo",0AH,0DH,24H
9C85  20 20 20 20
9C89  20 20 32 2D
9C8D  20 4C 6F 0A
9C91  0D 24
9C93  CD 851E     EQ38:  CALL  CONIN1
9C96  FE 31              CP    31H
9C98  28 06              JR    Z,EQ36
9C9A  FE 32              CP    32H
9C9C  28 11              JR    Z,EQ37
9C9E  18 F3              JR    EQ38
9CA0  3E E7       EQ36:  LD    A,0E7H
9CA2  32 84BE            LD    (LPTS1+1),A
9CA5  32 84C6            LD    (LPTS3+1),A
9CA8  3E A7              LD    A,0A7H
9CAA  32 84C2            LD    (LPTS2+1),A
9CAD  18 0D              JR    EQ31
9CAF  3E A7       EQ37:  LD    A,0A7H
9CB1  32 84BE            LD    (LPTS1+1),A
9CB4  32 84C6            LD    (LPTS3+1),A
9CB7  3E E7              LD    A,0E7H
9CB9  32 84C2            LD    (LPTS2+1),A
9CBC  CD 8001     EQ31:  CALL  TYPE
9CBF  20 20 43 61        DB    "  Caracteristici pentru copia grafica a "
9CC3  72 61 63 74
9CC7  65 72 69 73
9CCB  74 69 63 69
9CCF  20 70 65 6E
9CD3  74 72 75 20
9CD7  63 6F 70 69
9CDB  61 20 67 72
9CDF  61 66 69 63
9CE3  61 20 61 20
9CE7  65 63 72 61        DB    "ecranului la imprimanta",0AH,0DH
9CEB  6E 75 6C 75
9CEF  69 20 6C 61
9CF3  20 69 6D 70
9CF7  72 69 6D 61
9CFB  6E 74 61 0A
9CFF  0D
9D00  09 09 31 2D        DB    "                1- DA",0AH,0DH
9D04  20 44 41 0A
```

```
9D08  0D
9D09  09 09 32 2D                DB    "          2- NU",0AH,0DH,24H
9D0D  20 4E 55 0A
9D11  0D 24
9D13  CD 851E              EQ42: CALL  CONIN1
9D16  FE 31                      CP    31H
9D18  28 09                      JR    Z,EQ40
9D1A  FE 32                      CP    32H
9D1C  28 02                      JR    Z,EQ41
9D1E  18 F3                      JR    EQ42
9D20  C3 8980              EQ41: JP    STESC
9D23  CD 8001              EQ40: CALL  TYPE
9D26  20 20 54 69                DB    "  Tipul imprimantei: 1- SCAMP 9335",0AH,0DH
9D2A  70 75 6C 20
9D2E  69 6D 70 72
9D32  69 6D 61 6E
9D36  74 65 69 3A
9D3A  20 31 2D 20
9D3E  53 43 41 4D
9D42  50 20 39 33
9D46  33 35 0A 0D
9D4A  09 09 20 20                DB    "                2- ROBOTRON 6313, 6314"
9D4E  20 20 20 32
9D52  2D 20 52 4F
9D56  42 4F 54 52
9D5A  4F 4E 20 36
9D5E  33 31 33 2C
9D62  20 36 33 31
9D66  34
9D67  0A 0D                      DB    0AH,0DH
9D69  09 09 20 20                DB    "                3- ROBOTRON 6311, 6312"
9D6D  20 20 20 33
9D71  2D 20 52 4F
9D75  42 4F 54 52
9D79  4F 4E 20 36
9D7D  33 31 31 2C
9D81  20 36 33 31
9D85  32
9D86  20 73 61 75                DB    " sau ROMOM",0AH,0DH,24H
9D8A  20 52 4F 4D
9D8E  4F 4D 0A 0D
9D92  24
9D93  CD 851E              EQ46: CALL  CONIN1
9D96  21 91D5                    LD    HL,ML
9D99  FE 31                      CP    31H
9D9B  28 0A                      JR    Z,EQ43
9D9D  FE 32                      CP    32H
9D9F  28 0A                      JR    Z,EQ44
9DA1  FE 33                      CP    33H
9DA3  28 0B                      JR    Z,EQ45
9DA5  18 EC                      JR    EQ46
9DA7  CB AE                EQ43: RES   5,(HL)
9DA9  18 0F                      JR    EQ47
9DAB  CB EE                EQ44: SET   5,(HL)
9DAD  C3 9E48                    JP    EQ48
9DB0  CB EE                EQ45: SET   5,(HL)
9DB2  3E 4B                      LD    A,"K"
9DB4  32 9708                    LD    (AROB+1),A
9DB7  C3 9F23                    JP    EQ50
9DBA  CD 8001              EQ47: CALL  TYPE
9DBD  20 20 49 6D                DB    "  Imprimanta SCAMP- densitate grafica: 1-  72 "
```

```
9DC1  70 72 69 6D
9DC5  61 6E 74 61
9DC9  20 53 43 41
9DCD  4D 50 2D 20
9DD1  64 65 6E 73
9DD5  69 74 61 74
9DD9  65 20 67 72
9DDD  61 66 69 63
9DE1  61 3A 20 31
9DE5  2D 20 20 37
9DE9  32 20
9DEB  64 70 69 0A                 DB      "dpi",0AH,0DH
9DEF  0D
9DF0  09 09 09 09                 DB      "                                   2- 120 "
9DF4  20 20 20 20
9DF8  20 20 20 32
9DFC  2D 20 31 32
9E00  30 20
9E02  64 70 69 0A                 DB      "dpi",0AH,0DH
9E06  0D
9E07  09 09 09 09                 DB      "                                   3- 144 "
9E0B  20 20 20 20
9E0F  20 20 20 33
9E13  2D 20 31 34
9E17  34 20
9E19  64 70 69 0A                 DB      "dpi",0AH,0DH,24H
9E1D  0D 24
9E1F  CD 851E             EQ54:   CALL    CONIN1
9E22  FE 31                       CP      31H
9E24  28 0A                       JR      Z,EQ51
9E26  FE 32                       CP      32H
9E28  28 0E                       JR      Z,EQ52
9E2A  FE 33                       CP      33H
9E2C  28 12                       JR      Z,EQ53
9E2E  18 EF                       JR      EQ54
9E30  3E 47               EQ51:   LD      A,"G"
9E32  32 9758                     LD      (CSCA+1),A
9E35  C3 9F23                     JP      EQ50
9E38  3E 48               EQ52:   LD      A,"H"
9E3A  32 9758                     LD      (CSCA+1),A
9E3D  C3 9F23                     JP      EQ50
9E40  3E 49               EQ53:   LD      A,"I"
9E42  32 9758                     LD      (CSCA+1),A
9E45  C3 9F23                     JP      EQ50
9E48  CD 8001             EQ48:   CALL    TYPE
9E4B  20 20 49 6D                 DB      "  Imprimanta ROBOTRON 6313, 6314- densitate "
9E4F  70 72 69 6D
9E53  61 6E 74 61
9E57  20 52 4F 42
9E5B  4F 54 52 4F
9E5F  4E 20 36 33
9E63  31 33 2C 20
9E67  36 33 31 34
9E6B  2D 20 64 65
9E6F  6E 73 69 74
9E73  61 74 65 20
9E77  67 72 61 66                 DB      "grafica:",0AH,0DH
9E7B  69 63 61 3A
9E7F  0A 0D
9E81  09 09 31 2D                 DB      "                1-  480 P/8 toli",0AH,0DH
9E85  20 20 34 38
```

```
9E89  30 20 50 2F
9E8D  38 20 74 6F
9E91  6C 69 0A 0D
9E95  09 09 32 2D          DB    "              2-  960 p/8 toli viteza  6 "
9E99  20 20 39 36
9E9D  30 20 70 2F
9EA1  38 20 74 6F
9EA5  6C 69 20 76
9EA9  69 74 65 7A
9EAD  61 20 20 36
9EB1  20
9EB2  74 6F 6C 69          DB    "toli/s",0AH,0DH
9EB6  2F 73 0A 0D
9EBA  09 09 33 2D          DB    "              3-  960 p/8 toli viteza 10 "
9EBE  20 20 39 36
9EC2  30 20 70 2F
9EC6  38 20 74 6F
9ECA  6C 69 20 76
9ECE  69 74 65 7A
9ED2  61 20 31 30
9ED6  20
9ED7  74 6F 6C 69          DB    "toli/s",0AH,0DH
9EDB  2F 73 0A 0D
9EDF  09 09 34 2D          DB    "              4- 1200 p/8 toli",0AH,0DH,24H
9EE3  20 31 32 30
9EE7  30 20 70 2F
9EEB  38 20 74 6F
9EEF  6C 69 0A 0D
9EF3  24
9EF4  CD 851E        EQ59: CALL  CONIN1
9EF7  FE 31                CP    31H
9EF9  28 0E                JR    Z,EQ55
9EFB  FE 32                CP    32H
9EFD  28 11                JR    Z,EQ56
9EFF  FE 33                CP    33H
9F01  28 14                JR    Z,EQ57
9F03  FE 34                CP    34H
9F05  28 17                JR    Z,EQ58
9F07  18 EB                JR    EQ59
9F09  3E 4B          EQ55: LD    A,"K"
9F0B  32 9708              LD    (AROB+1),A
9F0E  18 13                JR    EQ50
9F10  3E 4C          EQ56: LD    A,"L"
9F12  32 9708              LD    (AROB+1),A
9F15  18 0C                JR    EQ50
9F17  3E 59          EQ57: LD    A,"Y"
9F19  32 9708              LD    (AROB+1),A
9F1C  18 05                JR    EQ50
9F1E  3E 5A          EQ58: LD    A,"Z"
9F20  32 9708              LD    (AROB+1),A
9F23  CD 8001        EQ50: CALL  TYPE
9F26  20 20 4E 75          DB    "  Numarul de blancuri lasate in partea stinga"
9F2A  6D 61 72 75
9F2E  6C 20 64 65
9F32  20 62 6C 61
9F36  6E 63 75 72
9F3A  69 20 6C 61
9F3E  73 61 74 65
9F42  20 69 6E 20
9F46  70 61 72 74
9F4A  65 61 20 73
```

```
9F4E  74 69 6E 67
9F52  61
9F53  0A 0D                   DB    0AH,0DH
9F55  20 20 20 2D             DB    "   -Numarul va fi in baza 16, max. 0FFH",0AH,0DH
9F59  4E 75 6D 61
9F5D  72 75 6C 20
9F61  76 61 20 66
9F65  69 20 69 6E
9F69  20 62 61 7A
9F6D  61 20 31 36
9F71  2C 20 6D 61
9F75  78 2E 20 30
9F79  46 46 48 0A
9F7D  0D
9F7E  20 20 20 2D             DB    "   -Se tasteaza numarul dupa care se actioneaza"
9F82  53 65 20 74
9F86  61 73 74 65
9F8A  61 7A 61 20
9F8E  6E 75 6D 61
9F92  72 75 6C 20
9F96  64 75 70 61
9F9A  20 63 61 72
9F9E  65 20 73 65
9FA2  20 61 63 74
9FA6  69 6F 6E 65
9FAA  61 7A 61
9FAD  20 74 61 73             DB    " tasta ENTER",0AH,0DH,24H
9FB1  74 61 20 45
9FB5  4E 54 45 52
9FB9  0A 0D 24
9FBC  11 0000                 LD    DE,0
9FBF  D5                EQ60: PUSH  DE
9FC0  CD 851E                 CALL  CONIN1
9FC3  D1                      POP   DE
9FC4  FE 0D                   CP    0DH
9FC6  28 18                   JR    Z,EQ61
9FC8  FE 30                   CP    30H
9FCA  FA 9FBF                 JP    M,EQ60
9FCD  FE 47                   CP    47H
9FCF  F2 9FBF                 JP    P,EQ60
9FD2  FE 40                   CP    40H
9FD4  28 E9                   JR    Z,EQ60
9FD6  53                      LD    D,E
9FD7  5F                      LD    E,A
9FD8  4F                      LD    C,A
9FD9  D5                      PUSH  DE
9FDA  CD 8523                 CALL  CONOUT1
9FDD  D1                      POP   DE
9FDE  18 DF                   JR    EQ60
9FE0  7A                EQ61: LD    A,D
9FE1  CD A00A                 CALL  CAZ
9FE4  4F                      LD    C,A
9FE5  7B                      LD    A,E
9FE6  CD A00A                 CALL  CAZ
9FE9  06 04                   LD    B,4
9FEB  CB 21             EQ71: SLA   C
9FED  10 FC                   DJNZ  EQ71
9FEF  B1                      OR    C
9FF0  32 970B                 LD    (AMARG),A
9FF3  7A                      LD    A,D
9FF4  32 A001                 LD    (MESC+3),A
```

```
9FF7  7B                  LD    A,E
9FF8  32 A002             LB    (MESC+4),A
9FFB  CD 8001             CALL  TYPE
9FFE  0A 0D 30 30 MESC:   DB    0AH,0DH,30H,30H,30H,"H",0AH,0DH,24H
A002  30 48 0A 0D
A006  24
A007  C3 8980             JP    STESC
A00A  FE 40       CAZ:    CP    40H
A00C  30 03               JR    NC,CAZ1
A00E  E6 0F               AND   0FH
A010  C9                  RET
A011  E6 0F       CAZ1:   AND   0FH
A013  C6 09               ADD   A,9
A015  C9                  RET
A016  01 9FFD     PR8253: LD    BC,9FFDH
A019  3E 1E               LD    A,1EH
A01B  ED 79               OUT   (C),A
A01D  06 9C               LD    B,9CH
A01F  ED 59               OUT   (C),E
A021  C9                  RET
A022  01 DFFD     PR8251: LD    BC,0DFFDH
A025  3E 40               LD    A,40H
A027  ED 79               OUT   (C),A
A029  ED 51               OUT   (C),D
A02B  ED 59               OUT   (C),E
A02D  05                  DEC   B
A02E  ED 78               IN    A,(C)
A030  C9                  RET


                          ;COMENZI SPECIALE

A031  21 91D5     ESCETX: LD    HL,ML
A034  CB C6               SET   0,(HL)      ;Seteaza modul de lucru comenzi
A036  C3 8980             JP    STESC       ;speciale
A039  79          CDSPEC: LD    A,C
A03A  21 91D5             LD    HL,ML
A03D  CB 5E               BIT   3,(HL)      ;Este setat modul de lucru pentru
                                            ;indicator tip dreapta ?
A03F  20 74               JR    NZ,CDS1     ;DA, salt
A041  CB 66               BIT   4,(HL)      ;Este setat modul de lucru pentru
                                            ;indicator numar caractere
                                            ;alfanumerice?
A043  20 25               JR    NZ,CDS2     ;DA, salt
A045  FE 01               CP    1           ;Comanda pentru set tip dreapta ?
A047  28 1D               JR    Z,CDS3      ;DA, salt
A049  FE 02               CP    2           ;Comanda pentru set numar
                                            ;caractere alfanumerice?
A04B  28 15               JR    Z,CDS4      ;DA, salt
A04D  FE 53               CP    53H         ;Comanda pentru set copie la
                                            ;imprimanta 1/1?
A04F  28 0D               JR    Z,CDS5      ;DA, salt
A051  FE 44               CP    44H         ;Comanda pentru set copie la
                                            ;imprim 2/1?
A053  28 05               JR    Z,CDS6      ;DA, salt
A055  CB 86       CDS7:   RES   0,(HL)      ;Ies din modul de lucru
                                            ;comanda speciala
A057  C3 8980     CDS8:   JP    STESC       ;Iesire din subrutina
A05A  CB F6       CDS6:   SET   6,(HL)      ;Set copie 2/1
A05C  18 F7               JR    CDS7
A05E  CB B6       CDS5:   RES   6,(HL)      ;Set copie 1/1
```

```
A060  18 F3            JR    CDS7
A062  CB E6     CDS4:  SET   4,(HL)      ;Set numar caractere alfanumerice
A064  18 F1            JR    CDS8
A066  CB DE     CDS3:  SET   3,(HL)      ;Set tip drepta
A068  18 ED            JR    CDS8
A06A  E6 7E     CDS2:  AND   7EH         ;Numar de caractere alfanumerice
A06C  FE 00            CP    0           ;pe un rind indicat de utilizator
A06E  28 04            JR    Z,CDS13     ;este preluat si introdus in
A070  FE 4F            CP    79          ;secventa de program ce
A072  38 08            JR    C,CDS11     ;realizeaza afisarea pe monitor
A074  AF        CDS13: XOR   A
A075  32 91D6          LD    (NCR),A
A078  3E 50            LD    A,80
A07A  18 03            JR    CDS12
A07C  32 91D6   CDS11: LD    (NCR),A
A07F  32 8868   CDS12: LD    (AD1+1),A
A082  32 88E1          LD    (AD2+1),A
A085  32 8972          LD    (AD5+1),A
A088  32 8A6A          LD    (CP80+1),A
A08B  32 8A8C          LD    (ESCJ80+1),A
A08E  3D               DEC   A
A08F  32 88E5          LD    (AD3+1),A
A092  32 8912          LD    (AD4+1),A
A095  32 8A17          LD    (AD6+1),A
A098  E5               PUSH  HL
A099  21 027F          LD    HL,639      ;Se calculeaza limita pentru zona
A09C  3A 91D6          LD    A,(NCR)     ;grafica si se introduce in
A09F  5F               LD    E,A         ;variabila XMAX
A0A0  16 00            LD    D,0
A0A2  06 03            LD    B,3
A0A4  CB 23     CDS30: SLA   E
A0A6  CB 12            RL    D
A0A8  10 FA            DJNZ  CDS30
A0AA  B7               OR    A
A0AB  ED 52            SBC   HL,DE
A0AD  22 A299          LD    (XMAX),HL
A0B0  E1               POP   HL
A0B1  CB A6            RES   4,(HL)
A0B3  18 A0            JR    CDS7
A0B5  CB 7E     CDS1:  BIT   7,(HL)      ;Preia tipul dreptei indicata de
A0B7  20 0E            JR    NZ,CDS9     ;utilizator si il depune in
                                         ;variabila TDR
A0B9  06 04            LD    B,4         ;Preia primii 4 biti
A0BB  CB 19     CDS20: RR    C
A0BD  17               RLA
A0BE  10 FB            DJNZ  CDS20
A0C0  32 91D7          LD    (TDR),A
A0C3  CB FE            SET   7,(HL)
A0C5  18 90            JR    CDS8
A0C7  3A 91D7   CDS9:  LD    A,(TDR)     ;Preia urmatorii 4 biti
A0CA  06 04            LD    B,4
A0CC  CB 19     CDS23: RR    C
A0CE  17               RLA
A0CF  10 FB            DJNZ  CDS23
A0D1  32 91D7          LD    (TDR),A
A0D4  CB BE            RES   7,(HL)
A0D6  CB 9E            RES   3,(HL)
A0D8  C3 A055          JP    CDS7

                ;SUBRUTINELE GRAFICE
```

```
                        ;TRASAREA UNEI DREPTE OARECARE
A0D8  79                GR:   LD    A,C
A0DC  FE 0D                   CP    0DH           ;Este codul lui CR ?
A0DE  28 04                   JR    Z,G1
A0E0  FE 1F                   CP    1FH           ;Este codul lui US ?
A0E2  20 08                   JR    NZ,G2
A0E4  21 91D5           G1:   LD    HL,ML         ;Iesire din regim grafic
A0E7  CB 96                   RES   2,(HL)
A0E9  C3 8987                 JP    RETCO
A0EC  FE 1D             G2:   CP    1DH           ;Este cod GS ?
A0EE  20 0A                   JR    NZ,G4
A0F0  21 91D8                 LD    HL,INDIC
A0F3  CB 8E                   RES   1,(HL)        ;DA, executa functia PLOT
A0F5  CB 96                   RES   2,(HL)        ;INDIC 1, INDIC 2
A0F7  C3 8987                 JP    RETCO
A0FA  FE 1B             G4:   CP    1BH           ;Este cod ESC ?
A0FC  CA 88C9                 JP    Z,CP1BP2
A0FF  DD 21 91D8              LD    IX,INDIC ;In IX se incarca adresa indicator
A103  E6 60                   AND   60H           ;Sint coordonate X,Y ?
A105  CA A3CF                 JP    Z,ERR         ;NU, salt la eroare
A108  FE 20                   CP    20H           ;Este HI Y sau HI X ?
A10A  20 17                   JR    NZ,G5
A10C  DD CB 00 46             BIT   0,(IX)        ;Este HI Y ?  INDIC0
A110  79                      LD    A,C
A111  20 08                   JR    NZ,G6
A113  E6 1F                   AND   1FH
A115  DD 77 06                LD    (IX+6),A      ;DA, depune-l in variabila HIYN
A118  C3 8987                 JP    RETCO
A11B  E6 1F             G6:   AND   1FH
A11D  DD 77 08                LD    (IX+8),A      ;NU, depune-l in variabila HIXN
A120  C3 8987                 JP    RETCO
A123  FE 60             G5:   CP    60H           ;Este LO Y ?
A125  20 0D                   JR    NZ,G7
A127  DD CB 00 C6             SET   0,(IX)        ;INDIC0
A12B  79                      LD    A,C
A12C  E6 1F                   AND   1FH
A12E  DD 77 05                LD    (IX+5),A      ;DA, depune-l in variabila LOYN
A131  C3 8987                 JP    RETCO
A134  DD CB 00 86       G7:   RES   0,(IX)        ;INDIC0
A138  79                      LD    A,C
A139  E6 1F                   AND   1FH
A13B  DD 77 07                LD    (IX+7),A      ;NU, depune-l in variabila LOXN
A13E  ED 5B 91DD              LD    DE,(LOYN)
A142  CD A269                 CALL  ALIN          ;Aliniaza variabilele HIYN, LOYN
A145  DD CB 00 4E             BIT   1,(IX)        ;INDIC1, se traseaza segmentul ?
A149  20 0A                   JR    NZ,G8
A14B  ED 53 91D9              LD    (LOYV),DE     ;NU, se depune variabila Y in
                                                  ;locatiile LOYV, HIYV
A14F  DD CB 00 CE             SET   1,(IX)        ;INDIC1, la urmatoarea trecere
A153  18 01                   JR    G9            ;se va trasa segmentul?
A155  D5                G8:   PUSH  DE            ;DA, se traseaza
A156  ED 5B 91DF        G9:   LD    DE,(LOXN)
A15A  CD A269                 CALL  ALIN          ;Aliniaza variabilele HIXN, LOXN
A15D  DD CB 00 56             BIT   2,(IX)        ;INDIC2, se traseaza segmentul ?
A161  20 0B                   JR    NZ,G10
A163  ED 53 91DB              LD    (LOXV),DE     ;NU, se depune variabila X in
                                                  ;locatiile LOXV, HIXV
A167  DD CB 00 D6             SET   2,(IX)        ;INDIC2, la urmatoarea trecere
A16B  C3 8987                 JP    RETCO         ;se va trasa segmentul
A16E  C1                G10:  POP   BC            ;DA, se traseaza
A16F  3E 00                   LD    A,0           ;Se initializeaza INDICA0...7
```

```
A171  2A 91DB            LD    HL,(LOXV)    ;Se stabileste semnul lui XABS
A174  B7                 OR    A
A175  ED 52              SBC   HL,DE        ;XV-XN
A177  38 09              JR    C,G11        ;Salt daca XV<XN
A179  20 04              JR    NZ,G12       ;Salt daca XV>XN
A17B  DD CB 00 FE        SET   7,(IX)       ;INDIC7, daca XV=XN
A17F  EB           G12:  EX    DE,HL        ;DE contine XABS, INDICA3 LO
A180  18 0B              JR    G13          ;pentru sgnXABS= -
A182  EB           G11:  EX    DE,HL
A183  ED 5B 91DB         LD    DE,(LOXV)
A187  B7                 OR    A
A188  ED 52              SBC   HL,DE        ;XN-XV
A18A  EB                 EX    DE,HL        ;DE contine XABS
A18B  CB DF              SET   3,A          ;INDICA3 HI pentru sgnXABS= +
A18D  D5           G13:  PUSH  DE           ;Se stabileste semnul lui YABS
A18E  59                 LD    E,C
A18F  50                 LD    D,B
A190  2A 91D9            LD    HL,(LOYV)
A193  B7                 OR    A
A194  ED 52              SBC   HL,DE        ;YV-YN
A196  38 13              JR    C,G14        ;Salt daca YV<YN
A198  20 0E              JR    NZ,G15       ;Salt daca YV>YN
A19A  DD CB 00 F6        SET   6,(IX)       ;INDIC6, daca YV=YN
A19E  DD CB 00 7E        BIT   7,(IX)
A1A2  28 04              JR    Z,G15
A1A4  DD CB 00 EE        SET   5,(IX)       ;INDIC5, daca XV=XN si YV=YN,
                                            ;deci pentru un segment punct
A1A8  EB           G15:  EX    DE,HL        ;DE contine YABS, INDICA4 LO
A1A9  18 0B              JR    G17          ;pentru sgnYABS= -
A1AB  EB           G14:  EX    DE,HL
A1AC  ED 5B 91D9         LD    DE,(LOYV)
A1B0  B7                 OR    A
A1B1  ED 52              SBC   HL,DE        ;YN-YV
A1B3  EB                 EX    DE,HL        ;DE contine YABS
A1B4  CB E7              SET   4,A          ;INDICA4 HI pentru sgnYABS= +
A1B6  D5           G17:  PUSH  DE
A1B7  C1                 POP   BC
A1B8  D1                 POP   DE           ;BC contine XABS, DE contine YABS
A1B9  D9                 EXX
A1BA  ED 4B 91D9         LD    BC,(LOYV)
A1BE  ED 5B 91DB         LD    DE,(LOXV)    ;BC' contine YV, DE' contine XV
A1C2  DD CB 00 6E        BIT   5,(IX)       ;Este un singur punct ?
A1C6  28 06              JR    Z,G40
A1C8  CD A29B            CALL  PLOT         ;DA, traseaza-l
A1CB  C3 A259            JP    G41          ;Sari la sfirsit
A1CE  DD CB 00 76  G40:  BIT   6,(IX)       ;Este un segment orizontal ?
A1D2  28 06              JR    Z,G42
A1D4  CD A2F6            CALL  ORIZ         ;DA, traseaza-l
A1D7  C3 A251            JP    G43
A1DA  DD CB 00 7E  G42:  BIT   7,(IX)       ;Este un segment vertical ?
A1DE  28 05              JR    Z,G44
A1E0  CD A353            CALL  VERTIC       ;DA, traseaza-l
A1E3  18 6C              JR    G43
A1E5  CD A29B      G44:  CALL  PLOT         ;In cazul unui segment oarecare
A1E8  D9                 EXX                ;se traseaza primul punct din
A1E9  D5                 PUSH  DE           ;segment. Se stabileste daca
A1EA  E1                 POP   HL           ;deplasarea e mai mare pe
A1EB  B7                 OR    A            ;orizontala sau pe verticala
A1EC  ED 42              SBC   HL,BC        ;XABS-YABS
A1EE  30 0A              JR    NC,G18       ;Salt daca XABS>=YABS
A1F0  69                 LD    L,C
```

```
A1F1  60          LD    H,B
A1F2  4B          LD    C,E
A1F3  42          LD    B,D
A1F4  5D          LD    E,L
A1F5  54          LD    D,H
A1F6  CB EF       SET   5,A       ;INDICA5 HI pentru YABS>XABS
A1F8  18 02       JR    G19
A1FA  6B    G18:  LD    L,E
A1FB  62          LD    H,D       ;INDICA5 LO pentru YABS<=XABS
A1FC  E5    G19:  PUSH  HL
A1FD  D9          EXX
A1FE  E1          POP   HL        ;HL=DE=HL'=MAX(XABS,YABS),
A1FF  2C          INC   L         ;BC=MIN(XABS,YABS)
A200  2D          DEC   L
A201  28 01       JR    Z,G33
A203  24          INC   H
A204  D9    G33:  EXX
A205  CB 3C       SRL   H
A207  CB 1D       RR    L         ;HL=HL/2
A209  D9          EXX
A20A  D9    G30:  EXX
A20B  09          ADD   HL,BC
A20C  E5          PUSH  HL
A20D  B7          OR    A
A20E  ED 52       SBC   HL,DE
A210  38 06       JR    C,G20     ;Salt daca MAX(XABS,YABS) >
A212  33          INC   SP        ;MAX(XABS,YABS)/2+MIN(XABS,YABS)
A213  33          INC   SP
A214  CB F7       SET   6,A       ;INDICA6 HI pentru deplasare pe
A216  18 03       JR    G21       ;diagonala
A218  E1    G20:  POP   HL
A219  CB B7       RES   6,A       ;INDICA6 LO pentru deplasare pe
                                  ;orizontala sau verticala
A21B  D9    G21:  EXX
A21C  CB 77       BIT   6,A       ;INDICA6, se face deplasarea pe
A21E  28 12       JR    Z,G22     ;diagonala?
A220  CB 5F       BIT   3,A       ;DA, INDICA3, cu deplasarea spre
A222  28 03       JR    Z,G23     ;dreapta?
A224  13          INC   DE        ;DA, diagonala cu deplasarea spre
A225  18 01       JR    G24       ;dreapta
A227  1B    G23:  DEC   DE        ;diagonala cu deplasare stinga
A228  CB 67    G24:  BIT   4,A     ;INDICA4, cu deplasare in sus ?
A22A  28 03       JR    Z,G25
A22C  03          INC   BC        ;DA, diag cu deplasare in sus
A22D  18 19       JR    G28
A22F  0B    G25:  DEC   BC        ;diagonala cu deplasare in jos
A230  18 16       JR    G28
A232  CB 6F    G22:  BIT   5,A     ;INDICA5, se face deplasarea
A234  28 0A       JR    Z,G26     ;vertical?
A236  CB 67       BIT   4,A       ;DA, INDICA4, vertical in sus ?
A238  28 03       JR    Z,G27
A23A  03          INC   BC        ;DA, vertical in sus
A23B  18 0B       JR    G28
A23D  0B    G27:  DEC   BC        ;vertical in jos
A23E  18 08       JR    G28
A240  CB 5F    G26:  BIT   3,A     ;INDICA3, orizontal spre dreapta ?
A242  28 03       JR    Z,G29
A244  13          INC   DE        ;DA, orizontal spre dreapta
A245  18 01       JR    G28
A247  1B    G29:  DEC   DE        ;orizontal spre stinga
A248  CD A29B  G28:  CALL  PLOT     ;Se traseaza punctul obtinut
```

```
A24B  2D              DEC   L           ;Mai exista puncte pe segment ?
A24C  20 BC           JR    NZ,G30      ;DA, salt
A24E  25              DEC   H
A24F  20 B9           JR    NZ,G30      ;DA, salt
A251  ED 43 91D9  G43:LD    (LOYV),BC   ;NU, depune coordonatele
                                        ;ultimului pct
A255  ED 53 91DB      LD    (LOXV),DE   ;Initializ indicatorii si iesi
A259  D9          G41:EXX               ;din subrutina
A25A  DD CB 00 B6 G31:RES   6,(IX)
A25E  DD CB 00 BE     RES   7,(IX)
A262  DD CB 00 AE     RES   5,(IX)
A266  C3 8987         JP    RETCO
A269  06 03       ALIN:LD   B,3         ;Subprogram pentru alinierea
A26B  CB 23       AL1: SLA  E           ;informatiei din locatiile
A26D  10 FC           DJNZ  AL1         ;pereche LOXN,HIXN si respectiv
A26F  06 03           LD    B,3         ;LOYN,HIYN
A271  CB 3A       AL2: SRL  D
A273  CB 1B           RR    E
A275  10 FA           DJNZ  AL2
A277  C9              RET

A278  AF          PLOTCR:XOR  A     ;La intrare in aceasta subrutina in DE
A279  69               LD   L,C     ;avem coordonata X, in BC coordonata Y
A27A  60               LD   H,B     ;La iesire din subrutina vom avea
A27B  CD A2E5          CALL IMP8    ;coordonatel caracterului alfanumeric din
A27E  EB               EX   DE,HL   ;care face parte punctul curent
A27F  67               LD   H,A     ;Astfel:D contine coordonata X a
A280  CD A2E5          CALL IMP8    ;caracterului alfanumeric, H contine
A283  55               LD   D,L     ;coordonata X a punctului in cadrul
A284  1C               INC  E       ;caracterului alfanumeric, iar E contine
A285  D6 07            SUB  7       ;coordonata Y a caracterului alfanumeric,
A287  ED 44            NEG          ;L contine coordonata Y a punctului in
A289  6F               LD   L,A     ;cadrul caracterului alfanumeric
A28A  3A 91D6          LD   A,(NCR)       ;Corectie datorata SCROLL-ului
A28D  82               ADD  A,D           ;hard
A28E  57               LD   D,A
A28F  3A FB90          LD   A,(ROLL)
A292  93               SUB  E             ;Corectie datorata pozitionarii
A293  30 02            JR   NC,P6         ;originii in stinga jos
A295  C6 18            ADD  A,CARR
A297  5F          P6:  LD   E,A
A298  C9               RET

A299  7F 02       XMAX:DB   7FH,2

A29B  C5          PLOT:PUSH BC          ;Subrutina pentru desenarea unui
A29C  D5               PUSH DE          ;punct oarecare
A29D  E5               PUSH HL
A29E  F5               PUSH AF
A29F  AF               XOR  A
A2A0  21 00BF          LD   HL,CARR*8-1 ;Daca o coordonata pe axa Y depa-
A2A3  ED 42            SBC  HL,BC       ;seste valoarea 191, respectiv
A2A5  30 03            JR   NC,P10      ;287 ea este redusa la aceasta
A2A7  01 00BF          LD   BC,CARR*8-1 ;valoare
A2AA  AF          P10: XOR  A
A2AB  2A A299          LD   HL,(XMAX)   ;Daca o coordonata pe axa X
A2AE  ED 52            SBC  HL,DE       ;depaseste valoarea din XMAX, ea
A2B0  30 04            JR   NC,P11      ;este redusa la valoarea din XMAX
A2B2  ED 5B A299       LD   DE,(XMAX)
A2B6  CD A278     P11: CALL PLOTCR
A2B9  E5               PUSH HL
```

```
A2BA  CD BB64          CALL  FHLP1
A2BD  C1               POP   BC       ;In HL avem adresa caracterului pe ecran
A2BE  0C               INC   C        ;Registrul C da deplasarea pe Y
A2BF  0D               DEC   C
A2C0  28 04            JR    Z,P7
A2C2  24          P1:  INC   H        ;In HL se inscrie adresa octetului ce
A2C3  0D               DEC   C        ;urmeaza a fi modificat pe ecran
A2C4  20 FC            JR    NZ,P1
A2C6  4E          P7:  LD    C,(HL)   ;In C octetul ce va fi modificat
A2C7  3A 91B7          LD    A,(TDR)  ;In A tipul dreptei indicat de utilizator
A2CA  07               RLCA           ;Punctul va fi stins?
A2CB  32 91B7          LD    (TDR),A
A2CE  04               INC   B        ;In B numarul bitului din octet
A2CF  38 08            JR    C,P2
A2D1  AF               XOR   A        ;DA, punctul va fi stins
A2D2  2F               CPL
A2D3  1F          P3:  RRA
A2D4  10 FD            DJNZ  P3
A2D6  A1               AND   C
A2D7  18 06            JR    P4
A2D9  AF          P2:  XOR   A        ;NU, punctul va fi aprins
A2DA  37               SCF
A2DB  1F          P5:  RRA
A2DC  10 FD            DJNZ  P5
A2DE  B1               OR    C
A2DF  77          P4:  LD    (HL),A   ;Se inscrie octetul modificat
A2E0  F1               POP   AF       ;in memoria VIDEO
A2E1  E1               POP   HL
A2E2  D1               POP   DE
A2E3  C1               POP   BC
A2E4  C9               RET
A2E5  06 03       IMP8: LD    B,3      ;Subrutina ce realizeaza impartirea
A2E7  CB 3A       IM1:  SRL   D        ;continutului lui DE la 8,
A2E9  CB 1B           RR    E        ;rezultatul in E, iar restul in A
A2EB  CB 1F           RR    A
A2ED  10 F8           DJNZ  IM1
A2EF  06 03           LD    B,3
A2F1  CB 07       IM2:  RLC   A
A2F3  10 FC           DJNZ  IM2
A2F5  C9              RET

A2F6  D9          OR12: EXX            ;Subrutina pt trasarea
A2F7  D5                PUSH  DE       ;segmentelor de dreapta orizontale
A2F8  D9                EXX            ;La intrare BC' contine YV, DE' contine
A2F9  E1                POP   HL       ;XV, iar DE contine lungimea segmentului
A2FA  CB 5F             BIT   3,A      ;INDICA3, se face deplasare spre
A2FC  20 06             JR    NZ,OR1   ;stinga?
A2FE  EB                EX    DE,HL    ;DA, spre stinga
A2FF  B7                OR    A        ;Se muta originea segmentului in
A300  ED 52             SBC   HL,DE    ;stinga pentru ca totusi desenarea
A302  EB                EX    DE,HL    ;sa se faca de la stinga spre dreapta
A303  D5                PUSH  DE
A304  F5          OR1:  PUSH  AF
A305  7B          OR3:  LD    A,E
A306  E6 07             AND   7        ;Originea segmentului e la
                                       ;inceputul unui octet ?
A308  20 09             JR    NZ,OR7
A30A  7D                LD    A,L      ;DA, exista mai mult de 7
A30B  FE 08             CP    8        ;puncte de desenat?
A30D  30 0B             JR    NC,OR2
A30F  24                INC   H
```

```
A310  25          DEC   H
A311  20 07       JR    NZ,OR2
A313  CD A29B     OR7:  CALL  PLOT    ;NU,se apeleaza subrutina care
A316  13                DEC   DE      ;inscrie cite un punct
A317  2B                DEC   HL      ;Se decrementeaza lungimea care
A318  18 0A              JR    OR5     ;mai e de trasat
A31A  CD A32F     OR2:  CALL  PLOTO   ;Se apeleaza subrutina care
A31D  3E 08             LD    A,8     ;inscrie cite un octet
A31F  13          OR4:  INC   DE
A320  2B                DEC   HL      ;Se scade lungimea cu 8
A321  3D                DEC   A
A322  20 FB             JR    NZ,OR4
A324  7D          OR5:  LD    A,L     ;Mai exista puncte de trasat ?
A325  B4                OR    H
A326  20 B0             JR    NZ,OR3
A328  F1                POP   AF      ;NU
A329  CB 5F             BIT   3,A     ;INDICA3, iesire din subrutina
A32B  20 01             JR    NZ,OR6
A32D  D1                POP   DE
A32E  C9          OR6:  RET

A32F  C5          PLOTO: PUSH  BC          ;Subrutina ce relizeaza
A330  D5                 PUSH  DE          ;inscrierea pe ecran a cite unui
A331  E5                 PUSH  HL          ;octet pe orizontala
A332  AF                 XOR   A
A333  2A A299            LD    HL,(XMAX)   ;Octetul ce trebuie inscris se
A336  ED 52              SBC   HL,DE       ;afla in zona grafica ?
A338  38 15              JR    C,PO3
A33A  CD A278            CALL  PLOTCR      ;DA, vezi subrutina PLOT
A33D  E5                 PUSH  HL
A33E  CD 8B64            CALL  FHLP1
A341  C1                 POP   BC
A342  41                 LD    B,C
A343  04                 INC   B           ;B relizeaza pozitionarea pe verticala
A344  05                 DEC   B
A345  28 03              JR    Z,PO1
A347  24          PO2:   INC   H
A348  10 FD              DJNZ  PO2
A34A  3A 9107     PO1:   LD    A,(TBR)     ;Se incarca tipul dreptei
A34D  00          PLOTOA: NOP              ;Cind subrutina este apelata sub
                                           ;modul de lucru Introducere
                                           ;Grafica in locul instructiei
                                           ;NOP se inscrie instructia XOR
A34E  77                 LD    (HL),A      ;Se inscrie in memoria VIDEO
A34F  E1          PO3:   POP   HL
A350  D1                 POP   DE
A351  C1                 POP   BC
A352  C9                 RET

A353  D9          VERTIC: EXX              ;Subrutina pentru trasarea
A354  C5                  PUSH  BC        ;segmentelor de dreapta verticale
A355  D9                  EXX
A356  E1                  POP   HL
A357  CB 67               BIT   4,A       ;Comentariu, vezi subrutina ORIZ
A359  20 0C               JR    NZ,VE1
A35B  C5                  PUSH  BC
A35C  E5                  PUSH  HL
A35D  C1                  POP   BC
A35E  E1                  POP   HL
A35F  B7                  OR    A
A360  ED 42               SBC   HL,BC
```

```
A362  C5           PUSH  BC
A363  E5           PUSH  HL
A364  C1           POP   BC
A365  E1           POP   HL
A366  C5           PUSH  BC
A367  F5     VE1:  PUSH  AF
A368  79     VE3:  LD    A,C
A369  E6 07        AND   7
A36B  20 09        JR    NZ,VE7
A36D  7D           LD    A,L
A36E  FE 08        CP    8
A370  30 0B        JR    NC,VE2
A372  24           INC   H
A373  25           DEC   H
A374  20 07        JR    NZ,VE2
A376  CD A29B VE7: CALL  PLOT
A379  03           INC   BC
A37A  2B           DEC   HL
A37B  18 0A        JR    VE5
A37D  CD A392 VE2: CALL  PLOTV
A380  3E 08        LD    A,8
A382  03      VE4: INC   BC
A383  2B           DEC   HL
A384  3D           DEC   A
A385  20 FB        JR    NZ,VE4
A387  7D      VE5: LD    A,L
A388  B4           OR    H
A389  20 DD        JR    NZ,VE3
A38B  F1           POP   AF
A38C  CB 67        BIT   4,A
A38E  20 01        JR    NZ,VE6
A390  C1           POP   BC
A391  C9      VE6: RET

A392  C5     PLOTV: PUSH  BC          ;Subrutina ce realizeaza
A393  D5            PUSH  DE          ;inscrierea a cite 8 puncte
A394  E5            PUSH  HL          ;pe verticala
A395  AF            XOR   A           ;Comentarii, vezi subrutinele
A396  21 00BF       LD    HL,CARR*8-1       ;PLOT si PLOTO
A399  ED 42         SBC   HL,BC
A39B  38 2E         JR    C,PV1
A39D  CD A278       CALL  PLOTCR
A3A0  E5            PUSH  HL
A3A1  CD 8B64       CALL  FHLP1
A3A4  C1            POP   BC
A3A5  0C            INC   C
A3A6  C5     PV6:   PUSH  BC
A3A7  4E            LD    C,(HL)
A3A8  3A 91D7       LD    A,(TDR)
A3AB  0F     PLOTVC: RRCA            ;In modul de lucru IG in locul
A3AC  00            NOP             ;instructiilor RRCA si NOP se
                                    ;inscrie RLCA si RLCA
A3AD  32 91D7       LD    (TDR),A
A3B0  04            INC   B
A3B1  38 09         JR    C,PV2
A3B3  AF            XOR   A
A3B4  2F            CPL
A3B5  CB 1F  PV3:   RR    A
A3B7  10 FC         DJNZ  PV3
A3B9  A1            AND   C
A3BA  18 07         JR    PV4
```

```
A3BC  AF           PV2:   XOR   A
A3BD  37                  SCF
A3BE  CB 1F        PV5:   RR    A
A3C0  10 FC               DJNZ  PV5
A3C2  B1           PLOTVA: OR   C              ;In modul de lucru IG in locul
                                               ;instructiei OR C se inscrie
                                               ;instructia XOR C
A3C3  77           PV4:   LD    (HL),A
A3C4  C1                  POP   BC
A3C5  24                  INC   H
A3C6  00           PLOTVB: NOP                 ;In modul de lucru IG in locul
A3C7  00                  NOP                  ;instructiilor NOP si NOP se
                                               ;inscrie INC H si DEC C
A3C8  0D                  DEC   C
A3C9  20 DB               JR    NZ,PV6
A3CB  E1           PV1:   POP   HL
A3CC  D1                  POP   DE
A3CD  C1                  POP   BC
A3CE  C9                  RET
A3CF  CD 8001      ERR:   CALL  TYPE
A3D2  20 43 6F 64         DB    ' Cod coordonata incorect',0DH,0AH,24H
A3D6  20 63 6F 6F
A3DA  72 64 6F 6E
A3DE  61 74 61 20
A3E2  69 6E 63 6F
A3E6  72 65 63 74
A3EA  0D 0A 24
A3ED  C3 8987              JP    RETCO
                  ;
A3F0  F3           ZOBIOS: DI
A3F1  31 FFFC              LD    SP,0FFFCH
A3F4  ED 5E                IM    2
A3F6  CD 9459              CALL  ERASE1
A3F9  CD 8001              CALL  TYPE
A3FC  0D 0A                DB    0DH,0AH
A3FE  54 69 6D 2D          DB    "Tim-S Plus CP/M 64K v2.2 13.12.89 Timisoara"
A402  53 20 50 6C
A406  75 73 20 43
A40A  50 2F 4D 20
A40E  36 34 4B 20
A412  76 32 2E 32
A416  20 31 33 2E
A41A  31 32 2E 38
A41E  39 20 54 69
A422  6D 69 73 6F
A426  61 72 61
A429  0D 0A                DB    0DH,0AH
A42B  48 65 6C 6C          DB    "Hello, man! How do you do?"
A42F  6F 2C 20 6D
A433  61 6E 21 20
A437  48 6F 77 20
A43B  64 6F 20 79
A43F  6F 75 20 64
A443  6F 3F
A445  0D 0A                DB    0DH,0AH
A447  24                   DB    24H
A448  11 FB2C              LD    DE,RTCNT
A44B  0E 3E                LD    C,TRK-RTCNT
A44D  AF                   XOR   A
A44E  CD 8495              CALL  CYBR
A451  21 00A9              LD    HL,VIOBYT
```

```
A454  22 0003                    LD      (10BYTE),HL
A457  F3                 WBOOT1: DI
A458  3E 0F                      LD      A,0FH
A45A  32 FB8B                    LD      (COCFD),A
A45D  3E 31                      LD      A,31H
A45F  32 FB8A                    LD      (C7FFD),A
A462  3E 01                      LD      A,001H
A464  32 FB59                    LD      (ATDLOC),A
A467  31 FB6A                    LD      SP,ATDLOC+17
A46A  21 A4E9                    LD      HL,RETEXI
A46D  E5                         PUSH    HL
A46E  E5                         PUSH    HL
A46F  E5                         PUSH    HL
A470  E5                         PUSH    HL
A471  E5                         PUSH    HL
A472  E5                         PUSH    HL
A473  E5                         PUSH    HL
A474  E5                         PUSH    HL
A475  31 FFFC                    LD      SP,0FFFCH
A478  3E FF                      LD      A,0FFH
A47A  32 F6E8                    LD      (DPBASE+16),A
A47D  32 F6F9                    LD      (DPBASE+33),A
A480  11 E5B8                    LD      DE,CCP+07B8H
A483  0E 48                      LD      C,048H
A485  AF                         XOR     A
A486  CD 8495                    CALL    CYBR
A489  11 E90A                    LD      DE,BDOS+0304H
A48C  0E 3D                      LD      C,03DH
A48E  CD 8495                    CALL    CYBR
A491  2F                         CPL
A492  32 FB38                    LD      (CURDEN),A
A495  01 0CFD                    LD      BC,0CFDH
A498  ED 78                      IN      A,(C)
A49A  F5                         PUSH    AF
A49B  E6 FE                      AND     0FEH
A49D  ED 79                      OUT     (C),A
A49F  CD F45B            NUI:    CALL    MOON
A4A2  21 DE00                    LD      HL,0DE00H
A4A5  01 0004                    LD      BC,4
A4A8  3E 06                      LD      A,6
A4AA  CD 05CB                    CALL    R05CB
A4AD  30 F0                      JR      NC,NUI
A4AF  01 0CFD                    LD      BC,0CFDH
A4B2  F1                         POP     AF
A4B3  F3                         DI
A4B4  ED 79                      OUT     (C),A
A4B6  3E C3                      LD      A,0C3H
A4B8  32 0000                    LD      (0000H),A
A4BB  21 F403                    LD      HL,WBOOTE
A4BE  22 0001                    LD      (0001H),HL
A4C1  32 0005                    LD      (0005H),A
A4C4  21 E606                    LD      HL,BDOS
A4C7  22 0006                    LD      (6),HL
A4CA  21 FB59                    LD      HL,ATDLOC
A4CD  7E                         LD      A,(HL)
A4CE  23                         INC     HL
A4CF  37                 IDRV:   SCF
A4D0  3F                         CCF
A4D1  1F                         RRA
A4D2  F5                         PUSH    AF
A4D3  E5                         PUSH    HL
```

```
A4D4  DC A4E5              CALL  C,IDRVX
A4D7  E1                   POP   HL
A4D8  F1                   POP   AF
A4D9  23                   INC   HL
A4DA  23                   INC   HL
A4DB  B7                   OR    A
A4DC  20 F1                JR    NZ,IDRV
A4DE  3A 0004              LD    A,(USRDSK)
A4E1  4F                   LD    C,A
A4E2  C3 F6A7              JP    BALOO
                           ;
A4E5  5E                   IDRVX: LD    E,(HL)
A4E6  23                   INC   HL
A4E7  56                   LD    D,(HL)
A4E8  D5                   PUSH  DE
A4E9  C9                   RETEXI: RET
                           END   START
```

Macros:

Symbols:

8964	A2	8803	A20	8377	ABCD
88B6	ABCY	8867	AD1	F498	AD13
88E0	AD2	88E4	AD3	8911	AD4
8971	AD5	8A16	AD6	FC1C	ADRCOD
8A0A	AINY	A26B	AL1	A271	AL2
8336	ALFA0	A269	ALIN	F9D8	ALV0
FA26	ALV1	FA74	ALV2	FA96	ALV3
FB6D	AMAN	970B	AMARG	85D2	ARD0
9707	AROB	FB59	ATDLOC	8555	AXH
8ABD	AY	8AAD	BA	F589	BAGHEERA
F6A7	BALOO	DC00	BASE	8C35	BASE1
F4C5	BD13	E606	BDOS	8F35	BEEP
8A62	BETA	F400	BIOS	970E	BIT
F590	BOOT	FB86	BORDER	8A36	BSNORM
0080	BUFF	FB2E	BUFWRT	FB8B	COCFD
9204	C36	970D	C37	970C	C48
FB8A	C7FFD	862A	CAP	9112	CARINC
0018	CARR	9066	CARVAL	A00A	CAZ
A011	CAZ1	DE00	CCP	847C	CDMA
A0B5	CDS1	A07C	CDS11	A07F	CDS12
A074	CDS13	A06A	CDS2	A0BB	CDS20
A0CC	CDS23	A066	CDS3	A0A4	CDS30
A062	CDS4	A05E	CDS5	A05A	CDS6
A055	CDS7	A057	CDS8	A0C7	CDS9
A039	CDSPEC	FB81	CDT	80D9	CHKUNA
86D5	CI1	86DD	CI2	86E5	CI3
86ED	CI4	86F1	CIN1	86B3	CITAB
89F6	CJOS	83C3	CKDEN	0040	CKSIZE
8477	CLOSDMA	FB83	CMAN	FB82	CMON
888E	COM	889F	COM1	90CE	COMINK
819B	COMM1	81E1	COMM2	81A3	COMMX
90B6	COMPAP	F5CA	COMUT	F68A	COMUT0
8000	COMUT1	F643	COMUT2	F655	COMUT2A
F64D	COMUT2C	F5E1	COMUT3	94FE	CON4
F59C	CONIN	851E	CONIN1	F5A2	CONOUT
8523	CONOUT1	F5A8	CONST	8531	CONST1
8546	CONSTX	862B	CONSUL	86BB	COTAB
9088	CP0	88B5	CP07	88EB	CP08
88D1	CP09	88AC	CP0A	88C5	CP1B
88C9	CP1BP2	87E5	CP3A	8A69	CP80
9058	CP81	FB8D	CPAPIN	89E9	CPB
8A0F	CPC	8A2B	CPD	8A42	CPE

9073	CPFO	8A4B	CPH	8A80	CPJ
8A5E	CPK	8ACD	CPN	8AD9	CPO
8AE0	CPY	88A3	CR	818F	CRSEC
9757	CSCA	905C	CSL	8806	CSLOCK
89E4	CSUS	FA05	CSV0	FA53	CSV1
FA85	CSV2	FAD7	CSV3	881C	CTRDGT
FB38	CURDEN	FB30	CURDPH	9646	CURO
9685	CUROJ	9698	CUROJ1	968A	CUROJ2
9670	CUROS	9675	CUROS2	965C	CURV
96A4	CURVD	96A9	CURVD2	96BA	CURVS
96CD	CURVS1	96BF	CURVS2	9726	CY11
9749	CY12	9790	CY120	973F	CY13
975C	CY14	9730	CY15	9786	CY16
97BF	CY17	9752	CY18	9789	CY19
8495	CYBR	809B	D0	F4D5	D10
F4B2	D13	8468	D14	883A	D15
8844	D15P4	89A3	D17	89C0	D171
898A	D18	91AC	D19	8351	D256
F6AE	D2VAL	800D	D4	F6B4	D5VAL
F756	DDSKNO	8366	DDX2	8FC2	DECBC
8A25	DEPLC	8409	DFOUND	89C5	DIOBYT
F958	DIRBUF	8836	DISPO	FB3A	DMAADD
F6D8	DPBASE	F6BA	DPBD2	F6C9	DPBD5
FB33	DPBPNT	F71C	DPBRDK	0010	DRAMD
8B54	DRAW	8240	DRDY	8014	DRNRDY
FB51	DSKNO	FB3C	DSKOP	83A0	DSKSEL
8052	DSKST	FB45	DSTS	8602	DT1
8600	DT105S	8611	DT2	861D	DT3
FB58	DTL	802D	DVNAME	94E3	EA1
94D9	EA2	94E6	EA3	94DA	EA4
8640	ECO1	863F	ECO2	8679	ECO3
8660	ECO4	8658	ECO5	8682	ECO6
8661	ECO7	9472	ECOM16	94A1	ECOM16E
~~94A6~~	~~ECOM16~~	~~F728~~	~~EDFBRDK~~	~~94F2~~	~~EGR1~~
94ED	EGR2	94F5	EGR3	94F6	EGR4
9190	ELCUR	FB56	EOT	9948	EQ1
99E0	EQ10	9A10	EQ11	9A7C	EQ12
9AD5	EQ13	9A6F	EQ14	9AC7	EQ15
9ACE	EQ16	9ABA	EQ17	9AD8	EQ18
9B46	EQ19	9C0D	EQ2	9B4D	EQ20
9B54	EQ21	9B5B	EQ22	9B31	EQ23
9B60	EQ24	9BED	EQ25	9BF4	EQ26
9BFB	EQ27	9C02	EQ28	9BD8	EQ29
993A	EQ3	9C07	EQ30	9CBC	EQ31
9C53	EQ32	9C5A	EQ33	9C46	EQ34
9C5F	EQ35	9CA0	EQ36	9CAF	EQ37
9C93	EQ38	99FF	EQ4	9D23	EQ40
9D20	EQ41	9D13	EQ42	9DA7	EQ43
9DAB	EQ44	9DB0	EQ45	9D93	EQ46
9DBA	EQ47	9E48	EQ48	9A01	EQ5
9F23	EQ50	9E30	EQ51	9E38	EQ52
9E40	EQ53	9E1F	EQ54	9F09	EQ55
9F10	EQ56	9F17	EQ57	9F1E	EQ58
9EF4	EQ59	9A03	EQ6	9FBF	EQ60
9FE0	EQ61	9A05	EQ7	9FEB	EQ71
9A07	EQ8	9A09	EQ9	F5C6	ERASE
9459	ERASE1	8322	ERCONT	855C	ERDSKRW
82E6	EREXIT	A3CF	ERR	8325	ERRIGN
FB91	ESC	98B0	ESCACK	9711	ESCETB
A031	ESCETX	93D4	ESCFF	8A8B	ESCJ80
9508	ESCSUB	DD00	ETPA	849B	EX

94D6	EXEA	94CD	EXEAGR	845A	EXEC
94EA	EXEGR	8919	EXTEND	917A	FADCOD
9413	FF10	9421	FF11	9437	FF12
9451	FF13	93E1	FF20	93EA	FF6
93F7	FF8	93F9	FF81	9407	FF9
8B60	FHL	8B64	FHLP1	83E8	FINDEN
8359	FLACEX	FB7B	FLAGS	FB7C	FLAGS2
9198	FLASH	827A	FLE1	829C	FLE2
82B5	FLE3	8257	FLERR	81F5	FLRDWR
81F2	FLREAD	81FE	FLRETR	844D	FLSEEK
81EC	FLWRT	FB7F	FRAME	A0E4	G1
A16E	G10	A182	G11	A17F	G12
A180	G13	A1AB	G14	A1A8	G15
A1B6	G17	A1FA	G18	A1FC	G19
A0EC	G2	A218	G20	A21B	G21
A232	G22	A227	G23	A228	G24
A22F	G25	A240	G26	A23D	G27
A248	G28	A247	G29	A20A	G30
A25A	G31	A204	G33	A0FA	G4
A1CE	G40	A259	G41	A1DA	G42
A251	G43	A1E5	G44	A123	G5
A11B	G6	A134	G7	A155	G8
A156	G9	811D	GETDPB	8B79	GH1
8B78	GH2	8B88	GH3	8B9E	GH5
8B6C	GHL	8BA3	GHLP1	909A	GOMODO
FB57	GPL	A0DB	GR	9119	GT20
913F	GT7F	FB53	HEAD	91E0	HIXH
91DC	HIXV	91DE	HIYN	91DA	HIYV
907D	HLADRC	F528	HOME	FB89	IOCFD
9159	ICPI	82D0	IDAM	8292	IDCRC
82D6	IDERR	A4CF	IDRV	A4E5	IDRVX
90C8	ILDA0	90C4	ILDBA	A2E7	IM1
A2F1	IM2	A2E5	IMP8	8BC6	IMPAR
886F	INCRY	9061	INCSL	852A	INDEX
91D8	INDIC	8825	INVAL	0003	IOBYTE
FB88	IP7FFD	90DB	IPOPAF	8F75	IRET
8FE0	IYKST0	81F8	JP001	F442	JR001
F44D	JR002	F456	JR003	F4E6	JR004
813C	JR007	8201	JR008	8218	JR009
8232	JR010	8317	JR011	84A4	JR013
8529	JR014	8526	JR015	8C22	JR016
85EF	KAP	F548	KAPOOR	86C3	KBINP
8705	KBSTS	870E	KBSTS2	8750	KEYSWR
8793	KNEW	FB70	KST0	FB74	KST4
871A	L028E	8722	L0296	8730	L029F
8732	L02A1	873C	L02AB	875F	L02BF
8766	L02C6	8771	L02D1	87A8	L0308
87B0	L0310	87BE	L031E	87D0	L032C
87D7	L0333	880A	L034A	87E9	L034F
880F	L0367	8833	L03B2	8F4E	L03D1
8F53	L03D6	8F71	L03F2	F439	L400
F487	L401	F47E	L402	F4BA	L41C
F4CE	L427	F4FC	L48B	F512	L49C
808C	L7B6	8BAB	LA	FB35	LACSEC
FB78	LASTK	91E1	LEFD	8C08	LI1
8C12	LI2	8C16	LI3	F5C0	LIST
8514	LIST1	8517	LIST1X	8693	LISTAR
98A3	LISTS1	F545	LISTST	8C19	LIZOP2
8BFD	LIZOST	F492	LL400	91DF	LOXN
91DB	LOXV	91DD	LOYN	91D9	LOYV
84A1	LPT	84A9	LPTB	84BD	LPTS1

94C1	LPTS2	84C5	LPTS3	8AED	MEMCAR
9AF5	MEMZC	9FFE	MESC	F50D	MINIRD
F518	MINIRD1	FAF7	MINIWR	F502	MINIWR1
91D5	ML	9003	MN0	9025	MN1
9014	MN2	FB7D	MODE	9049	MODE0
9096	MODE1	FB80	MOFL	F45B	MOON
F57F	MOWGLI	FB55	N	91D6	NCR
9206	NCRR1	9251	NCRR2	87FA	NCTRL
9AFC	NCURS	815B	NEWSEC	8117	NOOVF
908F	NOTAST	FC1E	NRCOD	0004	NRDSKS
803B	NRDYTX	970F	NROCT	8752	NUCTRL
A49F	NUI	887E	NUY0	981B	OC1
9902	OC10	9852	OC11	9859	OC12
9897	OC13	983A	OC15	9834	OC16
9887	OC17	9878	OC18	988E	OC19
981E	OC2	9800	OC20	980F	OC21
9863	OC22	9865	OC23	9845	OC3
9851	OC4	984E	OC5	9884	OC6
9812	OC7	9831	OC8	982B	OC9
9800	OCTET	895C	OK	8A7A	OLDX
8A9C	OLDXJ	8488	OPENDMA	A304	OR1
A31A	OR2	A305	OR3	A31F	OR4
A324	OR5	A32E	OR6	A313	OR7
A2F6	ORIZ	A2C2	P1	A2AA	P10
A2B6	P11	FB84	P1FFD	A2D9	P2
A2D3	P3	A2DF	P4	A2DB	P5
A297	P6	A2C6	P7	FB87	P7FFD
FB8C	PAPINK	8BD0	PAR	8BD8	PAR00
97DB	PARTE	896E	PC2	8B42	PICT
FB7E	PIP	F671	PIR	A29B	PLOT
A278	PLOTCR	A32F	PLOTO	A34D	PLOTOA
A392	PLOTV	A3C2	PLOTVA	A3C6	PLOTVB
A3AB	PLOTVC	8080	PNPROT	A34A	P01
A347	P02	A34F	P03	FB85	PORTFE
FB92	POZCUR	A022	PR8251	A016	PR8253
8492	PSPEC	8061	PTRDMR	8504	PUN1
F5AE	PUNCH	84FC	PUNCH1	86AB	PUNTAB
A3CB	PV1	A3BC	PV2	A3B5	PV3
A3C3	PV4	A3BE	PV5	A3A6	PV6
05CB	R05CB	854E	RAD	8440	RCAL
FB6C	RDCFD	8074	RDERR	842C	RDID
F5BA	RDRST	850F	RDRST1	86A3	RDRTAB
85C0	RDS0	85B9	RDS1	F569	RDS2
85D1	RDS3	8588	RDS4	F568	RDS5
85CC	RDS6	857A	RDSKRW	85D9	RDSKW
868B	RDSTAB	F5D4	READ	8099	READ1
F5B4	READER	8507	READER1	8429	READID
84F0	REC1	843C	RECAL	FB44	RECFL
84EC	RECVA	84FB	RECVB	91C2	REF
8AC5	REFBC	8AF0	REFCAR	FB79	REPDEL
FB7A	REPPER	F587	RERDSK	8987	RETC0
A4E9	RETEXI	0010	RETRY	FB90	ROLL
97F3	ROSC11	97F0	RS1	FB43	RSFLG
FB2C	RTCNT	F758	RWBUF	8131	RWBUF0
8140	RWBUF1	81BF	RWDONE	FB2D	RWFLG
81B6	RWMOVE	F535	RWP	F53F	RWRET
8222	RWRET1	FB46	RWSTBL	FB51	RWTBL
9359	S10	9371	S11	93BB	S12
9385	S13	9367	S14	93AA	S15
8B30	S2	9343	S20	934E	S21
9355	S22	8B3F	S3	92E0	S4

```
92B2  S40     92B4  S41     92CB  S42
92FA  S5      9337  S50     933E  S51
9332  S52     930E  S6      9311  S7
9314  S8      9317  S9      8BE0  SALHL
91AF  SALV    8B24  SARI    8B1D  SARI1
92D3  SCEXE   F676  SCOMUT0 F65A  SCOMUT2
9205  SCR0    929C  SCROL1  8B0C  SCROLL
8339  SCTTRN  FB54  SECT    F5DE  SECTR
8327  SECTR1  FB32  SEKDEN  FB2F  SEKDSK
FB39  SEKSEC  FB36  SEKTRK  F5CF  SELDSK
836C  SELDSK1 83DE  SETD1   83CA  SETDEN
F523  SETDMA  9710  SETSC   F530  SETSEC
F52B  SETTRK  F6AB  SFR1    9449  SFRA
944C  SFRGR   F5F2  SHERKAN 84CB  SIOAST
84FB  SIOBST  FC1F  SIRTAS  8129  SPREAD
8827  SSDGT   0100  STACK   F400  START
8BDE  STER    946B  STERA   9462  STEREC
8BF4  STERLI  8980  STESC   869B  STSTAB
958A  SU1     95C8  SU10    954C  SU2
9574  SU20    958E  SU21    95A8  SU22
95C2  SU23    954D  SU3     9546  SU30
9563  SU31    954E  SU32    956A  SU33
957A  SU4     95A4  SU5     9594  SU6
95BE  SU7     95AE  SU8     95D9  SU9
9146  T42     9152  T43     915F  T58
FC21  TA      96FB  TA1     9706  TA2
9207  TAB1    9252  TAB2    96E4  TAS1
96EF  TAS2    96F0  TAS58   96D9  TAS67
90A0  TBORD   91D7  TDR     9503  TDRM
FC59  TE      9134  TERM    89C6  TESTC
8748  TESTE   86F6  TFLAGS  918B  TFLASH
911D  TIPBL   8888  TIPCUR  FC20  TP
F754  TPALOC  9167  TPF     8F80  TRAT
FB6A  TRK     FB52  TRKN0   FB4D  TRKTBL
93A2  TRZ13   938E  TRZ21   9398  TRZ32
FB9C  TTF0    FBAC  TTF1    FBBC  TTF2
FBCC  TTF3    FBDC  TTF4    FBEC  TTF5
FBFC  TTF6    FC0C  TTF7    90F4  TTP
8001  TYPE    FB3E  UNACNT  FB3F  UNADSK
FB42  UNASEC  FB40  UNATRK  0004  USRDSK
A367  VE1     A37D  VE2     A368  VE3
A382  VE4     A387  VE5     A391  VE6
A376  VE7     A353  VERTIC  FB93  VIDEO
8AD3  VIDEOA  00A9  VIOBYT  F596  WBOOT
A457  WBOOT1  F403  WBOOTE  F58C  WERDSK
F5D9  WRITE   8095  WRITE1  81C7  WRSTAT
FB3D  WRTYPE  FB8F  X       94FF  XC
FB6E  XCOCFD  FB6F  XC7FFD  9504  XCM
9635  XCX1    963D  XCX2    9633  XCXCM
F73A  XLT1    A299  XMAX    84D0  XMI1
84E1  XMI2    84D6  XMITA   84FB  XMITB
90EE  XORA    FB8E  Y       8AAF  YA
890E  YAY     9501  YC      9506  YCM
8AED  YCURS   8BAD  ZB      FB94  ZCAR
8B05  ZCMEM   F5F8  ZINTR   9F76  ZINTR1
FB70  ZKST0   93BC  ZN1     93C1  ZN2
A3F0  ZOBIOS  91E2  ZONA1   91EC  ZONA2
91F8  ZONA3   F437  ZZINTR  F433  ZZKSY0
F435  ZZOBIOS 8FBF  ZZZZ
No Fatal error(s)
```