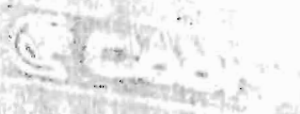




WARE SOFTWARE SOFTWARE



SISTEM DE OPERARE  
DE DISC FLEXIBIL  
PREZENTARE



COORDONATORII SERIEI

dr. ing. Dan Roman

dr. Emil Munteanu

TEHNOREDACTORI

COPERTA SERIEI

designer Liviu Derveșteanu

Implementarea sistemului de operare CP/M s-a realizat de un colectiv de la I.T.C.I. filiala Braşov format din: mat. Arefa Marcel, mat. Pop Mircea condus de: dr. ing. Toase Gheorghe

## C U P R I N S

1.1 Organizarea CP/M .....	6
1.1.1 Particularitățile ale sistemului CP/M CÖBIRA .....	7
1.1.2 Utilizarea memoriei interne .....	8
1.1.3 Organizarea discului .....	8
1.1.4 Dispozitivul consola .....	9
1.1.4.1 Tastatura .....	9
1.1.4.2 Afișajul .....	9
1.1.4.3 Funcțiile de terminal .....	11
1.1.5 Interfața serială RS232 .....	12
1.2 Execuția programelor tranzitorii .....	13
1.3 Convenții pentru apelul funcțiilor de sistem CP/M .....	14
1.4 Particularitățile în utilizarea rutinelor CP/M de lucru cu fișiere pe disc .....	16
1.5 Prezentarea rutinelor CP/M .....	20
RUTINA 0 - Reinițializare sistem CP/M .....	20
RUTINA 1 - Citire caracter de la consolă .....	20
RUTINA 2 - Scriere caracter la consolă .....	20
RUTINA 3 - Citire caracter de la dispozitivul „Reader” curent .....	21
RUTINA 4 - Scriere caracter la dispozitivul „Punch” curent .....	21
RUTINA 5 - Scriere caracter la dispozitivul „List” curent .....	21
RUTINA 6 - Citire/Scriere directă la consolă .....	21
RUTINA 7 - Citire octet IOBYTE .....	22
RUTINA 8 - Modificare octet IOBYTE .....	22
RUTINA 9 - Tipărire la consolă a unui șir de caractere .....	22
RUTINA 10 - Citire buffer consolă .....	23
RUTINA 11 - Citire stare consolă .....	24
RUTINA 12 - Citire versiune sistem .....	24
RUTINA 13 - Inițializare stare sistem discuri .....	24
RUTINA 14 - Selectare disc .....	25
RUTINA 15 - Deschidere fișier .....	25
RUTINA 16 - Închidere fișier .....	26
RUTINA 17 - Caută în „director” prima intrare .....	26
RUTINA 18 - Caută în „director” următoarea intrare .....	27
RUTINA 19 - Ștergere fișier .....	27
RUTINA 20 - Citire secvențială .....	28
RUTINA 21 - Scriere secvențială .....	28
RUTINA 22 - Creare fișier .....	29
RUTINA 23 - Schimbare nume fișier .....	29
RUTINA 24 - Citire vector de unități-disc active .....	30

RUTINA 25	— Citire număr disc selectat .....	30
RUTINA 26	— Modificare „adresa DMA” .....	30
RUTINA 27	— Citire adresă vector de alocare .....	31
RUTINA 28	— Setare atribut R/O pentru o unitate de disc .....	31
RUTINA 29	— Citire vector de unități R/O .....	31
RUTINA 30	— Modificare atribute fișier .....	32
RUTINA 31	— Citire adresă „bloc de parametri disc” .....	32
RUTINA 32	— Citire/Modificare număr utilizator .....	33
RUTINA 33	— Citire directă .....	33
RUTINA 34	— Scriere directă .....	34
RUTINA 35	— Determinare lungime fișier .....	35
RUTINA 36	— Determinare număr înregistrare .....	35
RUTINA 37	— Dezactivare discuri .....	36
RUTINA 46	— Scriere directă cu umplere cu zero .....	36
<b>ANEXA 1</b>	.....	37

## INTRODUCERE

Microblătonul COBRA este un calculator care la dorința utilizatorului poate configura în mașina BASIC-SPECTRUM sau CP/M.

La pornire, pe ecranul televizorului, se afișează emblema microsistemului și se așteaptă introducerea unei opțiuni.

### Opțiuni:

1. Încărcarea sistemului de operare CP/M:  
    prin apăsarea tastei „D”
2. Încărcarea interpretorului BASIC-SPECTRUM:  
    de pe caseta magnetică prin apăsarea tastei „C”  
    din memoria EPROM prin apăsarea tastei „B”

### Observații:

— încărcarea sistemului de operare CP/M presupune existența pe discul fișierului SYS.COM

— nu mai este necesar utilitarul SYSGEN, deoarece încărcarea sistemului de operare se face din fișier, deci multiplicarea acestuia se poate face cu unul din utilitarele DIP sau PIP

Precizări suplimentare privind organizarea și funcționarea sistemului de calcul pe cele două regimuri se pot obține consultând manualele: COBRA BASIC, COBRA CP/M.

C80K.SYS

C64K.SYS

Sau 1K.SYS

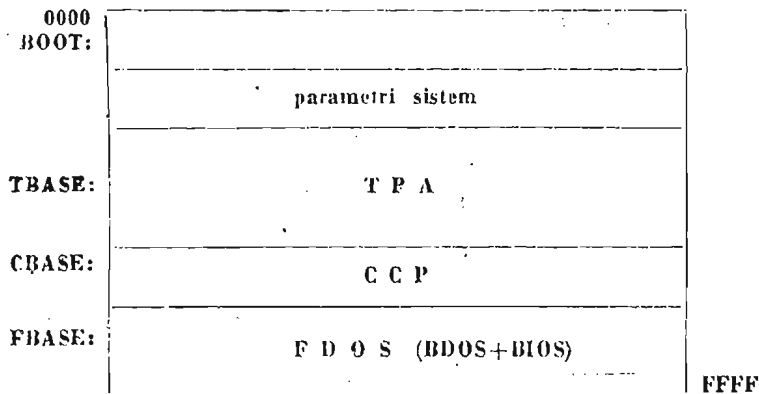
Acest manual descrie organizarea sistemului CP/M (inclusiv organizarea memoriei) și punctele de intrare în sistem. Se vor prezenta informațiile necesare pentru scrierea de programe executabile sub CP/M, programe ce utilizează facilitățile de I/E și de lucru cu discul oferite de sistem.

## 1.1 ORGANIZAREA CP/M

Sistemul CP/M este alcătuit din punct de vedere logic din următoarele patru părți:

- BIOS** — sistemul de I/E de bază, care oferă interfața cu perifericele
- BDOS** — sistemul de exploatare a discurilor, care oferă primitivele de acces la disc
- CCP** — procesorul de comenzi-consola
- TPA** — zona pentru programe tranzitorii

Componentele BIOS și BDOS sînt grupate într-un singur program numit **FDOS**, care are un punct de intrare unic. Componenta CCP este un program distinct, care utilizează programul FDOS pentru a oferi o interfață flexibilă între utilizator și informațiile existente pe disc. TPA este o zonă de memorie (i.e. zona de memorie care nu este utilizată de FDOS și CCP) în care se execută comenzile tranzitorii CP/M și programele-utilizator de aplicații. Organizarea memoriei într-un sistem standard CP/M este:



**REZERVAȚIE:** De obicei adresa BOOT este egală cu 0000H, adresa TPA este egală cu BOOT+0100H=0100H, iar adresele CBASE și FBASE depind de tipul sistemului CP/M.

Adresele 0000H — 0007H sînt rezervate pentru:

**0000H — 0002H** salt la rutina de reinițializare a sistemului CP/M existentă în BIOS (JMP WBOOT)

**0003H** octetul IOBYTE

**0004H** numărul utilizatorului curent și al discului instalat

**0005H — 0007H** salt la punctul de intrare în FDOS, respectiv în BDOS (JMP FBASE)

**0008H — 003FH** adresa de salt pentru instrucțiunile „RST n” neutilizate de sistem. Programele de depanare dinamică DDT, SID utilizează pentru punctele de intrerupere, instrucțiunea RST 7.

**0040H — 005BH** neutilizați de sistem.

**005BH — 007CH** blocul de control fișier implicit pregătit de CCP pentru programele încărcate în TPA.

**007DH — 00FEH** zona tampon de I/O implicită pentru operații cu discul (conține sectorul citit sau scris).  
 zona utilizată de CCP pentru a transmite programelor încărcate linia de comandă  
 zona utilizată ca stiva de lucru pentru programele executate în TPA.

**REZERVAȚIE:** a. Adresa 0005H este **PUNCT DE INTRARE** din programe tranzitorii în rutinele sistemului CP/M (în BDOS)

b. Adresa prezentă în locațiile 0006H — 0007H poate fi folosită pentru a determina dimensiunea maximă a memoriei disponibile (presupunind că se reacoperă componenta CCP)

c. Adresa 0003H este rezervată pentru octetul IOBYTE (configurația de I/E curentă)

d. Adresa 0004H este rezervată pentru a stoca numărul utilizatorului curent și numărul discului instalat; octetul de la această adresă are forma:

număr utilizator curent	număr disc instalat
0000 — 1111	0000 — 1111
	A—P

### 1.1.1. PARTICULARITĂȚI ALE SISTEMULUI CP/M — COBRA

Sistemul CP/M implementat pe microcalculatorul COBRA prezintă unele diferențe față de sistemul standard CP/M 2.2 elaborat de firma DIGITAL RESEARCH, dar este compatibil cu acesta în ceea ce privește formatul volumelor și fișierelor.

### 1.1.2. UTILIZAREA MEMORIEI INTERNE

Harta memoriei interne a sistemului **COBRA**:

FFFFH	BIOS	
FA00H	BDOS	
EC00H	CCP	
E400H	BIOS	
DE00H	RAM VIDEO	56882
C000H	GENERATOR CARACTERE BUFBERE	49152
A300H	TPA	
0100H	ZONA SISTEM	
000H		

### 1.1.3. ORGANIZAREA DISCULUI

Sistemul de operare CP/M implementat pe microcalculatorul **COBRA** recunoaște următoarele formate de disc:

- format standard 8" simplă densitate (compatibil **M18, M118, JUNIOR, CUBZ,...**)
- format standard 5 $\frac{1}{4}$ " dublă densitate 512 octeți/sector (compatibil **JUNIOR,...**)
- format **COBRA** — 8" simplă densitate (directory în pista 0)
  - 5 $\frac{1}{4}$ " dublă densitate (directory în pista 0)

Caracteristici:

- disc flexibil 8":
  - 77 piste/disc
  - 26 sectoare/pistă
  - 128 octeți/sector
- disc flexibil 5 $\frac{1}{4}$ "
  - dublă densitate (format **IBM**)
    - 40 piste/disc
    - 9 sectoare/pistă
    - 512 octeți/sector



— dublă densitate (format **COBRA**)

— 40 piste/disc

— 10 sectoare/pistă

— 512 octeți/sector

200ko

Alegerea unuia din formatele de mai sus se face folosind utilitarul **CDISK** prezentat în „Manual de utilizare” al microcalculatorului **COBRA**.

### 1.1.4 DISPOZITIVUL CONSOLĂ

O altă particularitate a microcalculatorului **COBRA** este faptul că el nu folosește ca dispozitiv consolă un terminal care să realizeze funcțiile de intrare, ieșire și de editare, așa cum este cazul celorlalte microcalculatoare din aceeași categorie, care au implementate sistemul de operare **CP/M**. Toate funcțiile de terminal amintite mai sus sunt realizate de un set de rutine specializate (dependente de mașină), implementate chiar în componenta **BIOS** a **CP/M**. Dăm în continuare o descriere a facilităților hard și a soluției soft adoptate, în vederea utilizării eficiente a unor elemente mai simple și mai ieftine, în locul unui terminal specializat: tastatura matricială **QWERTY** (în intrare), **TV** sau monitor **TV** alb-negru sau color (în ieșire).

#### 1.1.4.1 Tastatura

Este utilizată o tastatură tip **QWERTY** cu 6 linii  $\times$  8 coloane, deci 48 poziții independente, tastabile. Parte din aceste taste sînt dublate, triplate prin tastarea simultană a două taste (de ex. tasta **SHIFT** sau **CTRL** și încă o tastă), obținîndu-se astfel codurile literelor mici, ale semnelor grafice speciale și ale controalelor.

Deoarece spațiul de afișare are dimensiunea de 24 linii  $\times$  32 coloane, el este considerat numai ca o fereastră ce se mișcă lateral (stînga sau dreapta) în cadrul unui spațiu cu dimensiunea de 24 linii  $\times$  80 coloane, iar aceste deplasări pot fi comandate din tastatură. De asemenea mai poate fi comandată din tastatură și viteza de execuție a scrollului vertical și numărul de coloane cu care se deplasează fereastra în bufferul alfanumeric mare. Viteza de execuție a scrollului vertical este dată de faptul că acesta se realizează (opțional) pe 1, 2, 4 sau 8 linii **TV**. Iată aceste comenzi:

← scroll stînga 1 coloana

→ scroll dreapta 1 coloana

**GRAFICS** revenire în coloana 0

↓ la o tastare scade viteza cu o treaptă

↑ la o tastare crește viteza cu o treaptă

**NOSCROLL** se intrerupe afișarea pînă la relatare **NOSCROLL**

#### 1.1.4.2 Afișajul

Ca dispozitiv de afișare se folosește un aparat **TV** (obișnuit) alb-negru sau color. Procesul care realizează afișarea informațiilor pe ecranul **TV** este

pe scurt următorul: microprocesorul Z80 și componenta numită controller video, își dispută pe rând accesul la o zonă de memorie numită RAM VIDEO, controllerul video explorând în citire această zonă de memorie, afișează fiecare bit pe ecranul TV, de exemplu în convenția 0 punct stins, 1 punct aprins (aceasta la microcalculatoarele PRAE, AMIC sau ZX80).

Dimensiunile zonei afișate pe ecranul TV sînt de  $192 \times 256$  puncte (biți). Acest spațiu se consideră a fi împărțit în matrici de  $8 \times 8$  biți (puncte), deci în  $24 \times 32$  matrici de  $8 \times 8$  puncte. În continuarea acestei zone de memorie ce conține informație afișabilă, urmează o zonă de memorie ai cărei octeți se consideră în următoarea convenție: fiecare octet din această zonă condiționează binariv afișarea unei matrici de  $8 \times 8$  puncte din zona descrisă mai sus. Această condiționare se realizează prin următoarea structurare funcțională a biților din octeții acestei de a doua zone, numite memorie cu atribute:

F	B	Ink	Paper
7	6	543	210

În această convenție se depășește partajarea făcută mai sus în 0 punct stins, 1 punct aprins, ajungindu-se la următoarele: pe o matrice de  $8 \times 8$  puncte se pot defini două culori, una constituind fondul (hîrtia) reprezentată de biții de valoare 0 din matrice și careia i se atribuie culoare definită de biții 0-2 din octetul atribut, cealaltă culoare constituind scrisul (cerneala) reprezentată de biții de valoare 1 din matrice și careia i se atribuie culoarea definită de biții 3-5 din octetul atribut. Bitul 6 dă posibilitatea definiții a două nuanțe pentru fiecare culoare și se numește atribut de strălucire (0 normal, 1 cu strălucire), ridicînd deci la 16 numărul de culori. Cele opt culori selectabile se desemnează astfel:

- 000 negru
- 001 albastru
- 010 roșu
- 011 magenta
- 100 verde
- 101 cyan
- 110 galben
- 111 alb

Bitul 7 din octetul atribut validează funcționarea blinking (schimbarea alternativă a culorii fondului cu aceea a scrisului).

Pe ecranul TV, zona grafică de  $256 \times 192$  pixeli este încadrată de o margine numită BORDER a cărei culoare poate fi definită de utilizator în convenția de culoare de mai sus cu ajutorul funcției de terminal CTRL/B.

Modificarea culorii PAPER-ului și a INK-ului, modificarea stării de strălucire și blinking se face cu ajutorul funcției de terminal CTRL/C.

Iată în continuare harta zenelor de memorie descrise mai înainte:

adresa început	—	adresa sfîrșit	—	conținut
C000H		C7FFH		prima treime
C800H		CEFFH		a doua treime
D000H		D7FFH		a treia treime
D800H		DAFFH		zona cu atribute

Memoria grafică constituită din cele  $21 \times 32$  matrici de  $8 \times 8$  biți este utilizată întreg în zona C000H -- D7FH adresarea făcându-se cu următoarea convenție de adresare:

T	0	T1	T0	R2	R1	R0	L2	L1	L0	O4	O3	O2	O1	O0
---	---	----	----	----	----	----	----	----	----	----	----	----	----	----

unde R2 R1 R0 este o grupare de trei biți ce reprezintă numărul rîndului într-o linie de 32 de matrici de  $8 \times 8$  biți de afișare alfanumerică sau alfaneumerică, L2 L1 L0 este o grupare de trei biți ce reprezintă numărul liniei de matrice de  $8 \times 8$  biți (care în total sînt 21, T1 T0 este o grupare de doi biți ce reprezintă numărul treimii în care se află linia de matrice referită, iar O4 O3 O2 O1 O0 este o grupare de cinci biți ce reprezintă numărul matricii de  $8 \times 8$  biți (deci octetul ce reprezintă o linie de 8 biți din matrice) dintr-un rînd într-o linie cu pe o linie sau pe un rînd există 32 coloane). Precizăm faptul important că ordinea de avansare a acestor grupe de biți este următoarea: O, L, T; iar valorile limită ale acestor grupe de biți sînt: O (00000 -- 11111), L (000 -- 111), T (00 -- 10).

### 1.1.4.3 Funcțiile de terminal

Pe lângă funcția obișnuită de afișare a setului de caractere alfanumerice grafice, mai sînt implementate și alte funcții de terminal, care vor fi prezentate mai jos:

- ESC,1 = 1BH,31H -- poziționare absolută XY a cursorului
- ESC,3 = 1BH,33H -- comutare „wraparound”/blocare verticală
- ESC,4 = 1BH,34H -- comutare „wraparound”/blocare orizontală
- ESC,5 = 1BH,35H -- auto line-feed inactiv/activ
- ESC,A = 1BH,41H -- cursor în sus
- ESC,B = 1BH,42H -- cursor în jos
- ESC,C = 1BH,43H -- cursor la dreapta
- ESC,D = 1BH,44H -- cursor la stînga
- ESC,E = 1BH,45H -- ștergere ecran
- ESC,H = 1BH,48H -- cursor în poziția „home” (stînga sus)
- ESC,I = 1BH,49H -- regres linie cu defilare în jos
- ESC,J = 1BH,4AH -- ștergere pînă la sfîrșit de pagină
- ESC,K = 1BH,4BH -- ștergere pînă la sfîrșitul liniei
- ESC,L = 1BH,4CH -- inserare a unei linii vide
- ESC,M = 1BH,4DH -- eliminarea liniei curente
- ESC,N = 1BH,4EH -- activare afișare în invers video
- ESC,O = 1BH,4FH -- revenire la afișare în video direct
- ESC,R = 1BH,52H -- inserare spațiu în poziția curentă
- ESC,S = 1BH,53H -- ștergere a caracterului curent din linie
- ESC,Y = 1BH,59H -- poziționare absolută YX a cursorului

- CTRL/E = 05H -- eliminare a liniei curente
- CTRL/F = 06H -- inserare a unei linii vide
- CTRL/H = 08H -- cursor la stînga
- CTRL/I = 09H -- tabulare orizontală
- CTRL/J = 0AH -- LF avans linie
- CTRL/M = 0DH -- CR retur car
- CTRL/N = 0EH -- activare afișare în video invers
- CTRL/O = 0FH -- revenire la afișare în video direct

CTRL/R = 12H — inserare spațiu în poziția curentă  
 CTRL/U = 15H — cursor la dreapta  
 CTRL/V = 16H -- ștergere pînă la sfîrșitul liniei  
 CTRL/W = 17H — ștergere pînă la sfîrșitul paginii  
 CTRL/X = 18H — ștergere ecran  
 CTRL/Y = 19H — cursor în poziția „home” (stînga sus)  
 CTRL/Z = 1AH — cursor în sus  
 CTRL/9 = 1BH — ESC inițiază o secvență de escape  
 CTRL/B = 02H — setează culoarea BORDER-ului (acumulatorul va  
 conține codul culorii dorite pentru BORDER)  
 CTRL/C = 03h — setează octetul atribuit (ca mai sus)  
 CTRL/D = 04h — programează interfața RS232  
 DEL = 7FH — șterge caracterul precedent

#### 1.1.4.4 Interfața serială RS232

Interfața serială RS232 este implementată soft și deci nu este folosit nici un circuit specializat în acest scop. Aceasta e constituită din trei componente care asigură și funcțiile pentru care a fost implementată: programarea parametrilor interfeței, emisia și recepția. Utilizarea funcțiilor de emisie și recepție se face prin intrările (BIOS) CONIN, CONOUT, după modificarea în prealabil a octetului IOBYTE (corespunzător), sau numai în emisie prin intrarea LIST. Parametrii de funcționare ai interfeței se programează cu ajutorul utilitarului RS232.COM sau cu ajutorul funcției de terminal CTRL/D unde acumulatorul va conține un cod ce are următoarea semnificație:

p	p	x	b	b	v	v	v
---	---	---	---	---	---	---	---

unde: viteza vvv = 000 — 150 bauds  
                                   001 — 300 bauds  
                                   010 — 600 bauds  
                                   011 — 1200 bauds  
                                   100 — 2400 bauds  
                                   101 — 4800 bauds  
                                   110 — 9600 bauds  
                                   111 — 19200 bauds

nr. biți bb= 00 — 5 biți  
                                   01 — 6 biți  
                                   10 — 7 biți  
                                   11 — 8 biți

XONN-XOFF x = 0 — nu  
                                   1 — da

paritate pp = 00 — fără paritate  
                                   01 — paritate indiferentă  
                                   10 — paritate pară  
                                   11 — paritate impară

## 1.2 EXECUȚIA PROGRAMELOR TRANZITORII

Programele tranzitorii sînt comenzi tranzitorii CP/M și programe-utilizator de aplicații.

Orice program tranzitoriu se încarcă de pe disc în zona TPA și se execută după cum va fi prezentat în continuare.

Utilizatorul comunică cu componenta CCP (deci cu sistemul CP/M) prin introducerea, după fiecare prompter CP/M (">") a unei linii de comandă. Fiecare linie de comandă are una din următoarele forme:

- (1) comanda <CR>
- (2) comanda specificator-fișier <CR>
- (3) comanda specificator-fișier 1 specificator-fișier 2 <CR>

unde „comanda” este numele unei comenzi CP/M rezidente (ex: ERA, DIR, TYPE etc.) sau numele unei comenzi CP/M tranzitorii sau numele unui program-utilizator. Dacă „comanda” este numele unei comenzi CP/M, atunci această comandă este executată imediat. În caz contrar, CCP caută pe discul specificat (indicat înainte de comandă) sau pe discul instalat, un fișier cu numele:

### comanda. COM

Dacă un astfel de fișier este găsit, atunci se presupune că el reprezintă imaginea-memorie a unui program care se execută în zona TPA și care, implicit, se încarcă în memorie începînd de la adresa TBASE. Componenta CCP încarcă fișierul tip „.COM” de pe disc în memorie, începînd de la adresa TBASE și îi predă controlul printr-o instrucțiune de tip „CALL”. La sfîrșitul execuției programului, controlul poate reveni în CCP (printr-o instrucțiune de tip „RET”) sau în CP/M (printr-o instrucțiune „JMP BOOT”). Dacă se dorește ca la sfîrșitul execuției programului controlul să revină în CCP, atunci programul trebuie să nu suprascrie zona CBASE — FBASE. În caz contrar, programul poate să folosească memoria pînă la adresa FBASE — 1.

Dacă în linia de comandă există unul sau doi specificatori de fișier, atunci componenta CCP pregătește și unul sau două „blocuri de control fișier” (FCB), în zona de memorie rezervată pentru „parametri sistem”. Aceste FCB-uri sînt construite în formatul impus de FDOS pentru accesul la fișiere (vezi cap. 1.4).

Programele tranzitorii pot folosi:

- facilitățile CP/M de I/E pentru a comunica cu consola și cu dispozitivele periferice, precum și
- subsistemul de lucru cu discul, pentru accesul la fișiere rezidente pe acest suport.

Accesul din programe tranzitorii în sistemul de I/E al CP/M se face prin transmiterea către sistemul CP/M, prin punctul de intrare în FDOS (existent la adresa BOOT+0065H), a unui număr de rutină și a unei adrese pentru informații specifice rutinei. După execuția rutinei, FDOS întoarce o valoare care indică modul de desfășurare a operației (operație desfășurată corect sau modul de eroare (numeric), dacă aceasta a eșuat).

### 1.3 CONVENȚII PENTRU APELUL FUNCȚIILOR DE SISTEM CP/M

Sistemul CP/M pune la dispoziția utilizatorilor o serie de rutine care pot fi apelate în cadrul programelor tranzitorii. Rutinele se împart în două categorii:

- rutine de I/E pentru periferice simple
- rutine de I/E pentru lucrul cu fișiere pe disc

Rutinele de I/E pentru periferice simple sînt:

- citire caracter de la consolă
- scriere caracter la consolă
- citire/scriere directă la consolă
- citire caracter de la dispozitivul tip „READER”
- scriere caracter la dispozitivul tip „PUNCH”
- scriere caracter la dispozitivul tip „LIST”
- citire/modificare octet IOBYTE
- tipărire la consolă a unui șir de caractere
- citire buffer consolă
- citire stare consolă.

Rutinele de I/E pentru lucrul cu fișiere pe disc sînt

- creare fișier
- deschidere fișier
- închidere fișier
- căutare în „director”
- modificare nume fișier
- ștergere fișier
- citire secvențială sau directă a unui fișier
- scriere secvențială sau directă a unui fișier
- modificare atribute fișier
- inițializare „adresa DMA”
- inițializare stare sistem discuri
- s.a.

În ANEXA 1 este prezentată lista completă a rutinelor CP/M disponibile.

Accesul la rutinele FDOS se realizează prin transmiterea în:

- registrul „C” a numărului rutinei și
- perechea de registre „DXE” a unor parametri necesari rutinei.

Rutinele FDOS pot avea ca ieșiri valori pe un octet (în registrul „X”) sau pe doi octeți (în perechea de registre „HX”).

**OBSERVAȚII:** a. Pentru rutinele care au ca ieșiri valori pe doi octeți, aceste valori se găsesc și în registrele „A” și „B” (i.e. (A)=L și (B)=H)

- b. Convențiile de apel al rutinelor CP/M respectă standardele PL/M de comunicare parametri.
- c. Există rutine CP/M care folosesc doar registrul „E” pentru transmiterea unor parametri necesari apelului lor
- d. Există rutine CP/M care nu necesită parametri (apelul lor presupune doar transmiterea, prin registrul „C”, a numărului rutinei)
- e. Există rutine CP/M care nu au ieșiri.

Rezultă că secvența standard necesară pentru apelul unor rutine CP/M

mai:

```

BDOS      EQU      0005H

          MVI      C, număr rutină
          [ LXI    D, parametrii specifici rutinei ]
          [ MVI    E, parametru specific rutinei ]
          CALL    BDOS      ;apel rutină prin punctul de intrare
                               ; in FDOS
  
```

**OBSERVAȚIE:** Liniiile cuprinse între [ , ] reprezintă linii opționale, dependente de tipul rutinei.

S-a arătat în capitolul 1,2 că după încărcarea de pe disc, în memorie, în unui program tranzitoriu, componenta CCP îi predă acestuia controlul printr-o instrucțiune de tip „CALL”. Execuția programului tranzitoriu începe având SP-ul poziționat pe o stivă cu o capacitate de 8 nivele (16 octeți), în care există înscrisă doar adresa de revenire în CCP (7 nivele sînt încă libere). Deși această stivă nu este de obicei folosită de către programele tranzitorii (majoritatea acestora rezervîndu-și o stivă proprie și revenind în CCP printr-o instrucțiune de tip „JMP BOOT”), totuși este util de cunoscut faptul că ea este suficient de mare pentru a realiza apeluri de rutine CP/M. Acest lucru este posibil întrucît componenta FDOS la fiecare intrare într-o rutină de sistem, comută SP-ul pe o stivă locală, neafectînd astfel stiva inițială a programului. Programul în limbaj de asamblare de mai jos reprezintă un exemplu în acest sens, el realizînd citirea unor caractere de la consolă, pînă la întîlnirea unui caracter „\*” care determină întoarcerea controlului în CCP:

```

BDOS      EQU      0005H      ; punct de intrare standard in rutinele
                               ; CP/M
CONIN     EQU      1         ; rutina „Console Input”
          ORG      100H      ; adresa de bază pentru TPA

NEXTC

          MVI      C,CONIN   ; pregătire apel rutina CONIN
          CALL    BDOS      ; citire caracter de la consolă cu pre-
                               ; luarea caracterului în registrul „A”
          CPI     *          ; test pentru sfîrșit de prelucrare”
          JNZ     NEXTC     ; reluare prelucrare dacă nu e *
          RET                    ; revenire în CCP
  
```

#### 1.4. PARTICULARITĂȚI ÎN UTILIZAREA RUTINELOR CP/M DE LUCRU CU FIȘIERE PE DISC

Pentru lucrul cu discul flexibil sistemul CP/M implementează, pe fiecare volum disc, o structură de fișiere identificate prin nume. Fiecare unitate de disc este, din punct de vedere logic, distinctă, avînd o zonă rezervată pentru „director” și o altă zonă pentru fișierele de date. Fiecare fișier are asociat un identificator alcătuit din:

- codul pentru selectarea unității de disc (o literă A—P)
- numele (alcătuit din 1—8 caractere ASCII diferite de spațiu)
- extensia (tipul) fișierului (alcătuită din 0—3 caractere ASCII diferite de spațiu)

Extensiile definesc categoria generică din care face parte un anumit fișier. În timp ce numele identifică în mod unic fișierul în cadrul categoriei respective. Astfel, sistemul CP/M utilizează următoarele extensii standard:

- **ASM** pentru fișiere sursă în limbaj de asamblare tratabile cu asamblorul ASM sau MAC
- **PRN** pentru fișiere listing
- **HEX** pentru fișiere hexa
- **BAS** pentru fișiere sursă în limbaj BASIC
- **INT** pentru fișiere cod-obiect intermediar
- **COM** pentru fișiere cod-obiect direct executabil
- **BEL** pentru fișiere cod-obiect relocabil
- **COB** pentru fișiere sursă în limbaj COBOL
- **FOR** pentru fișiere sursă în limbaj FORTRAN
- **MAC** pentru fișiere sursă în limbaj de asamblare tratabile cu asamblorul M80
- **BAK** pentru fișiere ce reprezintă versiuni anterioare într-un proces de editare texte
- **\$\$\$** pentru fișiere temporare
- s.a.

Fișierele sursă sînt tratate ca o secvență de caractere ASCII, în care fiecare „linie” din fișier se termină prin secvența de caractere <CR> <LF> (ODH OAH). Astfel, o înregistrare CP/M (de 128 de octeți) poate conține mai multe linii de text sursă. Sfîrșitul unui fișier ASCII este indicat prin caracterul CTRL/Z (1AH) sau prin „sfîrșitul fizic” de fișier, detectat de către rutina CP/M de citire. Caracterele CTRL/Z existente într-un fișier cod-obiect (de exemplu, în fișiere tip COM) sînt ignorate, sfîrșitul de fișier fiind detectat de rutina CP/M de citire.

Orice fișier CP/M este o secvență de maximum 65536 înregistrări, de cîte 128 octeți fiecare, numerotate de la 0 la 65535. Deși din punct de vedere logic înregistrările într-un fișier sînt contigue totuși, fizic (pe disc) ele pot să nu fie contigue.

Fiecare fișier este, intern, împărțit în segmente de cîte 16 KB, denumite „extensii logice”. În cadrul fiecărei „extensii logice” există 128 de înregistrări (128 \* 128 B = 13 KB) numerotate de la 0 la 127 (00H—7FH). Se observă că în cadrul unei „extensii logice” contorul de înregistrări poate fi reprezentat pe 8 biți. Informațiile privind fiecare „extensie logică” a unui fișier ocupă spațiu în „directorul” discului respectiv. O „extensie logică” (16



Un bloc este formată din mai multe blocuri de alocare. Un bloc de alocare reprezintă spațiul disc minim ce poate fi alocat unui fișier. Un bloc de alocare este de minimum 1 KB și maximum 16 KB; dimensiunea blocului de alocare este stabilită la generarea sistemului CP/M.

Pentru utilizarea rutinelor CP/M de lucru cu-fișiere pe disc trebuie respectate următoarele convenții:

Informațiile de identificare a oricărui fișier se transmit către rutinele BIOS într-un format standard, și anume sub forma unui „bloc de control fișier” (File Control Block = FCB). Dimensiunea FCB depinde de tipul accesului la fișier (este de 33 de octeți pentru acces secvențial și de 36 de octeți pentru acces direct). Adresa FCB-ului se transmite în general prin registrele I/O.

— orice operație de citire/scriere date într-un fișier se realizează asupra unui buffer înregistrări de 128 de octeți.

— adresa de început a zonei de memorie (de 128 de octeți) utilizată ca buffer în operațiile de citire/scriere se numește „adresa DMA”. Această adresă nu se transmite ca parametru, ea fiind inițializată de către sistemul CP/M sau de către o rutină CP/M specială (rutina 26).

— se numește „disc selectat” acea unitate de disc care a fost activată prin:

— acțiunea componentei CCP (discul instalat prin CCP este în momentul lansării unui program în TPA „disc selectat”) sau

— prin rutina CP/M de selectare disc (rutina 14)

— se numește „disc activ” acea unitate de disc, care de la ultima inițiere/reinițializare a sistemului CP/M sau de la ultima operație de inițiere stare sistem discor (rutina 13), a făcut obiectul unei selecții:

— **explicite** (prin CCP sau prin rutina CP/M de selectare disc (rutina 14)) sau

— **implicit** (printr-o rutină de deschidere sau creare a unui fișier cu octet 00 din FCB diferit de zero).

**OBSERVAȚIE:** Sistemul CP/M folosește intern pentru FCB zona de memorie 005CH — 007FH (36 octeți), iar ca „adresă DMA” adresa 0080H (bufferul pentru operații de citire/scriere este de la adresa 0080H până la 00FFH (128 octeți)). Utilizatorul poate să folosească în programe aceste zone de memorie pentru FCB-ul, respectiv pentru bufferul propriu.

Structura standard a unui FCB este următoarea:

— octetul 00 — codul unității de disc pe care se găsește fișierul, respectiv:

00H — pentru discul selectat

01H — pentru unitatea „A”

10H — pentru unitatea „P”

- octeții 01—08 — numele fișierului exprimat în ASCII (caractere majuscule, cu B7=0); dacă numele fișierului are mai puțin de 8 caractere, atunci el trebuie completat la dreapta cu blanșuri.
- octeții 09—11 — extensia (tipul) fișierului exprimat în ASCII (caractere majuscule); dacă extensia are mai puțin de 3 caractere, atunci acest cimp trebuie completat la dreapta cu blanșuri.  
Dacă fișierul este protejat la scriere (R/O) atunci B7 din octetul 09 este egal cu 1; altfel, acest bit este egal cu 0.  
Dacă fișierul este invizibil (SYS) atunci B7 din octetul 10 este 1; altfel, acest bit este egal cu 0.
- octetul 12 — numărul curent al „extensiei logice” a fișierului; de obicei acest octet este setat de utilizator pe 00H.
- octetul 13 — rezervat pentru sistem
- octetul 14 — rezervat pentru sistem; acest octet este setat pe 00H atunci cînd se execută operații de tip OPEN, MAKE, SEARCH.
- octetul 15 — contor de înregistrări în cadrul „extensiei logice” curente (ia valori între 00H și 7FH); acest cimp este completat de către sistem.
- octeții 16—31 — rezervați pentru sistem (ei vor fi complețați de către sistem)
- octetul 32 — numărul înregistrării din „extensia logică” curentă; se folosește în accesul secvențial la fișiere; în mod normal acest octet este setat de către utilizator pe 00H la deschiderea fișierului.
- octeții 33—35 — reprezintă un parametru opțional folosit numai în accesul direct la fișiere. El indică numărul înregistrării de scris/citit (are valori între 0000H și FFFFH cu posibilitate de depășire în octetul 35). Octeții 33 și 34 reprezintă o valoare pe 16 biți cu partea cea mai puțin semnificativă în octetul 33 și cea mai semnificativă în octetul 34.

Fiecare fișier care trebuie exploatat prin CP/M trebuie să aibă un FCB propriu, care să furnizeze, pentru fiecare operație cu fișierul informații privind numele și alocarea acestuia. Orice acces la un fișier implică inițializarea de către utilizator (programator) a FCB-ului corespunzător, respectiv prin înscrierea în octeții 00—11 a specificatorului fișierului și prin umplerea cu 00H a restului de octeți (12—31/35). Informațiile din FCB-urile corespunzătoare fișierelor de pe un disc se găsesc înregistrate în „directorul” discului respectiv și sînt aduse în memoria internă înainte ca utilizatorul să înceapă lucrul asupra fișierului/fișierelor (vezi rutinele OPEN, MAKE). Copia din

Comanda a FCB-ului este actualizată pe măsură ce au loc operații asupra discului, iar la terminarea lucrului cu acesta ea este înregistrată pe disc (vezi CLOSURE).

Atunci când o comandă (program) se lansează în execuție prin:

Comanda specificator-fișier 1 <CR>

Comanda specificator-fișier 1 specificator-fișier 2 <CR>

Componenta CCP construiește (după cum s-a arătat în cap. 1.2) primii 16 octeți din două FCB-uri pornind de la specificatorul/specificatori de fișier în linia de comandă (după numele comenzii). Automat, CCP completează (dacă este cazul) numele și extensia fișierelor cu blankuri. Primul FCB este construit la adresa 005CH și poate fi folosit ca atare pentru operații ulterioare asupra fișierului „specificator-fișier”. Al doilea FCB este construit în octeții 16-31 din primul FCB (adică de la adresa 006CH) și trebuie să fie în altă zonă de memorie înainte de utilizarea lui. Dacă de exemplu, utilizatorul introduce comanda:

```
PROGRAMME B:XZOT Y:ZAP <CR>
```

Programul PROGRAMME.COM de pe discul instalat va fi încărcat în zona TPA, buclul de control de la adresa 005CH va fi inițializat astfel:

```
octetul 00 -- 02H (cod unitate „B”)
octeții 01-08 -- 'X'
octeții 09-11 -- 'ZOT'
octeții 12-15 -- 001H
octetul 16 -- 001H (cod disc selectat, care în acest caz este chiar
discul instalat)
octeții 17-24 -- 'Y'
octeții 25-27 -- 'ZAP'
octeții 28-31 -- 001H
```

**REMARKS:** Programatorul trebuie să salveze conținutul celui de-al doilea FCB (cei 16 octeți începând de la adresa 006CH) înainte de a deschide fișierul corespunzător primului FCB (de la adresa 005CH), întrucât prin deschiderea acestuia informațiile referitoare la cel de-al doilea fișier vor fi șterse (suprascrise) de către sistem (de către rutina OPEN). Dacă într-o linie de comandă CP/M nu apare nici un specificator-fișier atunci zonele 005DH-005FH și 006DH-007FH vor conține blankuri. Componenta CCP asigură automat transformarea minusculilor în majuscule.

O caracteristică importantă pentru utilizatorul a sistemului CP/M este că după recepționarea unei linii de comandă el păstrează la adresa 0080H un bufer pentru consolă, în care există o copie a conținutului liniei de comandă excepțind numele comenzii. Astfel, pentru exemplul precedent anterior, buferul de la adresa 0080H va avea următorul conținut:

```
octetul 00 -- 0EH (numărul de caractere utile din linia de comandă
excepțind numele comenzii)
```

```
octetul 01 -- 00H
octeții 02-08 -- 'B:XZOT'
```

octetul '09 =  
octeții 10-14 = 'Y.ZAP'

Este, și în acest caz, sarcina utilizatorului de a extrage informațiile din acest buffer, înainte de a executa orice operație asupra unui fișier, operație prin care această zonă este suprascrisă (inițial „adresa DMA” este egală cu 0080H, adică tocmai adresa de început a acestui buffer consolă).

### 1.5 PREZENTAREA RUTINELOR CP/M

**RUTINA 0: Reinițializare sistem CP/M  
(System Reset)**

**Intrări:**  
registru C: 00H

**Efect:** întoarce controlul din programul utilizator în CP/M; această funcție are același efect ca „JMP 0060H”.

**RUTINA 1: Citire caracter de la consolă  
(Console Input)**

**Intrări:**  
registru C: 01H

**Ieșiri:**  
registru A: codul unui caracter ASCII

**Efect:** preia un caracter de la consolă și-l transmite în registru „A”. Toate caracterele tipăribile și în plus <CR>, <LF> și <BS> (CTRL H) sînt transmise în ecou la consolă. De asemenea caracterul CTRL/I (TAB) mută cursorul în următoarea poziție de tabulare. Restul de caractere netipăribile nu sînt transmise în ecou la consolă. Rutina așteaptă un timp nelimitat pînă cînd se tastează un caracter la consolă.

**RUTINA 2: Scriere caracter la consolă  
(Console Output)**

**Intrări:**  
registru C: 02H  
registru E: codul caracterului ASCII

**Efect:** transmite la consolă caracterul specificat prin registru „E”. Caracterele „TAB” (CTRL/I) sînt expandate iar caracterul CTRL/S este

interpretat drept stop defilare. Reluarea defilării, după CTRL/S se face cu un  
altre caracter diferit de CTRL/C, care reinițializează sistemul CP/M.

**RUTINA 3: Citire caracter de la dispozitivul**  
„Reader” curent  
(Reader Input)

**Intrări:**

registru C: 03H

**Ieșiri:**

registru A: codul unui caracter ASCII

**Efect:** preia un caracter de la dispozitivul RDR; curent și-l depune în  
registru „A”. Rutina așteaptă un timp nelimitat preluarea caracterului de  
la RDR.

**RUTINA 4: Scriere caracter la dispozitivul**  
„Punch” curent  
(Punch Output)

**Intrări**

registru C: 04H

registru E: codul unui caracter ASCII

**Efect:** transmite la dispozitivul PUN; curent caracterul specificat prin  
registru „E”.

**RUTINA 5: Scriere caracter la dispozitivul**  
„List” curent  
(List Output)

**Intrări:**

registru C: 05H

registru E: codul unui caracter ASCII

**Efect:** transmite la dispozitivul LST; curent caracterul specificat prin  
registru „E”.

**RUTINA 6: Citire/Scriere directă la consolă**  
(Direct Console I/O)

**Intrări:**

registru C: 06H

registru E: — 0FFH (pentru citire de la consolă)  
— codul unui caracter ASCII (pentru scriere  
la consolă)

**Ieșiri:**

registru A: codul unui caracter ASCII sau  
octet de stare

Efect: dacă registrul „E” este egal cu 00FH, atunci rutina realizează  
00FH (fără ecou) la unu caracter de la consolă. Registrul „A” va conține  
codul caracterului ASCII introdus sau 00H, dacă nu s-a introdus nici un  
caracter.

**OBSERVAȚII:** Rutina nu așteaptă nelimitat introducerea unui caracter  
de la consolă (ea întoarce imediat (A)=00H, dacă în regi-  
strul de interfață al consolei nu există nici un caracter dis-  
ponibil). Este indicat ca utilizatorul să aștepte prin program  
introducerea unui caracter de la consolă.

Nu sînt active în acest caz, caracterele de editare ale siste-  
mului CP/M (ex: DEL, TAB, CTRL/R etc.).

Dacă registrul „E” conține codul unui caracter ASCII, atunci rutina  
realizează serierea la consolă a caracterului respectiv.  
Rutina 6 nu trebuie să fie folosită împreună cu alte rutine CP/M care  
realizează intrări/ieșiri cu consola (rutinele 1, 2, 9, 10 și 11).

**RUTINA 7: Citire octet IOBYTE**  
(Get I/O Byte)

Intrări:  
registrul C: 07H  
Ieșiri:  
registrul A: valoarea curentă în octetului IOBYTE

Efect: citește octetul de la adresa 0003H și îl plasează în registrul „A”.

**RUTINA 8: Modificare octet IOBYTE**  
(Set I/O Byte)

Intrări:  
registrul C: 08H  
registrul E: valoare pentru octetul IOBYTE

Efect: scrie conținutul registrului „E” la adresa 0003H, modificînd  
astfel configurația de I/E curentă.

**RUTINA 9: Tipărire la consolă a unui șir de caractere**  
(Print String)

Intrări:  
registrul C: 09H  
registrele D & E: adresa șirului de caractere

Efect: tipărește la consolă șirul de caractere ASCII a cărui adresă de  
început este specificată în registrele „D & E”. Tipărirea se termină atunci

nu a întâlnit caracterul „\$”. Rutina tratează caracterele TAB (CTRL/I) înalte, mutând cursorul în următoarea poziție de tabulare. La fel ca în rutina 9, se face verificare pentru caracterul CTRL/S (stop defilare).

**RUTINA 10: Citire buffer consolă**  
(Read Console Buffer)

**Intrări:**

registru C: 0AH  
registrele D & E: adresă buffer

**Efect:** rutina permite citirea unei linii introduse de la consolă cu transferul conținutului ei într-o zonă de memorie a cărei adresă de început este dată în registrele „D & E”.

O linie editată la consolă se consideră terminată atunci când s-a introdus caracterul <CR> sau caracterul <LF> sau atunci când s-a depășit capacitatea bufferului consolei specificată de utilizator în primul octet din buffer. Rutina aduce în bufferul a căruia adresă este dată în registrele „D & E” urătorul conținut:

- octetul 00 = numărul maxim de caractere din bufferul consolei (cu valori între 1 și 255); acest cimp este inițializat de către utilizator înaintea apelului rutinei 10
- octetul 01 = numărul de caractere introduse în linie (fără <CR> și <LF>)
- octeții 02-n = caracterele din linia de editare (c1, c2, c3, ..., cn)

**REMARKS:** Dacă numărul de caractere din linia de editare este mai mic decât numărul maxim de caractere din buffer, atunci după ultimul caracter citit din linia de editare (i.e. caracterul „cr”) și pînă la poziția corespunzătoare ultimului caracter posibil în buffer, vor exista (în buffer) o serie de caractere fără semnificație pentru utilizator (ele reprezintă un rest neinițializat din bufferul consolei). În timpul introducerii de la consolă a liniei sînt active, pentru corecții, caracterele de editare ale sistemului CP/M:

**RU/OUT/DEL** — șterge din bufferul de intrare și redă în ecou ultimul caracter introdus de la consolă

**CTRL/C** — reîncărcarea sistemului de operare

**CTRL/E** — indică sfîrșitul fizic al unei linii; cursorul se poziționează pe începutul liniei dar linia nu se transmite decît atunci cînd se tastează <CR>

**CTRL/H** — introduce în bufferul de intrare un caracter „back space” care are ca efect întoarcerea cursorului pe ecran cu o poziție în urmă

- CTRL/J** — este echivalent unui caracter <LF> și reprezintă sfârșitul unei linii
- CTRL/M** — este echivalent unui caracter <CR> și reprezintă sfârșitul unei linii
- CTRL/R** — tipărește la consolă pe linia imediat următoare conținutul curent al bufferului de intrare. Prin acest caracter se poate vizualiza conținutul curent al unei linii în care s-au efectuat corecții prin RUBOUT (DEL)
- CTRL/U** — șterge integral linia introdusă de la consolă
- CTRL/X** — identic cu CTRL/U

**RUTINA 11: Citire stare consolă (Get Console Status)**

**Intrări:**

registru C: 0BH

**Ieșiri:**

registru A: stare consolă

**Efect:** rutina verifică dacă s-a introdus un caracter de la consolă sau nu. Dacă în registrul de interfață al consolei există un caracter disponibil, atunci rutina întoarce în registrul „A” valoarea OFFH. În caz contrar, în registrul „A” se va afla valoarea 00H.

**RUTINA 12: Citire versiune sistem  
(Return Version Number)**

**Intrări:**

registru C: 0CH

**Ieșiri:**

registrele H & L: număr de versiune

**Efect:** rutina întoarce în registrele „H & L” o valoare egală cu numărul de versiune al sistemului CP/M sub care se lucrează, respectiv (H)=00H iar (L)=numărul de versiune (ex: 22H pentru versiunea 2.2).

**RUTINA 13: Inițializare stare sistem discuri  
(Reset Disk System)**

**Intrări:**

registru C: 0DH

**Efect:** rutina dezactivează logic toate unitățile de disc (le acordă atributul R/W), asignează ca disc selectat unitatea „A” și stabilește ca „adresa



adresa 0080H. Rutina poate fi folosită atunci când o aplicație necesită schimbări de volume disc fără a se reinițializa sistemul CP/M (prin CTRL/C)

**RUTINA 14: Selectare disc**  
(Select Disk)

Intrări:

registru C: 0EH  
registru E: număr unitate selectată

**Efect:** rutina desemnează unitatea specificată în registrul „E” ca „disc selectat”. Numărul unității de disc se specifică prin valorile: 00H pentru unitatea „A”, 01H pentru unitatea „B”,..., 0FH pentru unitatea „F”. În urma execuției rutinei, unitatea specificată în registrul „E” este trecută în starea „activ” (disc activ) care încarcă „directorul” volumului respectiv; unitatea rămîne în această stare pînă la o nouă inițializare sau reinițializare a sistemului CP/M sau pînă la o nouă operație de „inițializare stare sistem disc” (rutina 13). Dacă în timp ce o unitate este „activă” se fac schimbări de volume disc, atunci automat unitatea este desemnată de către sistem ca „0” (vezi și rutina 28).

**REZERVAȚIE:** Toate FCB-urile care au primul octet egal cu 00H se referă implicit la fișiere care se găsesc pe discul selectat.

**RUTINA 15: Deschidere fișier (Open File)**

Intrări:

registru C: 0FH  
registre D & E: adresa FCB

Ieșiri

registru A: octet de stare

**Efect:** rutina realizează activarea unui fișier care se găsește în „directorul” discului specificat prin octetul 00 din FCB și care aparține utilității curent. Adresa FCB-ului fișierului de deschis este dată prin registrele D & E.

Programul FDOS caută în directorul discului specificat o intrare identică cu valoarea octeților 1-12 din FCB.

**REZERVAȚIE:** În FCB octeții 12 și 32 trebuie setați de către utilizator pe 00H, înaintea apelului rutinei 15.

Dacă programul FDOS găsește o astfel de intrare, atunci informațiile în „director” corespunzătoare ei sînt copiate în octeții 15-31 din FCB, permițîndu-se astfel accesul la fișier pentru operații ulterioare de citire/seriere.

Rutina întoarce în registrul „A” o valoare 0-3, dacă operația de deschidere s-a efectuat corect și o valoare egală cu 255 (OFFH), dacă aceasta eșuat.

**REZERVAȚIE:** Programatorul nu trebuie să efectueze operații asupra unui fișier înainte să se realizeze corect deschiderea sa.

Există posibilitatea ca în cadrul FCB-ului, în octeții 1-11 să apară un specificator-multiplu de fișier, adică să apară caractere „?” (care înlocuiesc orice caracter în poziția respectivă). În acest caz, programul FDOS caută în „director” prima intrare care corespunde specificatorului-multiplu de fișier din FCB.

**OBSERVAȚIE:** Dacă fișierul deschis prin această rutină urmează să fie exploatat secvențial, începând cu primul său articol, atunci utilizatorul trebuie să seteze octetul 32 din FCB pe 00H (pentru ca prima citire/scriere să se aplice asupra primei înregistrări din fișier).

**RUTINA 16: Închidere Fișier**

(Close File)

**Intrări:**

registru C: 10H

registrele D & E: adresa FCB

**Ieșiri:**

registru A: octet de stare

Efect: rutina realizează reversul rutinelor 15 (OPEN) și 22 (MAKE). Astfel, presupunând că FCB-ul a cărui adresă este specificată în registrele „D & E” a fost activat anterior printr-o rutină de „deschidere fișier” (rutina 15) sau de „creare fișier” (rutina 22), rutina de „închidere fișier” înregistrează FCB-ul curent în „directorul” discului specificat, actualizând astfel intrarea din „director” corespunzătoare fișierului respectiv.

Rutina întoarce în registrul „A” o valoare egală cu 0-3 dacă operația de închidere s-a desfășurat corect sau o valoare egală cu 255 (OFFH) dacă numele fișierului din FCB nu a fost găsit în „director”.

Închiderea fișierelor care au fost exploatate doar în citire este opțională. Numai fișierele în care s-au efectuat operațiile de scriere trebuie închise (pentru a actualiza în „director” informațiile referitoare la acele fișiere).

Dacă în FCB-ul fișierului de închis apare un specificator multiplu (i.e. caractere „?”), atunci rutina va efectua căutarea în „director” așa cum face rutina 15:

**RUTINA 17: Caută în „director” prima intrare**

(Search for First)

**Intrări:**

registru C: 11H

registrele D & E: adresa FCB

**Ieșiri:**

registru A: octet de stare

Efect: rutina caută în „director” prima intrare care corespunde valorilor octeților 0-12 din FCB-ul a cărui adresă este dată în registrele „D & E”. Rutina întoarce în registrul „A” valoarea 255 (OFFH) dacă nu a găsit o astfel de intrare sau o valoare cuprinsă între 0-3 dacă a găsit-o. Dacă în „director”

... este găsită o intrare identică cu specificatorul fișierului din FCB, atunci  
... de memorie a cărei adresă este „adresa DA/A” va fi completată cu o  
... înregistrare de „director” (128 octeți) și anume: cu acea înregistrare din „di-  
... care conține intrarea respectivă. Adresa relativă a intrării, în cadrul  
... înregistrării de „director”, este egală cu (A) - 32 (i.e. conținut de registru „A”  
... pe stânga cu 5 biți sau „ADD A” de 5 ori). Programele de aplicații pot  
... pe baza acestei adrese relative, din bufferul care conține înregistra-  
... de „director”, informațiile din „director” relative la intrarea găsită.

Dacă FCB-ul conține un specificator-multiplu (i.e. apar caractere „?”  
... pozițiile 1-12), atunci rutina întoarce PRIMA intrare din „director” care  
... conține specificatorul. Dacă octetul 00 din FCB conține caracterul „?”,  
... atunci rutina întoarce **antimat prima intrare din „directorul” discului selectat**  
... diferent de numărul utilizatorului cărui îi aparține intrarea respectivă,  
... diferent de conținutul acestei intrări și indiferent dacă intrarea este ștersă  
... sau nu).

**RUTINA 16: Caută în „director” următoarea intrare**  
(Search for Next)

**Intrări:**

registru C: 12H

**Ieșiri:**

registru A: octet de stare

**Efect:** această rutină este similară rutinei 17 cu excepția faptului că  
... „directorul” discului specificat nu se investighează de la începutul său (ca în  
... celelalte rutine), ci se caută intrarea corespunzătoare FCB-ului înce-  
... put de la ultima intrare din „director” găsită.

Rutina întoarce în registru „A” (ca și rutina 17), valoarea 255 (OFFH),  
... dacă nu se mai găsește în „director” nici o intrare identică cu FCB-ul speci-  
... ficat.

**RUTINA 19: Ștergere fișier (Delete File)**

**Intrări:**

registru C: 13H

registrele D & E: adresa FCB

**Ieșiri:**

registru A: octet de stare

**Efect:** rutina realizează ștergerea unuia sau mai multor fișiere, speci-  
... ficat prin FCB-ul a cărui adresă este dată în registrele „D & E”. FCB-ul poate  
... conține un specificator-individual de fișier sau un specificator-multiplu de  
... fișiere (pot apărea caractere „?” în zona de nume sau de extensie a fișierului,  
... dar nu și în zona pentru numele unității de disc pe care se găsește fișierul --  
... așa cum se putea în rutinele 17 și 18).

Rutina întoarce în registru „A” valoarea 255 (OFFH), dacă fișierul/fi-  
... șierele specificate în FCB nu au fost găsite, și o valoare 0-3, dacă operația  
... de ștergere s-a efectuat normal.

**RUTINA 20: Citire secvențială**  
(Read Sequential)

**Intrări:**

registru C: 14H  
registre D & E: adresa FCB

**Ieșiri:**

registru A: octet de stare

**Efect:** presupunind că FCB-ul a cărui adresă este specificată în registrele „D & E” a fost activat printr-o rutină de „deschidere fișier” (rutina 15) sau de „creare fișier” (rutina 22), rutina „citire secvențială” realizează citirea din fișier a următoarei înregistrări de 128 de octeți și transferarea ei în memorie, într-o zonă a cărei adresă este „adresa DMA”. Numărul înregistrării din cadrul „extensiei logice” curente este specificat prin octetul 32 din FCB. După citire, valoarea acestui octet va fi automat incrementată cu 1. Dacă valoarea rezultată în octetul 32 depășește 127 (7FH) atunci, automat, următoarea „extensie logică” a fișierului este deschisă și octetul 32 ia valoarea 00H, fiind astfel pregătit pentru următoarea operație de citire. Dacă operația de citire s-a efectuat normal, atunci registrul „A” va avea valoarea 00H; în caz contrar, adică atunci când nu mai există date în fișier (s-a atins sfârșitul fișierului), registrul „A” va avea o valoare diferită de 00H.

**RUTINA 21: Scriere secvențială**  
(Write Sequential)

**Intrări:**

registru C: 15H  
registre D & E adresa FCB

**Ieșiri:**

registru A: octet de stare

**Efect:** presupunind că FCB-ul a cărui adresă este specificată în registrele „D & E” a fost activat printr-o operație de „deschidere fișier” (rutina 15) sau „creare fișier” (rutina 22) anterioară, rutina „scriere secvențială” realizează scrierea în fișier a unei înregistrări de 128 de octeți. Înregistrarea de scris este luată din memorie, de la o adresă egală cu „adresa DMA” și este plasată în fișier în poziția dată de valoarea octetului 32 din FCB (numărul înregistrării în cadrul „extensiei logice” curente). După scrierea înregistrării în fișier, conținutul octetului 32 din FCB este automat incrementat cu 1. Dacă în urma incrementării rezultă o depășire (o valoare mai mare de 127 (i.e. 7FH)) atunci, automat, este deschisă următoarea „extensie logică” a fișierului și octetul 32 din FCB este inițializat cu 00H, în vederea unor operații de scriere ulterioare. Operația de „scriere secvențială” poate avea loc și în cadrul unor fișiere deja create corect, caz în care înregistrările ce se scriu se vor suprapune peste cele existente, practic înlocuindu-le pe cele vechi.

Rutina întoarce în registrul „A” valoarea 00H dacă operația de scriere a decurs normal sau o valoare diferită de 00H, dacă operația de scriere a eșuat datorită lipsei de spațiu pe disc.

**RUTINA 22: Creare fișier**  
(Make File)

**Intrări:**

registru C: 16H  
registre D & E: adresa FCB

**Ieșiri:**

registru A: octet de stare

Efect: rutina are același efect ca și rutina „deschidere fișier” (rutina 15), cu excepția faptului că în acest caz, FCB-ul trebuie să conțină numele unui fișier care nu există în „directorul” discului specificat.

Programul FDOS creează intrarea din „director” corespunzătoare FCB-ului și inițializează atât FCB-ul cit și „directorul” discului, forțând lungimea intrării pe 0.

**REZERVAȚIE:** Programul trebuie să evite duplicarea numelor fișierelor în „director”, respectiv trebuie să se asigure că în „director” nu există un alt fișier cu nume identic cu cel al fișierului de creat. În acest scop, este indicat ca el să efectueze anterior rutinei 22 o operație de „ștergere fișier” (rutina 19).

Rutina 22 întoarce în registrul „A” o valoare 0-3 dacă operația s-a desfășurat normal sau o valoare 255 (OFFH) dacă nu mai există spațiu în „directorul” discului. Rutina 22 are ca efect secundar și activarea FCB-ului, astfel încât nu mai este necesară o operație ulterioară de „deschidere fișier”.

**RUTINA 23: Schimbare nume fișier**  
(Rename File)

**Intrări:**

registru C: 17H  
registre D & E: adresa FCB

**Ieșiri:**

registru A: octet de stare

Efect: rutina realizează schimbarea numelui unui fișier. Rutina utilizează FCB-ul adresat prin registrele „D & E” astfel:

- primii 16 octeți din FCB reprezintă numele vechi al fișierului
- ultimii 16 octeți din FCB reprezintă numele nou al fișierului
- octetul 00 din FCB reprezintă numele unității pe care se găsește fișierul de redenumit (octetul 16 din FCB trebuie să fie 00H).

Rutina întoarce în registrul „A” o valoare 0-3 dacă operația s-a desfășurat normal sau valoarea 255 (OFFH) dacă nu s-a găsit în „directorul” discului specificat un fișier cu nume identic cu cel al fișierului de redenumit.

**RUTINA 24: Citire vector de unități-disc active**  
(Return Log-in Vector)

Intrări:

registru C: 18H

Ieșiri:

registrele H & L, vectorul de unități-disc active

Efect: rutina analizează care din unitățile de disc A-P este „activă”, respectiv care din aceste unități a fost activată:

- explicit printr-o rutină de „selectare disc” (rutina 11)
- implicit printr-o operație de deschidere/creare fișier (cu valoare diferită de 00H în octetul 00 din FCB).

Pentru unitățile de disc active, rutina întoarce o valoare logică „1”, iar pentru cele care nu sînt active o valoare logică „0”. Bitul B0 din registrul „L” reprezintă starea unității „A”, iar bitul B7 din registrul „H” reprezintă starea unității „P”. Astfel, prin registrele „H&L.” (respectiv B & A) rutina întoarce un vector ce indică starea tuturor unităților A-P.

**RUTINA 25: Citire număr disc selectat**  
(Return Current Disk)

Intrări:

registru C: 19H

Ieșiri:

registru A: numărul discului selectat

Efect: rutina întoarce în registrul „A” numărul „discului selectat”. Acest număr este 00H pentru unitatea „A” și ... 0FH pentru unitatea „P”.

**RUTINA 26: Modificare „adresa DMA”**  
(Set DMA Address)

Intrări:

registru C: 1AH

registrele D & E: adresa DMA

Efect: rutina permite modificarea „adresei DMA”, adică a adresei bufferului de 128 octeți folosiți în operațiile de citire/scriere fișiere. În general, „adresa DMA” stabilită la inițializarea CP/M, la reinițializarea CP/M precum și după o operație de „inițializare stare sistem discuri” (rutina 13), este adresa 008011. Rutina permite comutarea acestei adrese pe orice altă adresă (dată în registrele „D & E”), permițînd astfel localizarea bufferului de 128 de octeți în orice zonă de memorie.

Rutina stabilește „adresa DMA” ca fiind egală cu adresa specificată în registrele „D & E”. Noua valoare pentru „adresa DMA” este valabilă pînă la:

- o inițializare sau reinițializare a sistemului CP/M
- un alt apel al rutinei 26
- o operație de „inițializare stare sistem discuri” (rutina 13)

**RUTINA 27:** Citire adresa vector de alocare  
(Get ADDR (Alloc))

Intrări:

registru C: IBIL

Ieșiri:

registrele H & L: adresa vectorului de alocare

**Efect:** rutina întoarce în registrele „H & L” adresa vectorului de alocare asociat discului selectat. Sistemul CP/M păstrează în memorie, pentru fiecare unitate „activă”, un vector de alocare. Acest vector poate fi folosit pentru a determina dimensiunea spațiului-disc rămas liber pe un volum (vezi comanda tranzitorie STAT).

**OBSERVAȚIE:** Informațiile cuprinse în vectorul de alocare asociat unei unități de disc care a fost desemnată ca R/O de către CP/M (în urma schimbării unui volum disc fără inițializarea sistemului CP/M sau fără o operație de „inițializare stare sistem discuri” (rutina 13)) pot fi false.

**RUTINA 28:** Setare atribut R/O pentru o unitate de disc  
(Write Protect Disk)

Intrări:

registru C: ICH

**Efect:** rutina desemnează temporar discul selectat ca disc R/O. Orice încercare de scriere pe acel disc, până la o inițializare sau reinițializare a sistemului CP/M sau până la o operație de „inițializare stare sistem discuri” (rutina 13), va produce mesajul:

BDOS ERR on d:R/O

**RUTINA 29:** Citire vector de unități R/O  
(Get Read/Only Vector)

Intrări:

registru C: IDI

Ieșiri:

registrele H & L: vectorul de unități R/O

**Efect:** rutina întoarce în registrele „H & L” un vector ce indică unitățile de disc care sînt desemnate ca R/O în acel moment. Bitul B0 din registrul „L” corespunde unității „A”, iar bitul B7 din registrul „H” corespunde unității „P”. O valoare logică „1” indică faptul că unitatea respectivă este R/O.

O unitate de disc devine R/O după un apel al rutinei 28 sau în urma schimbării volumului disc din acea unitate (sistemul CP/M). În acest caz, desemnează autonom unitatea respectivă ca R/O).

**RUTINA 30: Modificare atribute fișier**  
(Set File Attributes)

Intrări:

registru C: 1EH  
registre D & E: adresa FCB

Ieșiri:

registru A: octet de stare

**Efect:** rutina permite modificarea atributelor R/O și SYS ale unui fișier specificat în FCB-ul a cărui adresă este dată în registrele „D & E”. FCB-ul trebuie să conțină un specificator--individual de fișier. Noile atribute ale fișierului se specifică prin:

- bitul B<sub>7</sub> din octetul 09 din FCB („1” reprezintă fișier protejat la scriere (R/O))
- bitul B<sub>7</sub> din octetul 10 din FCB („1” reprezintă fișier invizibil (SYS))

Rutina caută în „director” o intrare care corespunde octeților 1—11 din FCB; comparația se face ignorând valorile biților B<sub>7</sub> din octeții 1—11 din FCB și din „director”. Dacă o astfel de intrare este găsită rutina modifică corespunzător intrarea din „director” corespunzătoare. Rutina întoarce în registrul „A” o valoare 0—3 pentru cazul în care operația s-a desfășurat corect sau o valoare egală cu 255 (OFFH) pentru cazul în care nu a fost găsită o astfel de intrare.

**RUTINA 31: Citire adresa „bloc de parametri disc”**  
(Get ADDR (Disk Params))

Intrări:

registru C: 1FH

Ieșiri:

registre H & L: adresa blocului de parametri al discului

**Efect:** rutina întoarce în registrele „H & L” adresa „blocului de parametri ai discului”, bloc care este rezident în BIOS. Această adresă poate fi folosită:

- pentru a extrage din zona respectivă informații privind parametrii discului (informații necesare pentru a fi afișate sau pentru a se realiza, pe baza lor, calcul)
- pentru a modifica, prin program, parametrii discului; de obicei programele de aplicații nu folosesc rutine 31 în acest scop.



**RUTINA 32: Citire/Modificare număr utilizator curent**  
(Set/Get User Code)

**Intrări:**

registru C: 20H

registru E: --0FFH (pentru citire număr utilizator curent)

— numărul utilizatorului curent (pentru modificarea numărului utilizatorului curent)

**Ieșiri:**

registru A: numărul utilizatorului curent (dacă (E) = -- 0FFH) sau nici o valoare (dacă (E) ≠ 0FFH)

**Efect:** rutina permite citirea numărului utilizatorului curent (dacă (E) = 0FFH) și întoarcerea acestui număr în registrul „A” sau modificarea numărului utilizatorului curent, în funcție de valoarea curentă a registrului „E” (modulo 16). Numărul utilizatorului curent variază între 00H și 0FH.

**RUTINA 33: Citire directă**  
(Read Random)

**Intrări:**

registru C: 21H

registre D & E: adresa FCB

**Ieșiri:**

registru A: octet de stare

**Efect:** rutina este similară rutinei „citire secvențială” (rutina 20) cu excepția faptului că nu se citește din fișier înregistrarea de 128 de octeți cu numărul specificat în octetul 32 din FCB, ci înregistrarea al cărei număr este dat în octeții 33 și 34 din FCB. Octeții 33 și 34 din FCB reprezintă o valoare pe 16 biți cuprinsă între 0000H -- 0FFFFH cu partea cea mai puțin semnificativă în octetul 33 și cea mai semnificativă în octetul 34. Octetul trebuie să fie 00H întrucât o valoare diferită de 00H indică o „depășire” în afara sfârșitului fișierului.

Citirea directă necesită în prealabil deschiderea „extensiei logice” cu numărul 0 a fișierului (prima „extensie logică” a fișierului), operație care se realizează prin rutina 15.

Dacă operația de citire directă s-a efectuat corect, atunci:

— registrul „A” va avea valoarea 00H

— înregistrarea citită din fișier se va găsi depusă în memorie la „adresa

MA”

— valorile octeților 12 (numărul „extensiei logice” curente) și 32 (numărul înregistrării în cadrul „extensiei logice” curente) vor fi automat modificate în funcție de numărul înregistrării citite (octeții 33 și 34)

— valoarea octetului 32 nu va fi incrementată cu 1 (ca în rutina 20)

**REZERVAȚII:** 1. După o operație de „citire directă” pot fi folosite operații de „citire secvențială”/„scriere secvențială”. Programatorul însă trebuie să țină cont de faptul că prima operație de „citire secvențială”/„scriere secvențială” se va aplica asupra aceleiași înregistrări care s-a preluat prin „citire directă” (întrucât octetul 32 nu a fost incrementat cu 1). Se poate însă, printr-o „citire secvențială” falsă incrementa, octetul 32 din FCB, astfel încât operațiile de „citire/scriere secvențială” următoare să se aplice asupra înregistrărilor care urmează celei preluate prin „citire directă”.

2. Dacă operația de „citire directă” s-a aplicat asupra ultimei înregistrări dintr-o „extensie logică”, nu se realizează automat deschiderea „extensiei logice” următoare (ca în rutina 20)....

Dacă operația de „citire directă” nu s-a efectuat corect, atunci registrul va conține codul de eroare, respectiv:

- 01 citirea unei înregistrări nescrise
- 03 imposibilitate de închidere a „extensiei logice” curente (trebuie redeschisă sau recitită „extensia logică” numărul 0 a fișierului)
- 04 acces la „extensie logică” a fișierului care nu a fost creată
- 06 octetul 33 este diferit de 0011 (încercare de căutare în afara limitelor fizice ale fișierului)

În general, codurile de eroare diferite de 0011 pot fi interpretate ca „psă de date”.

<b>RUTINA 34: Scriere directă</b>	
<b>(Write Random)</b>	
<b>Întrări:</b>	registrul C: 22H registrele D & E: adresa FCB
<b>Ieșiri:</b>	registrul A: octet de stare

**Efect:** rutina este identică cu rutina de „citire directă” cu excepția faptului că o înregistrare de 128 de octeți aflată în memorie la „adresa DMA” de scrisă pe disc. Înregistrarea se va scrie în fișier în poziția corespunzătoare numărului ei (octeții 33 și 34). Dacă acesteia poziții nu îi fusese alocat încă, atunci rutina, înainte de scriere, realizează această alocare.

**REZERVAȚII:** Dacă înregistrarea de scris nu are un număr astfel încât ea să fie prima înregistrare dintr-un bloc nealocat, atunci rutina va ocupa fictiv toate înregistrările, anterioare înregistrării de scris, din blocul respectiv de alocare. Înregistrările ocupate fictiv (care sînt de fapt „găuri” în fișierul respectiv) vor fi contabilizate în „contorul de înregistrări” din cadrul „extensiei logice” curente (octet prezent

in intrarea de „director“), dar vor avea un conținut aleator (vezi și rutina 40).

La urma unei operații de „scriere directă“ valoarea octeților 12 și 32 se schimbă, dar octetul 32 nu se incrementază cu 1. Toate observațiile referitoare la rutina 33 sînt valabile și pentru rutina 34.

Dacă operația de „scriere directă“ s-a efectuat corect, atunci registrul va avea valoarea 0011; în caz contrar el va conține codul de eroare. Codurile de eroare posibile sînt cele de la rutina 33 plus codul 05 care indică lipsita scrierii datelor intrușit nu a mai fost spațiu în „director“ pentru crearea unei noi „extensii logice“.

**RUTINA 35: Determinare lungime fișier**  
(Compute File Size)

**Intrări:**

registru C: 23H  
registre D & E: adresu FCB

**Ieșiri:**

lungimea virtuală a fișierului  
(în octeții 33, 34 și 35 din FCB)

Scopul: rutina necesită ca FCB-ul adresat prin registrele „D & E“ să conțină de octeți și să conțină un specificator-individual de fișier. Rutina caută în „director“ informații privind fișierul specificat în FCB și completează octeții 33, 34 și 35 cu o valoare egală cu numărul corespunzător primei înregistrări de pe disc care urmează după sfîrșitul fizic al fișierului. Astfel, octeții 33 și 35 reprezintă „lungimea fișierului“, lungime care poate fi:

lungime reală (fizică) a fișierului (dacă fișierul a fost creat (scris) secvențial);

lungime virtuală a fișierului (dacă fișierul a fost creat în acces direct și există „găuri“ în alocarea fișierului)

Dacă octetul 35 are valoare egală cu 0111 atunci, rezultă că fișierul conține numărul maxim de înregistrări posibile (și anume 65535).

Rutina poate fi folosită pentru a adăuga înregistrări într-un fișier. Prin apelul ei se determină numărul de ordine al primei înregistrări care urmează de după sfîrșitul fizic al fișierului, număr de ordine ce poate fi folosit pentru continuarea de către o secvență de operații de „scriere directă“.

**RUTINA 36: Determinare număr înregistrare**  
(Set Random Record)

**Intrări:**

registru C: 24H  
registre D & E: adresa FCB

**Ieșiri:**

numărul înregistrării

Scopul: rutina întoarce în octeții 33, 34 și 35 din FCB numărul înregistrării curente dintr-un fișier care a fost citit/scris secvențial. Rutina poate fi folosită astfel:

— pentru determinarea numărului de ordine al unor înregistrări dintr-un fișier, care conțin o anumită „cheie”. În acest caz, fișierul se parcurge (în citire) secvențial, se verifică dacă înregistrarea citită conține „cheia” căutată și dacă o conține, atunci se apelează rutina 36 pentru a determina „numărul de ordine” al înregistrării respective. Acest număr de ordine se stochează și apoi se continuă investigarea (secvențială) a fișierului. La sfârșitul prelucrării se va dispune de o listă a tuturor numerelor înregistrărilor care conțin „cheia” respectivă. Pe baza acestei liste, utilizatorul poate citi direct înregistrările care îl interesează.

— atunci când se dorește trecerea de la accesul secvențial într-un fișier la accesul direct. În acest caz, după ce un fișier a fost explorat secvențial până la un anumit punct, se apelează rutina 36 pentru a determina „numărul de ordine” al înregistrării curente. Pe baza acestui număr de ordine se pot realiza, în continuare, operații de citire/scriere directă (operații ce se aplică de la un anumit punct (selectat) din fișier în continuare).

**RUTINA 37: Dezaactivare discuri  
(Reset Drive)**

**Intrări:**

registru C: 25H

**Ieșiri:**

registru A: 00H

**Efect:** dezactivează unitățile de disc specificate în vectorul definit prin conținutul registrelor „D & E” și acordă acestor unități atributul R/W. Bitul B0 din registrul „E” corespunde unității „A” iar bitul B7 din registrul „D” corespunde unității „P”. O valoare logică „1” în vectorul definit reprezintă o opțiune-utilizator de „dezactivare” a unității respective.

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

**OBSERVAȚIE:** Discul selectat nu poate fi dezactivat prin această rutină ci numai printr-o rutină 13.

**RUTINELE 38 și 39: Aceste rutine nu au nici un efect  
în această versiune de sistem**

**RUTINA 40: Scriere directă cu umplere cu 0  
(Write Random With Zero Fill)**

**Intrări:**

registru C: 28H

registrele D&E: adresa FCB

**Ieșiri:**

registru A: octet de stare

**Efect:** rutina este similară rutinei 34 (scriere directă) cu excepția faptului că înainte de a se scrie o înregistrare, într-un bloc nealocat, acesta este automat umplut cu zerouri. Astfel, toate „găurile” dintr-un fișier creat în acces direct vor fi recunoscute prin conținutul lor (zerouri).

A N E X A 1

Nr. rut (hex)	Denumire rutina	Intrări	Ieșiri
0	1	2	3
0	Reinițializare sistem C/P/M	C=00H	
1	Citire caracter de la consolă	C=01H	A=caracter ASCII
2	Scriere caracter la consolă	C=02H E=caracter ASCII	
3	Citire caracter de la dispozitivul „Reader” curent	C=03H	A=caracter ASCII
4	Scriere caracter la dispozitivul „Punch” curent	C=04H E=caracter ASCII	
5	Scriere caracter la dispozitivul „List” curent	C=05H E=caracter ASCII	
6	Citire/Scriere directă la consolă	C=06H E=0FFH =caracter ASCII	A=caracter ASCII sau =octet stare
7	Citire octet IOBYTE	C=07H	A=valoare octet IOBYTE
8	Modificare octet IOBYTE	C=08H E=valoare pentru octetul IOBYTE	
9	Țigărire la consolă a unui șir de caractere	C=09H D & E=adresa șir	
0A	Citire buffer consolă	C=0AH D & E=adresa buffer	
0B	Citire stare consolă	C=0BH	A=stare consolă
0C	Citire versiune sistem	C=0CH	H&L=număr de versiune
0D	Inițializare stare sistem discuri	C=0DH	
0E	Selectare disc	C=0EH E=număr unitate selectată	
0	Deschidere fișier	C=0FH D & E=adresa FCB	A=octet stare
10	Închidere fișier	C=10H D & E=adresa FCB	A=octet stare
11	Caută în „director” prima intrare	C=11H D & E=adresa FCB	A=octet stare
12	Caută în „director” următoarea intrare	C=12H	A=octet stare
13	Ștergere fișier	C=13H D & E=adresa FCB	A=octet stare
14	Citire secvențiale	C=14H D & E=adresa FCB	A=octet stare
15	Scriere secvențiale	C=15H D & E=adresa FCB	A=octet stare
16	Creare fișier	C=16H DE= adresa FCB	A=octet stare

0	1	2	3
17	Schimbare nume fișier	C=17H D & E=adresa FCB	A=octet stare
18	Citire vector de unități disc active	C=18H	H & L=vectorul de unități disc active
19	Citire număr disc selectat	C=19H	A=număr disc selectat
1A	Modificare adresa DMA	C=1AH D & E=adresa DMA	
1B	Citire adresa vector de alocare	C=1BH	H & E=adresa vector de alocare
1C	Setare atribut P/O pentru o unitate de disc	C=1CH	
1D	Citire vector de unități R/O	C=1DH	H & L=vectorul de unități R/O
1E	Modificare atribute fișier	C=1EH D & E=adresa FCB	A=octet stare
1F	Citire adresa bloc de parametri disc	C=1FH	H & L=adresa blocului de parametri disc
20	Citire/Modificare număr utilizator curent	C=20H E=0FFH număr utilizator curent	A=număr utilizator sau nimic
21	Citire directă	C=21H D & E=adresa FCB	A=octet stare
22	Scriere directă	C=22H D & E=adresa FCB	A=octet stare
23	Determinare lungime fișier	C=23H D & E=adresa FCB	lungime virtuală în octeții 33, 34 și 35 din FCB
24	Determinare număr înregistrare	C=24H D & E=adresa FCB	numărul înregistrare
25	Dezactivare discul	C=25H	A=00H
26	NEIMPLEMENTATĂ		
27	NEIMPLEMENTATĂ		
28	Scriere directă cu umplere cu zero	C=28H	A=octet stare

Editat de I.T.C.I. Braşov

---

Bun de tipar: 25 V 88

---



Tiparul executat sub comanda nr. 1205

Intreprinderea Poligrafică Braşov