

ARHITECTURA CALCULATOARELOR



Tematica disciplinei Arhitectura Calculatoarelor:

Tema 1. Introducere. Bazele fundamentale ale Arhitecturii Calculatoarelor;

Tema 2. Microprocesoare și Microcontrolere. Limbajul de programare Assembler;

Tema 3. Dispozitive pentru achiziția datelor;

Tema 4. Dispozitive pentru afișarea și imprimarea datelor;

Tema 5. Dispozitive pentru stocarea datelor.

Tema 5. Dispozitive pentru stocarea datelor :

1. Clasificarea dispozitivelor pentru stocarea datelor.
2. Sisteme de fișier: FAT, NTFS, EXTx.
3. Funcții BIOS și DOS pentru managementul datelor.
4. FDD. Programarea operațiilor de intrare/ieșire.
5. HDD. Programarea operațiilor de intrare/ieșire.
6. CD ROM.
7. Memorii externe: Flash, SD Card.
8. Tehnologii RAID.

Clasificarea suportului și a dispozitivelor pentru stocarea datelor:

Tehnologia de înregistrare a datelor: Deformare, Găurire, Magnetică, Electrică, Optică;

Tehnologia de citire a datelor: Contact electric, Optice, Magneto-inductivă; Electrică;

Suportul fizic: Hârtie, Bandă magnetică, Cartele magnetice, Discuri magnetice / optice, DSC;

Modul de accesare a datelor: Secvențial, Direct;

Durata pastrării datelor: Depinde de suport și modul de exploatare;

Modul de interacțiune suport – mediu: Deformare suport, Acțiune câmp ele./magn.;

Volum de date: Zeci de Octeți – Zeci de TB.

Sistemul de Fișier ca parte componentă a Sistemului de Operare (SO). Funcțiile SO:

Un **Sistem de Operare SO/OS**, este un ansamblu de programe care are rolul de a gestiona și de a facilita utilizatorului accesul la resursele sistemului de calcul. Metaforic, se poate spune că un SO joacă rol de dirijor, orchestrând doleanțele utilizatorilor prin atribuirea de partituri adecvate către diversele resurse ale calculatorului și veghind la interpretarea armonizată a respectivelor partituri, astfel încât scopurile pentru care s-a apelat la calculator să fie atinse. Pentru aceasta, nucleul sistemului de operare este primul program încărcat în memoria calculatorului și rămâne activ pe toată durata funcționării acestuia.



Fig. 6.1

Interacțiunea și funcțiile Sistemului de Operare (SO):

Funcțiile Generale ale SO:

- DOS & DOS;
- Controlează distribuția resurselor unui calculator;
- Mediază comunicarea dintre hardware și aplicațiile utilizatorilor.

Structura SO:

- nucleul (kernelul) este un mediu în cadrul căruia pot rula alte aplicații;
- - gestionează resursele fizice;
- - oferă o interfață comună utilizatorului / aplicațiilor;
- programele de bază, Drivere;
- - permit interacțiunea cu nucleul și cu sistemul fizic;
- exemple: interpretoare de comenzi, utilitare de gestiune a utilizatorilor și a sistemului de fișiere, biblioteci, etc.

Exemple de SO:

generaliste (general purpose)

- Windows (Microsoft)
- Mac OS X (Apple)
- GNU/Linux
- FreeBSD, OpenBSD, NetBSD

pe dispozitive mobile (mobile, embedded)

- iOS
- Android
- Symbian

pe dispozitive de rețea (Network Operating Systems)

- IOS (Cisco)
- JUNOS (Juniper)

...și altele

Funcțiile de administrare ale SO:

administrarea resurselor hardware

- procesor, memorie, hard disk, comunicare în rețea, dispozitive de intrare / ieșire
- concurența utilizatorilor
- concurența aplicațiilor – inclusiv a SO însuși

portabilitate

- o aplicație poate rula pe sisteme fizice diferite

comunicarea cu dispozitive periferice

- drivere

gestiunea utilităților de diagnosticare și monitorizare

Responsabilitățile puse pe SO:

gestiunea procesorului

gestiunea memoriei

gestiunea echipamentelor și perifericelor

gestiunea sistemelor de stocare a datelor

medierea: interfața cu aplicațiile și cu utilizatorii (system API)

utilitare: aplicații pentru utilizator

Funcțiile detaliate ale SO:

- controlul execuției programelor (încărcare în memoria internă, lansarea în execuție, încheierea execuției acestora),
- planificarea și controlul fluxului aplicațiilor,
- detectarea și păstrarea controlului erorilor ce apar în cursul prelucrărilor,
- administrarea spațiului memoriei interne și externe,
- gestionarea sistemului de fișiere,
- efectuarea operațiilor de intrare-ieșire,
- gestionarea periferiei calculatorului,
- oferirea unui set de comenzi ce să permită utilizatorului accesul optim la resursele calculatorului
- asigurarea interfeței cu utilizatorul.
- asigurarea securității, prin controlarea accesibilității utilizatorilor la resursele sistemului de calcul.

Procesul de încărcare a SO:

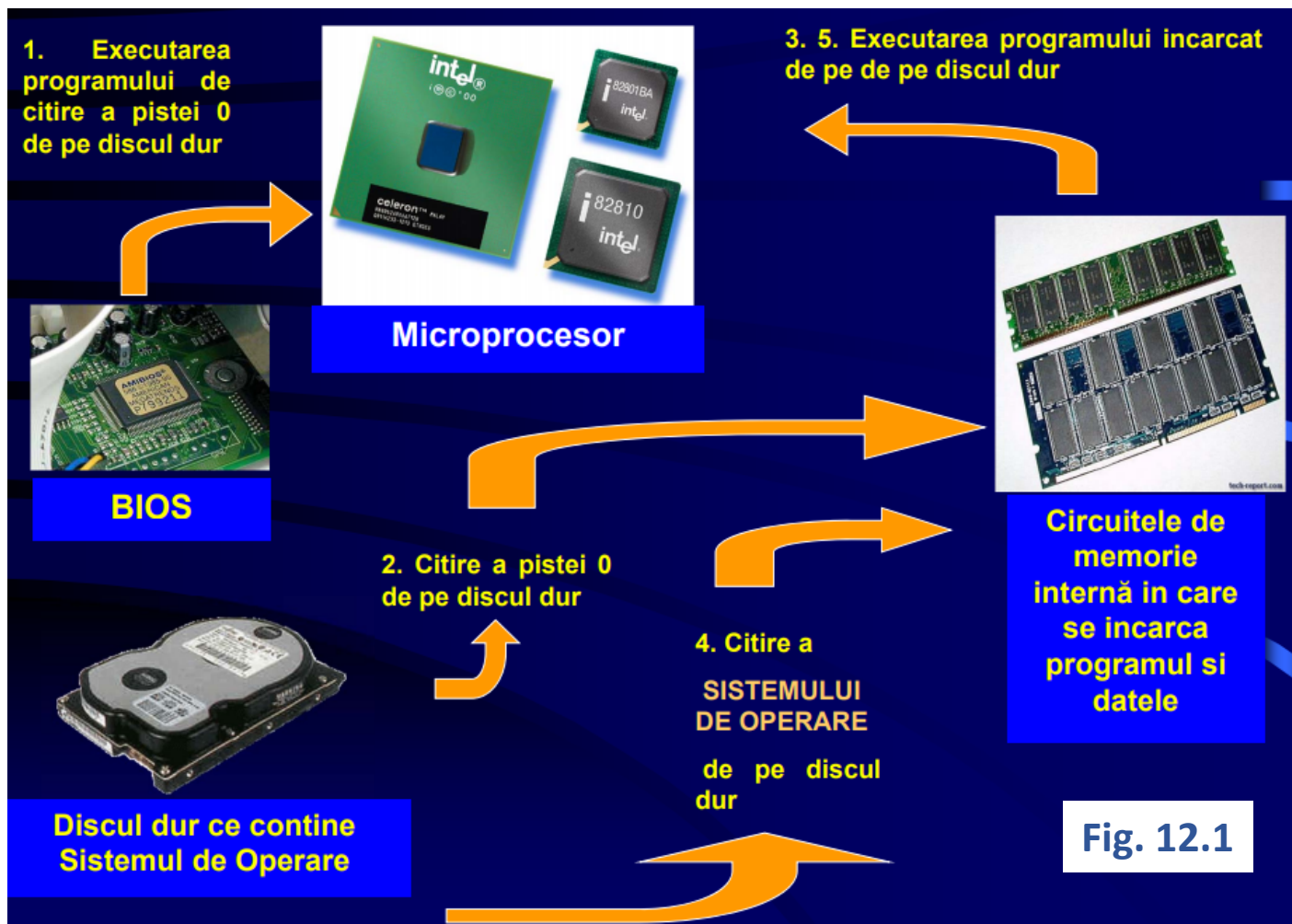


Fig. 12.1

1. BIOS: Executarea programului de citire a pistei 0 de pe discul de sistem;
2. HDD: Citirea pistei 0 de pe discul de sistem;
3. RAM: Executarea programului de încărcare a SO;
4. HDD: Citirea SO de pe discul de sistem;
5. RAM: Executarea programului Sistemul de Operare.

Interacțiunea componentelor Hard și Soft :



Fig. 13.1

Clasificarea și caracteristica SO:

- **multi-user:** Permit ca doi sau mai multi utilizatori să ruleze în același timp programe (utilizatori concurenți). Anumite sisteme de operare permit sute sau chiar mii de utilizatori concurenți.
- **multiprocesor:** Permit executia unui program pe mai mult de un microprocesor.
- **multitasking:** Permit mai multor programe să ruleze în același timp (executie concurentă).
- **timp real (real time):** Răspund instantaneu la diferite intrări. Sistemele de operare de uz general, nu sunt sisteme de operare de timp real. Calculatoarele ce conduc procese industriale utilizează acest tip de sistem de operare.

Definiția de Fișier:

Putem defini fișierul ca forma sub care regăsim informația, pe un suport nevolatil.

Din punct de vedere al conținutului fișierul este în general omogen, atât ca organizare cât și ca informație.

Pentru identificare, fișierul are un nume. Numele de fișier se supune unor restricții privind numărul de caractere, felul lor (litere mari, litere mici, cifre, semne speciale).

Extensia numelui fișierului este o terminație, separată de nume prin caracterul punct, care ne codifică tipul de fișier. Codificarea ne este necesară pentru a ști cu ce program să folosim fișierul.

Deschiderea unui fișier înseamnă decodificarea modului în care a fost pusă informația (proprie fiecărui program în parte).

Definiția de Director/Folder/Mapă:

Notiunea de director (folder)



Fig. 16.1

Structura arborescentă

În arborele informatic ramurile vor fi denumite **directoare (foldere)** și nu vor face altceva decât să grupeze fișiere. Cum ramurile mai tinere cresc din cele mai în vârstă, așa și în arborele informatic vom avea directoare derivate din alte directoare, ele fiind **subdirectoare** ale directorului din care provin.

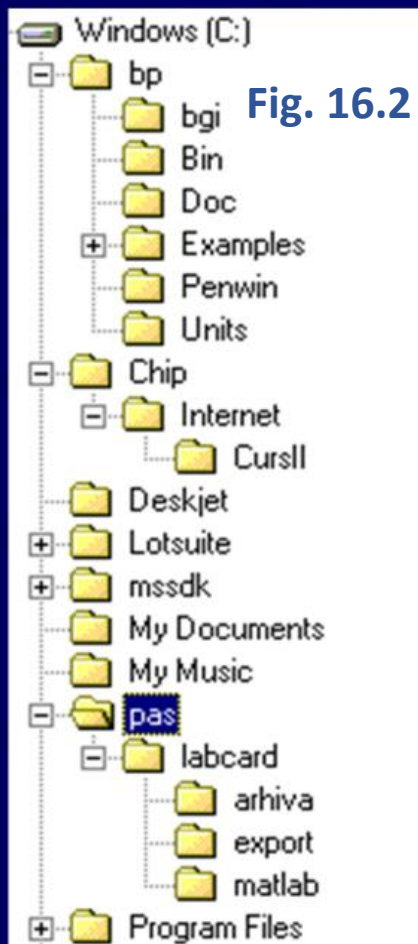


Fig. 16.2

Un director este o structură de catalogare a unui Sistem de Fișiere care conține referințe la alte fișiere și/sau alte directoare. În unele cazuri se asociază cu dosare sau foldere.

Fișierele sunt organizate prin stocarea fișierelor înrudite în același director.

Un director conținut în interiorul altui director se numește subdirector.

Termenii părinte și copil sunt adesea folosiți pentru a descrie relația dintre un subdirector și directorul în care acesta este catalogat, cel din urmă fiind părintele. Directorul din vârf al structurii ierarhice se numește directorul rădăcină (în limba engleză *root*).

Componentele fundamentale ale DOS:

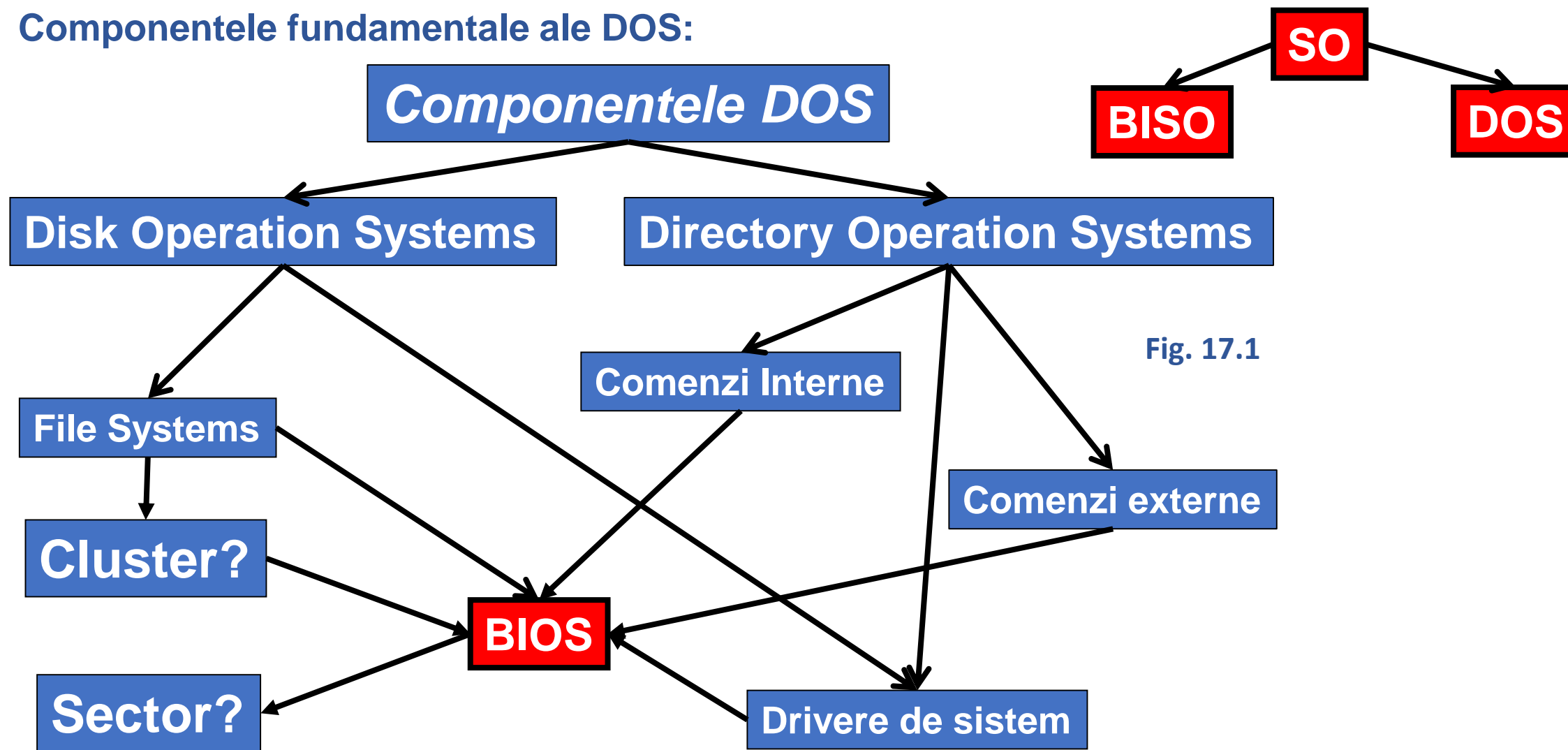


Fig. 17.1

Sistemul de Fișiere:

Sistemul de fișiere (*filesystem*) desemnează organizarea internă a unui purtător de date, care de obicei este cuplat la un calculator. Pe parcursul vieții lor, fișierele sunt supuse multor operații: ele trebuiesc create, denumite, regăsite, renumite, citite, modificate, duplicate, deplasate, reorganizate, defragmentate, șterse ș.a. În tot acest timp este nevoie și de o corespondență clară între numele unui fișier, util în primul rând omului, și adresa sa de pe purtătorul de date, necesară calculatorului. Regăsirea rapidă a unui fișier este și ea importantă, în condițiile în care un singur purtător de date actual poate găzdui chiar și sute de mii de fișiere. Pentru a asigura viteze de funcționare mulțumitoare, organizarea internă a stocării și accesării fișierelor unui purtător de date trebuie să țină cont și să profite de caracteristicile acestuia. Programele necesare pentru operațiile cu fișiere fac de obicei parte din sistemul de operare al calculatorului la care sunt cuplate.

FileSystems: FAT12, FAT16, FAT32, NTFS, EXT2, EXT3, EXT4, ISO 9660

Cele mai cunoscute Sisteme de Fișiere:

- **EXT2** (second extended file system) este folosit de majoritatea sistemelor de operare din familia LINUX.
- **EXT3** (third extended file system) este o perfecționare a lui EXT2 în sensul monitorizării în diverse jurnale a tuturor activităților efectuate; are tendința de a înlocui EXT2 dar încă are erori.
- **FAT12/16/32** (File Allocation Table) este sistemul de fișiere de bază dezvoltat pentru DOS și Windows; este considerat cel mai simplu și, datorită popularității sale, este cel mai folosit pentru discurile flexibile.
- **UMSDOS** este un sistem de fișiere pentru LINUX care simulează cele mai avansate facilități UNIX folosind un sistem de fișiere de tip FAT.
- **HFS** (Hierarchical File System) este folosit de sistemul de operare Apple Mac OS.
- **HPFS** (High Performance File System) a fost creat pentru sistemul de operare OS/2.
- **ISO 9660** (împreună cu extensiile Rock Ridge și Joliet) este un standard care definește sistemul de fișiere pentru CD-ROM; este implementat în Windows, Mac OS, LINUX, UNIX etc.
- **NTFS** (New Technology File System) este sistemul de fișiere standard pentru Windows NT și predecesoarele sale Windows 2000 și Windows XP;
- **UDF** (Universal Disk Format) este folosit pentru înregistrarea informațiilor pe CD-RW și CD-R; se deosebește de ISO 9660 prin înregistrarea de pachete de date cu lungimi diferite.
- **UFS** (Unix File System) a fost creat inițial pentru UNIX; se întâlnește curent în sistemele de operare FreeBSD, NetBSD, OpenBSD și SOLARIS.

Caracteristici ale Sistemelor de Fișiere:

Nume	FAT12	FAT16	FAT32
	12 biți	16 biți	28 biți (4 biți rezervați)
Introdus	1977-1980	1984-1987	1996
Capacitate maximă	16 MB	2 GB	2 TB (1 TB = 2^{40} bytes)
Cluster	0,5 KB – 4KB	2 KB – 32 KB	4 KB -32 KB
Număr maxim clustere	4086	65.526	~268.435.456
Nume fișiere	8.3	8.3	LFN
Sistem operare	DOS	DOS, Windows 3.x	după Windows 95 OSR 2

Nume	NTFS
Introdus	1993
Capacitate maximă	2^{64} B – 1KB
Nume fișiere	255 caractere UTF-16
Sistem operare	Familia Windows NT

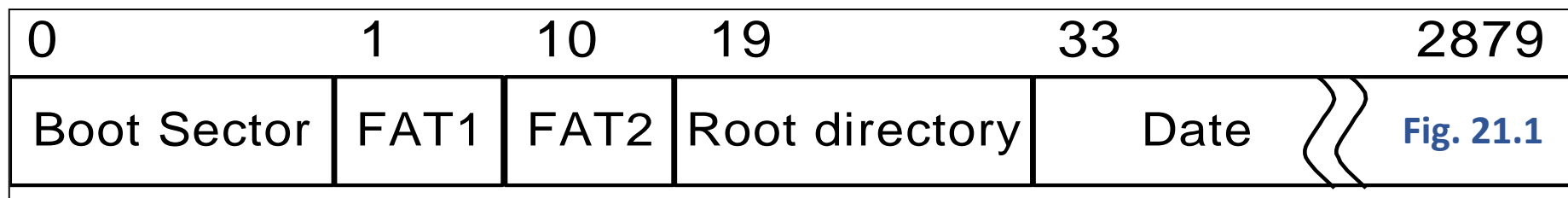
Nume	EXT2
Introdus	1993
Capacitate maximă	16-32 TB
Nume fișiere	255 caractere
Număr maxim fișiere	10^{18}
Dimensiunea maximă a unui fișier	2 TB (kernel<2.6) – 64 TB
Sistem operare	Linux, BSD, Mac OS X

Sistemul de Fișiere File Allocation Table (FAT):

A fost conceput la sfarsitul anilor 70, proiectat initial pentru discuri magnetice externe (floppy disks) si extins apoi spre sistemele de operare de pe calclatoarele personale, fiind utilizat pana in ziua de azi.

Conceptul original al FAT-ului a fost facut pe 8 bits urmatoarele versiuni fiind: FAT12, FAT16, FAT32 si FAT32+.

Structura și deplasamentul în sectoare al fiecărei zone



Sistemul de Fișiere FAT. Tabelul FAT:

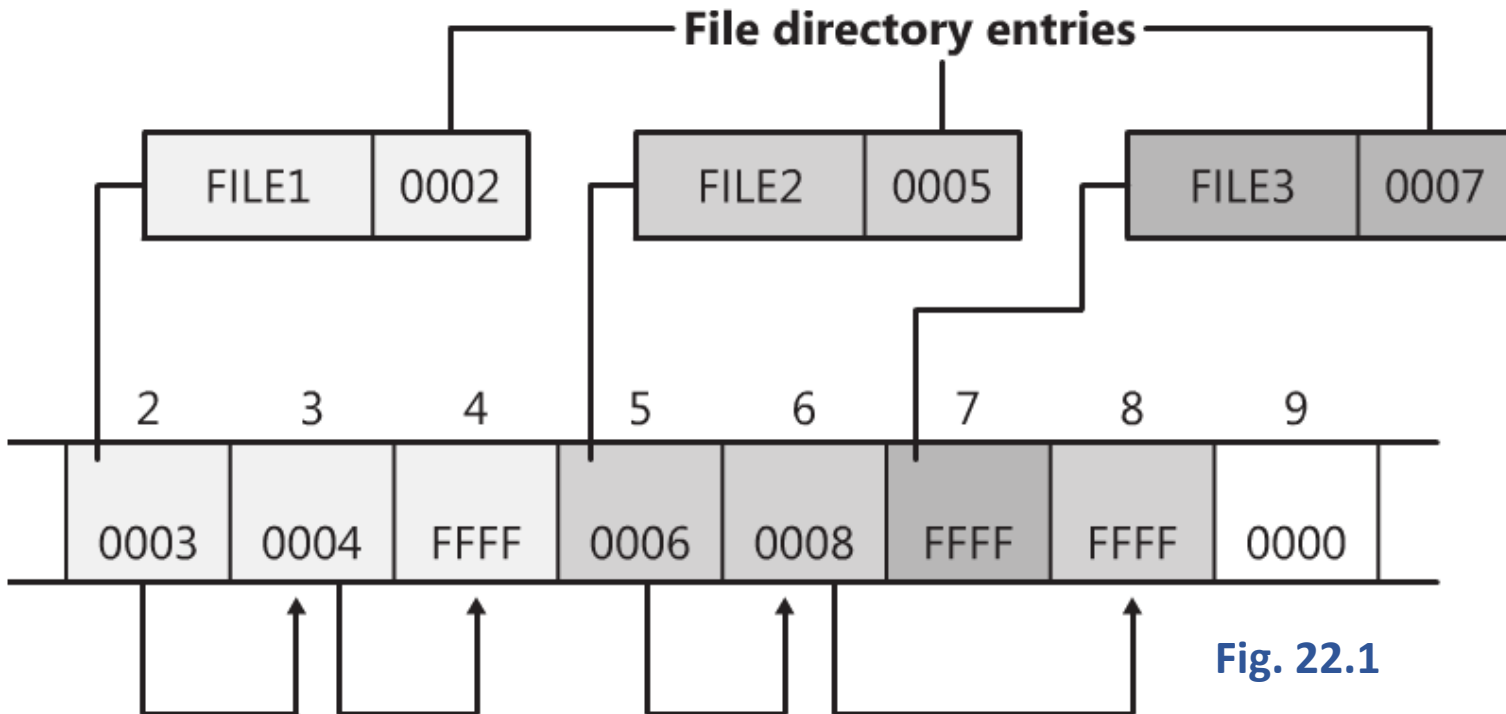


Fig. 22.1

Sistemul de Fișiere FAT. Structura Directorului FAT:

Offset	Dimensiune	Continut
+00h	8	Numele fisierului
+08h	3	Extensia numelui de fisier
+0Bh	1	Atribute
+0Ch	0Ah	Rezervat
+16H	2	Ora ultimei modificari a fisierului
+18h	2	Data ultimei modificari a fisierului
+1Ah	2	Numarul primului cluster ocupat de fisier
+1Ch	4	Dimensiunea fisierului (in octeti)

Sistemul de Fișiere FAT. Parametri limită:

Dimensiunea maxima a fisierelor

4,294,967,295 bytes (4 GiB - 1) with FAT16B and FAT32
274,877,906,943 bytes (256 GiB - 1) only with FAT32+

Numarul maxim de fisiere

FAT12: 4,068 for 8 KiB clusters
FAT16: 65,460 for 32 KiB clusters
FAT32: 268,173,300 for 32 KiB clusters

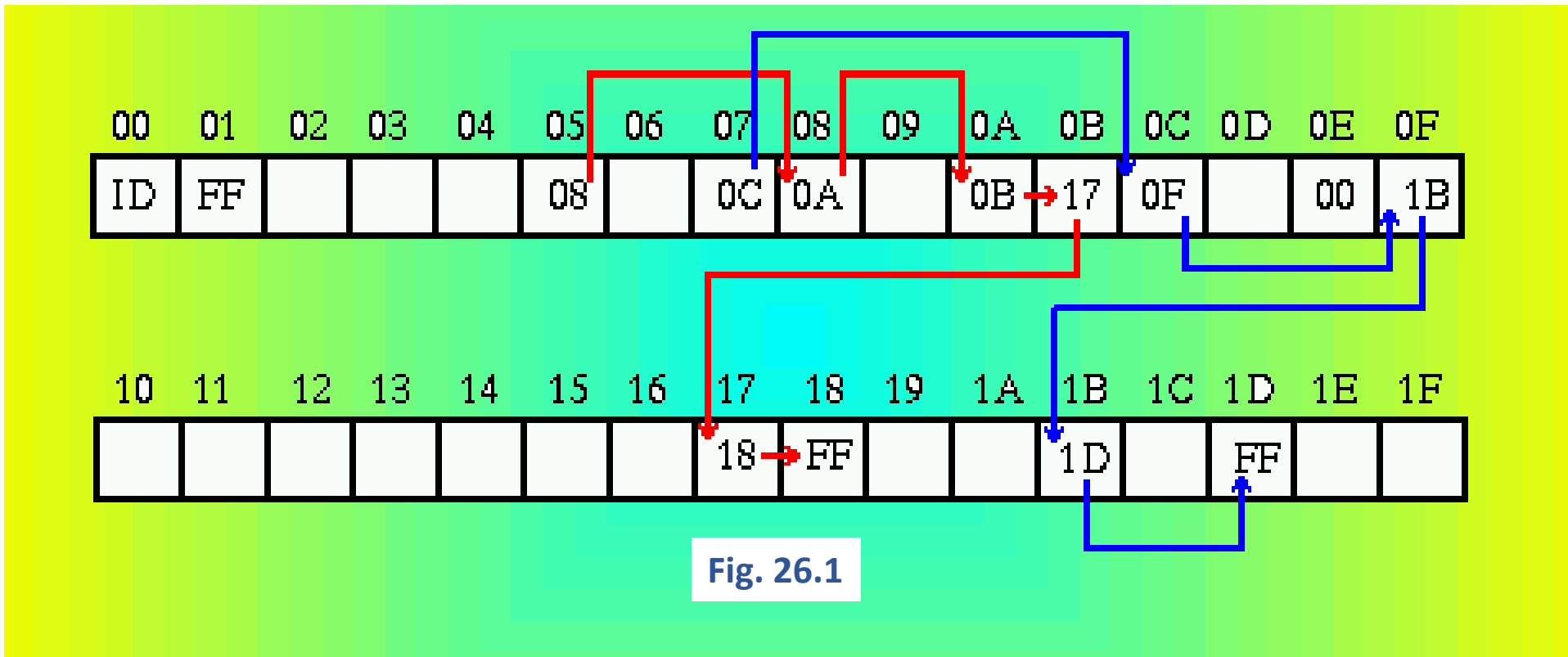
Numele fisierelor

8.3 filename, sau 255 caractere UCS-2 (Universal Character Set) cu sistemul LFN (long filenames)

Sistemul de Fișiere FAT. Structura sectorului BOOT Record:

Offset	Dimensiune (octeti)	Continut
+00h	3	JMP <i>adresa</i> . Salt la rutina de incarcare a sistemului de operare
+03h	8	Numele producatorului si versiunii
+0Bh	2	Numarul de octeti pe sector
+0Dh	1	Numarul de sectoare pe cluster
+0Eh	2	Numarul de sectoare rezervate (inaintea FAT)
+10h	1	Numarul de FAT-uri
+11h	2	Numarul maxim de intrari in directorul radacina
+13h	2	Numarul total de sectoare
+15h	1	Media descriptor
+16h	2	Numarul de sectoare dintr-un FAT
+18h	2	Numarul de sectoare pe pista
+1Ah	2	Numarul de capete de citire/scriere
+1Bh	2	Numarul de sectoare ascunse
+1Dh	...	Codul de bootare

Sistemul de Fișiere FAT. Exemplu de amplasare și adresare a cluster-elor în SF FAT:



Sistemul de Fișiere FAT16/32. Structura sectorului MBR:

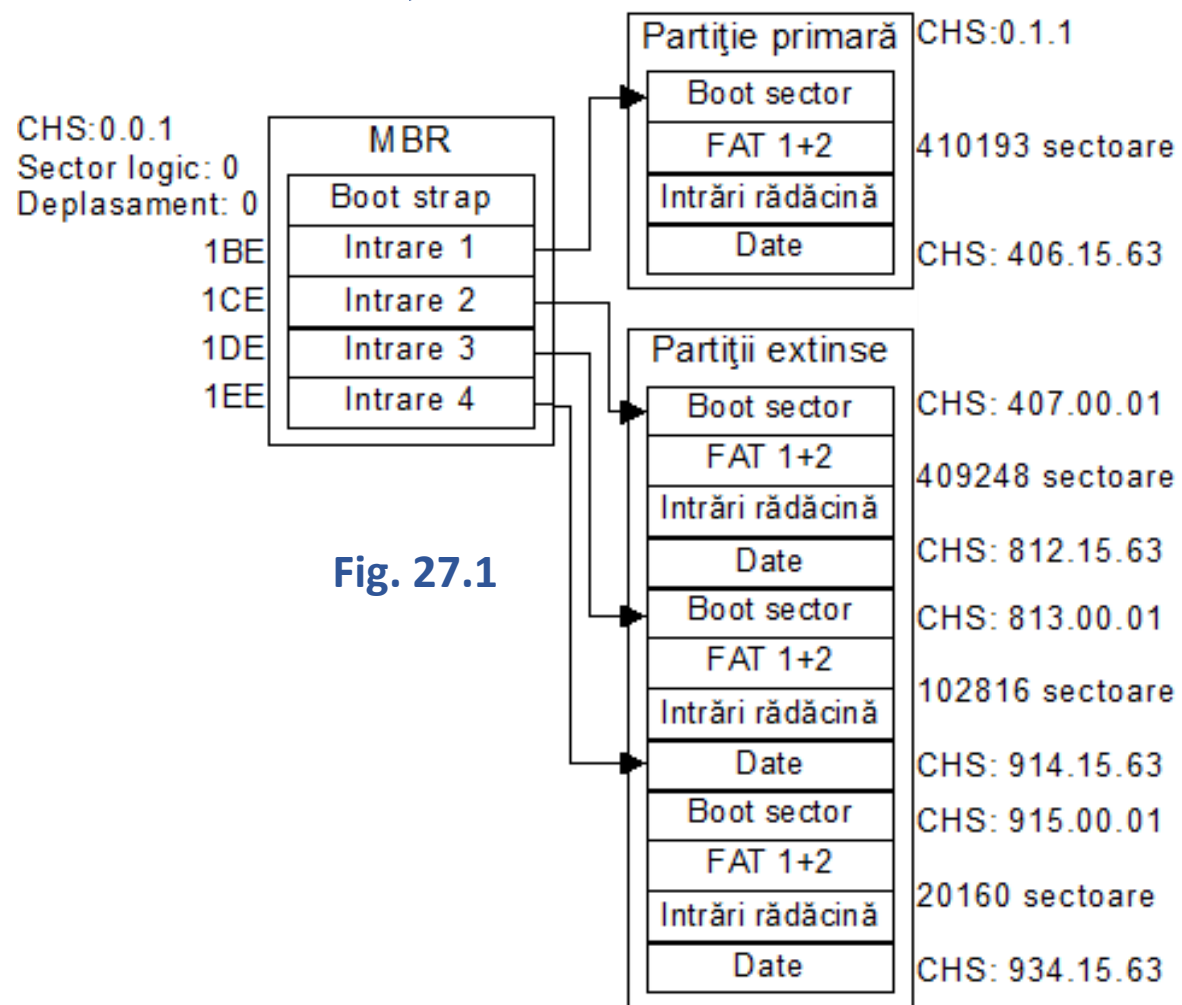


Fig. 27.1

Sistemul de Fișiere FAT16/32. Structura sectorului MBR:

Conținutul sectorului boot pentru FAT HDD

Deplasament	Dimensiune	Conținut	Semnificație
0000h	3	EB 34 90	Salt absolut la bootstrap
0003h	8	MSWIN4.1	Sistemul de operare cu care s-a formatat
000Bh	25	0200 ...	Bloc de parametri BIOS (BPB)
0024h	26	01 00 ...	Bloc de parametri BIOS extins (EPB)
003Eh	338	FA 33 ...	Bootstrap (program cod mașină)
01FEh	2	55AAh	Semnătură sector executabil

Sistemul de Fișiere FAT16/32. Structura sectorului MBR:

Structura sectorului de partiție		
Deplasament	Dimensiune	Conținut și descriere
000h	446	Program executabil în cod mașină
1BEh	16	Prima intrare în tabela de partiționare
1CEh	16	A doua intrare în tabela de partiționare
1DEh	16	A treia intrare în tabela de partiționare
1EEh	16	A patra intrare în tabela de partiționare
1FE h	2	Semnătură de identificare (55AAh)

Sistemul de Fișiere FAT16/32. Structura sectorului MBR:

Structura intrării în tabela de partiție		
Offset	Octeți	Conținut și descriere
00h	1	Starea partiției: 00 – partiție extinsă; 80 – partiție principală.
01h	1	Numărul suprafeței HDD unde începe partiția
02h	2	Sectorul și cilindrul unde începe partiția
04h	1	Tipul partiției.
05h	1	Numărul suprafeței HDD unde se termină partiția
06h	2	Sectorul și cilindrul unde se termină partiția
08h	4	Deplasamentul (în sectoare) față de sectorul de boot
0Ch	4	Numărul de sectoare ale partiției

Sistemul de Fișiere FAT16/32. Structura sectorului MBR. Exemplu de completare:

Partiția	Stare	C.H.S start	Tip	C.H.S. sfârșit	Adresă	Nr.sectoare
Partiția 1	principală	00.01.01	06	406.15.63	63	410193
Partiția 2	extinsă	407.00.01	0F	812.15.63	410256	409248
Partiția 3	extinsă	813.00.01	05	914.15.63	819504	102816
Partiția 4	extinsă	915.00.01	01	934.15.63	922320	20160

Sistemul de Fișiere NTFS:



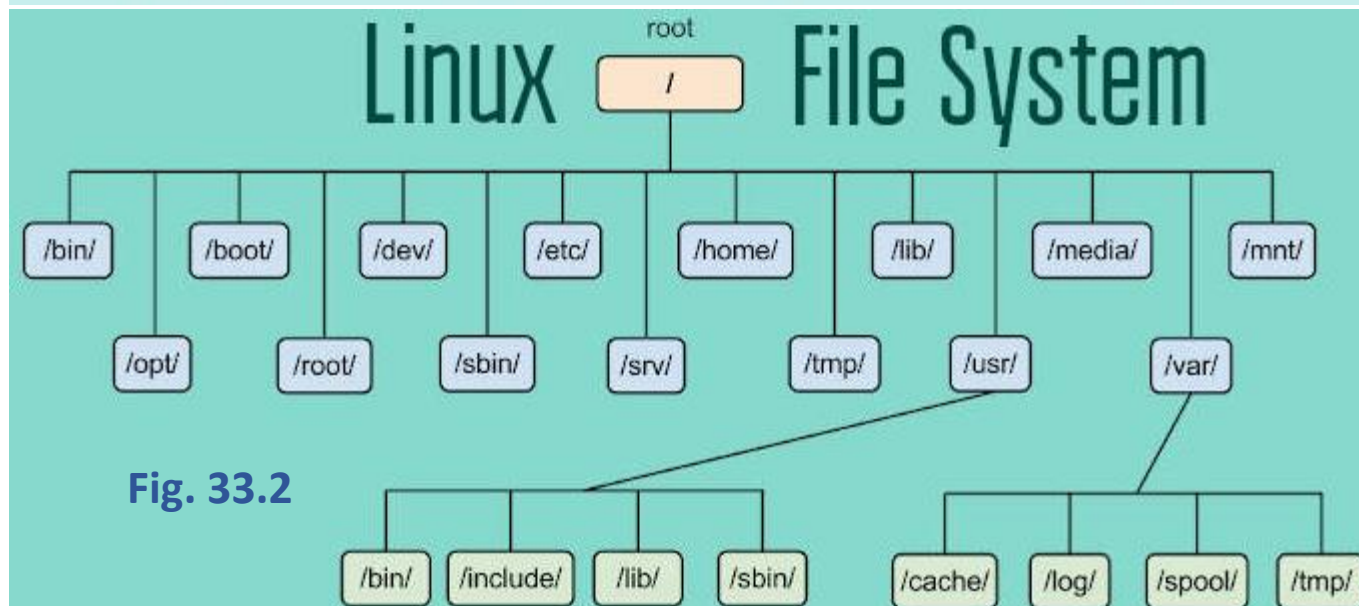
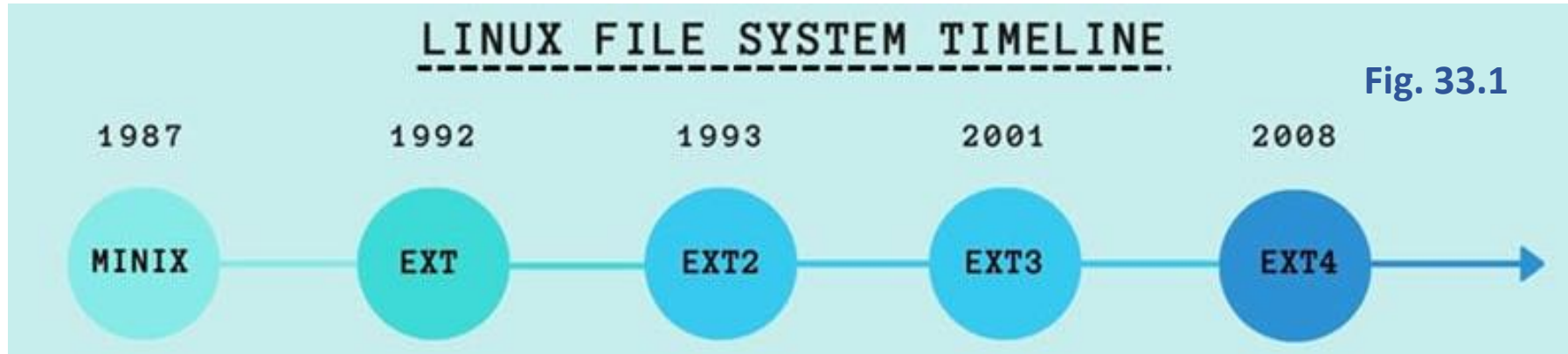
Fig. 32.1

NTFS File System

NTFS – este un sistem de fișiere modern. Caracteristici noi, față de sisteme de fișiere:

- **Viteză îmbunătățită:** NTFS este un sistem de fișiere rapid și comparat;
- **Comprimarea:** Permite să comprime direct fișierele pe disc, ceea ce reduce consumul de spațiu de stocare;
- **Criptarea:** NTFS permite criptarea fișierelor cu certificatele electronice;
- **Verificarea de sectoare defecte:** Permite sistemului de fișiere NTFS să detecteze și să blocheze sectoarele defecte;
- **Poate fi recunoscut după tipul de partiție "07", care este folosit și de HPFS.**

Sistemul de Fișiere EXT, EXT2/3/4:



Întreruperea BIOS INT 13h pentru managementul datelor:

INT 13h / AH = 00h - reset disk system.

INT 13h / AH = 02h - read disk sectors into memory.

INT 13h / AH = 03h - write disk sectors.

input:

AL = number of sectors to read/write (must be nonzero)

CH = cylinder number (0..79).

CL = sector number (1..18).

DH = head number (0..1).

DL = drive number (0..3 , for the emulator it depends on quantity of FLOPPY_ files).

ES:BX points to data buffer.

return:

CF set on error.

CF clear if successful.

AH = status (0 - if successful).

AL = number of sectors transferred.

Note: each sector has **512** bytes.

Întreruperea DOS INT 21h pentru managementul datelor:

INT 21h / AH= 39h - make directory.

entry: **DS:DX** -> ASCIZ pathname; zero terminated string.

Example:

```
org 100h
```

```
mov dx, offset filepath
```

```
mov ah, 39h
```

```
int 21h
```

```
ret
```

```
filepath DB "C:\mydir", 0 ; path to be created.
```

```
end
```

INT 21h / AH= 3Ah - remove directory.

Entry: **DS:DX** -> ASCIZ pathname of directory to be removed.

Întreruperea DOS INT 21h pentru managementul datelor:

INT 21h / AH= 3Bh - set current directory.

Entry: **DS:DX** -> ASCIZ pathname to become current directory (max 64 bytes).

INT 21h / AH= 3Ch - create or truncate file.

Entry: **CX** = file attributes:

`mov cx, 0 ; normal - no attributes.`

`mov cx, 1 ; read-only.`

`mov cx, 2 ; hidden.`

`mov cx, 4 ; system`

`mov cx, 7 ; hidden, system and read-only!`

`mov cx, 16 ; archive`

DS:DX -> ASCIZ filename.

Example:

```
org 100h
mov ah, 3ch
mov cx, 0
mov dx, offset filename
mov ah, 3ch
int 21h
jc err
mov handle, ax
jmp k
filename db "myfile.txt", 0
handle dw ?
err: ; ....
k: ret
```

Întreruperea DOS INT 21h pentru managementul datelor:

INT 21h / AH= 3Dh - open existing file.

Entry: **AL** = access and sharing modes:

`mov al, 0 ; read`

`mov al, 1 ; write`

`mov al, 2 ; read/write`

DS:DX -> ASCIZ filename.

Example:

`org 100h`

`mov al, 2`

`mov dx, offset filename`

`mov ah, 3dh`

`int 21h`

`jc err`

`mov handle, ax`

`jmp k`

`filename db "myfile.txt", 0`

`handle dw ?`

`err: ;`

`k: ret`

Întreruperea DOS INT 21h pentru managementul datelor:

INT 21h / AH= 3Eh - close file.

Entry: **BX** = file handle

INT 21h / AH= 3Fh - read from file.

Entry:

BX = file handle.

CX = number of bytes to read.

DS:DX -> buffer for data.

INT 21h / AH= 40h - write to file.

Entry:

BX = file handle.

CX = number of bytes to write.

DS:DX -> data to write.

INT 21h / AH= 41h - delete file (unlink).

Entry:

DS:DX -> ASCIZ filename (no wildcards, but see notes).

Înteruperea DOS INT 21h pentru managementul datelor:

INT 21h / AH= 42h - SEEK - set current file position.

Entry:

AL = origin of move: **0** - start of file. **1** - current file position. **2** - end of file.

BX = file handle.

CX:DX = offset from origin of new file position.

Example:

org 100h

mov ah, 3ch

mov cx, 0

mov dx, offset filename

mov ah, 3ch

int 21h ; create file...

mov handle, ax

mov bx, handle

mov dx, offset data

mov cx, data_size

mov ah, 40h

int 21h ; write to file...

```
mov al, 0
mov bx, handle
mov cx, 0
mov dx, 7
mov ah, 42h
int 21h ; seek...
mov bx, handle
mov dx, offset buffer
mov cx, 4
mov ah, 3fh
int 21h ; read from file...
mov bx, handle
mov ah, 3eh
int 21h ; close file...
Ret
filename db "myfile.txt", 0
handle dw ?
data db "hello files! "
data_size=$-offset data
buffer db 4 dup(' ')
```

Înteruperea DOS INT 21h pentru managementul datelor:

INT 21h / AH= 47h - get current directory.

Entry:

DL = drive number (00h = default, 01h = A:, etc)

DS:SI -> 64-byte buffer for ASCIZ pathname.

INT 21h / AH= 56h - rename file / move file.

Entry:

DS:DX -> ASCIZ filename of existing file.

ES:DI -> ASCIZ new filename.

FDD. Resursele rezervate de sistem pentru gestiunea FDD:

- Întreruperi Hardware: IRQ6;
- Întreruperi Software: INT 13h și INT 21h;
- Canale DMA: DMA2;
- Porturi I/O: 03F0h – 03F7h.

5.4. FDD. Programarea operațiilor de intrare/ieșire:

FDD Exemple: 8", 5,25" și 3,5".

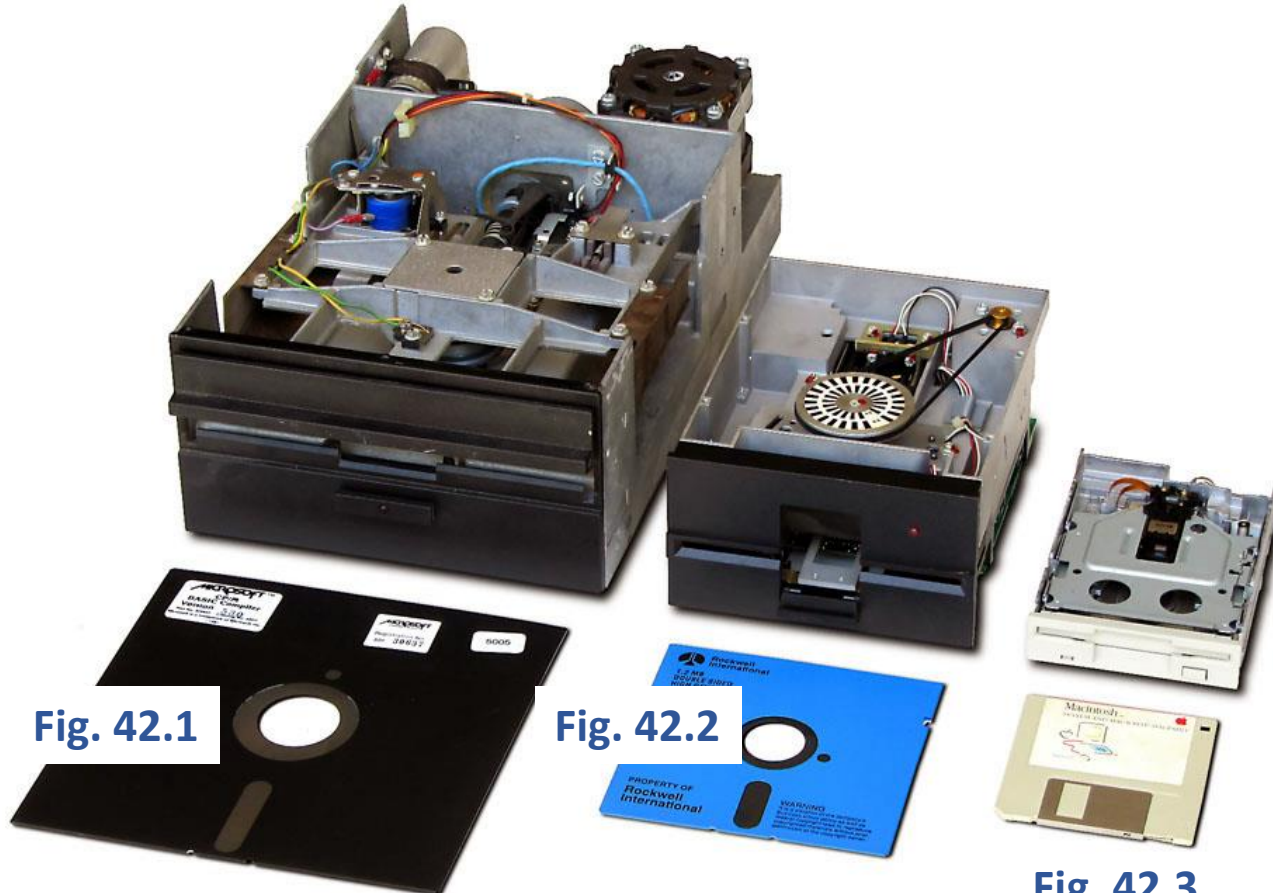


Fig. 42.1

Fig. 42.2

Fig. 42.3

	5 ¼ INCH		3 ½ INCH		
Parametrii	DD	HD	DD	HD	ED
Nr. piste pe inch	48	96	135	135	135
Biți pe inch	5,876	9,646	8,717	17,434	34,896
Material magnetic	Ferită	Cobalt	Cobalt	Cobalt	Bariu
Intensitatea CM (A/m)	2400	4800	4800	5760	6000
Grosime (μ inch)	100	50	70	40	100
Magnetizare	Oriz.	Oriz.	Oriz.	Oriz.	Vert.

FDD. Caracteristici 5,25” și 3,5”:

Dimensiunea dischetei (inch)	3 ½ “	3 ½ “	3 ½ “	5 ¼ “	5 ¼ “
Capacitatea dischetei (kB)	2880	1440	720	1200	360
Număr capete	2	2	2	2	2
Număr piste pe fiecare față	80	80	80	80	40
Număr de sectoare pe pistă	36	18	9	15	9
Număr de biți pe sector	512	512	512	512	512
Număr sectoare pe cluster	2	1	2	1	2
Număr sectoare pentru FAT	9	9	3	7	2
Sectoare în director rădăcină	15	14	7	14	7
Număr sectoare pe disc	5760	2880	1440	2400	720

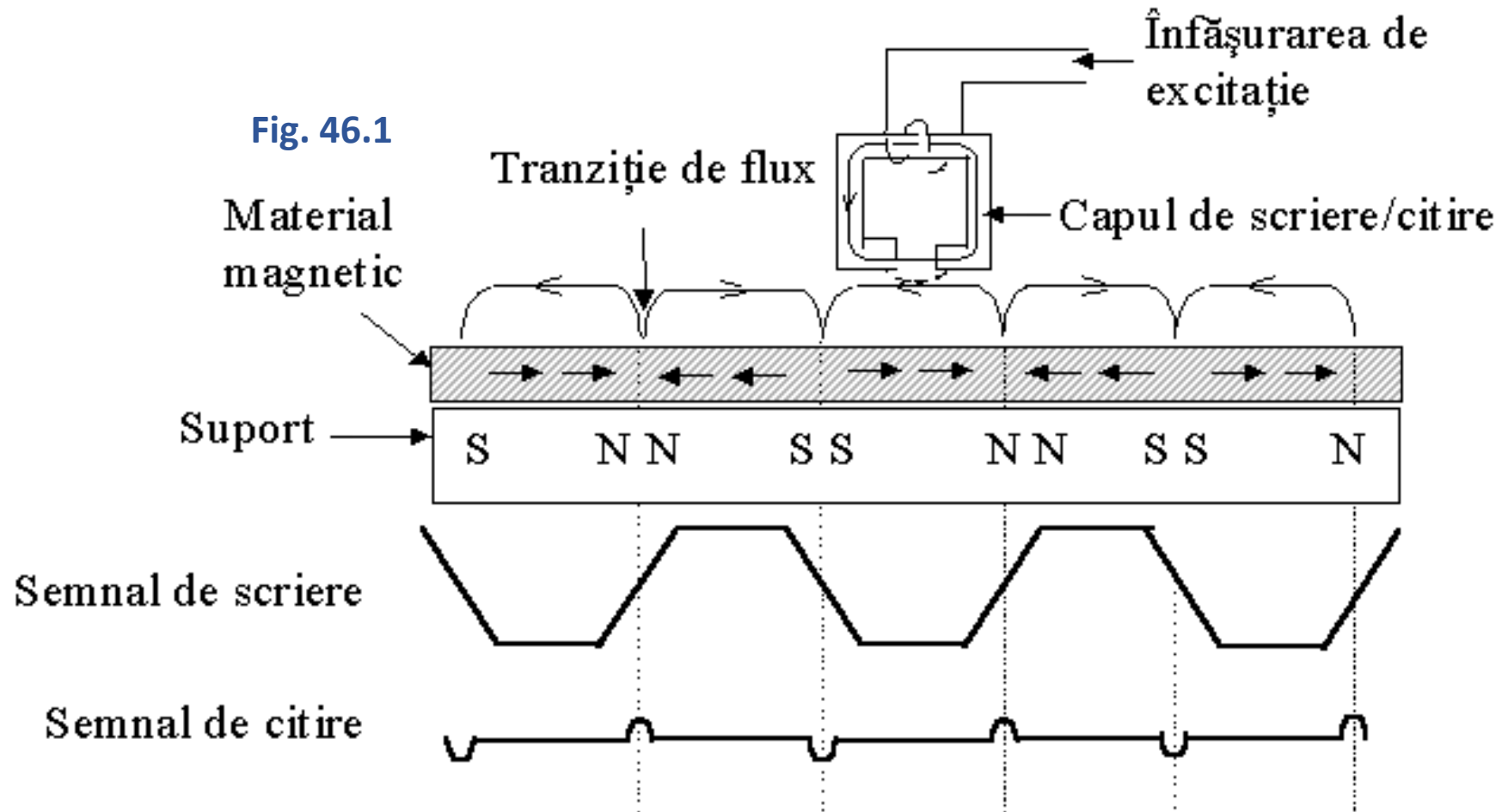
HDD. Resursele rezervate de sistem pentru gestiunea HDD:

- Întreruperi Hardware: IRQ14, IRQ15;
- Întreruperi Software: BIOS - INT 13h și DOS - INT 21h;
- Canale DMA: DMA7;
- Porturi I/O: 0168h – 016Fh, 0170h-0177h, 01E8h-01EFh, 01F0h-01F7h....

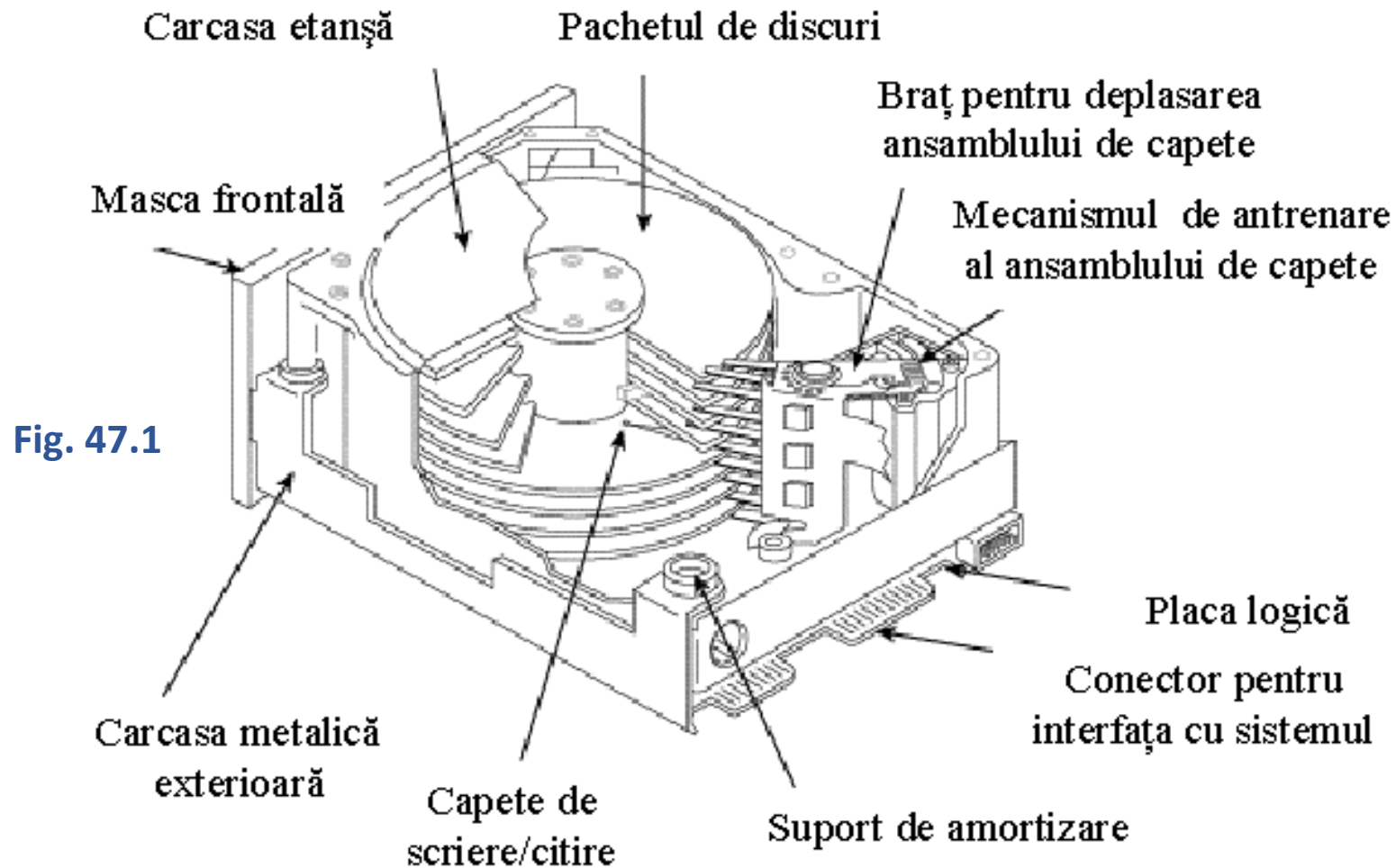
HDD. Parametri și caracteristici:

- Capacitatea;
- Densitatea de suprafață;
- Parametrii de poziționare;
- Rata externă de transfer;
- Memoria cache;
- Viteza de rotație;

HDD. Tehnologia de înregistrare a datelor:



HDD. Elementele constructive ale unui HDD:



HDD. Discurile magnetice în formă de platane. HDD folosesc 2, 3, ..., 12 platane:

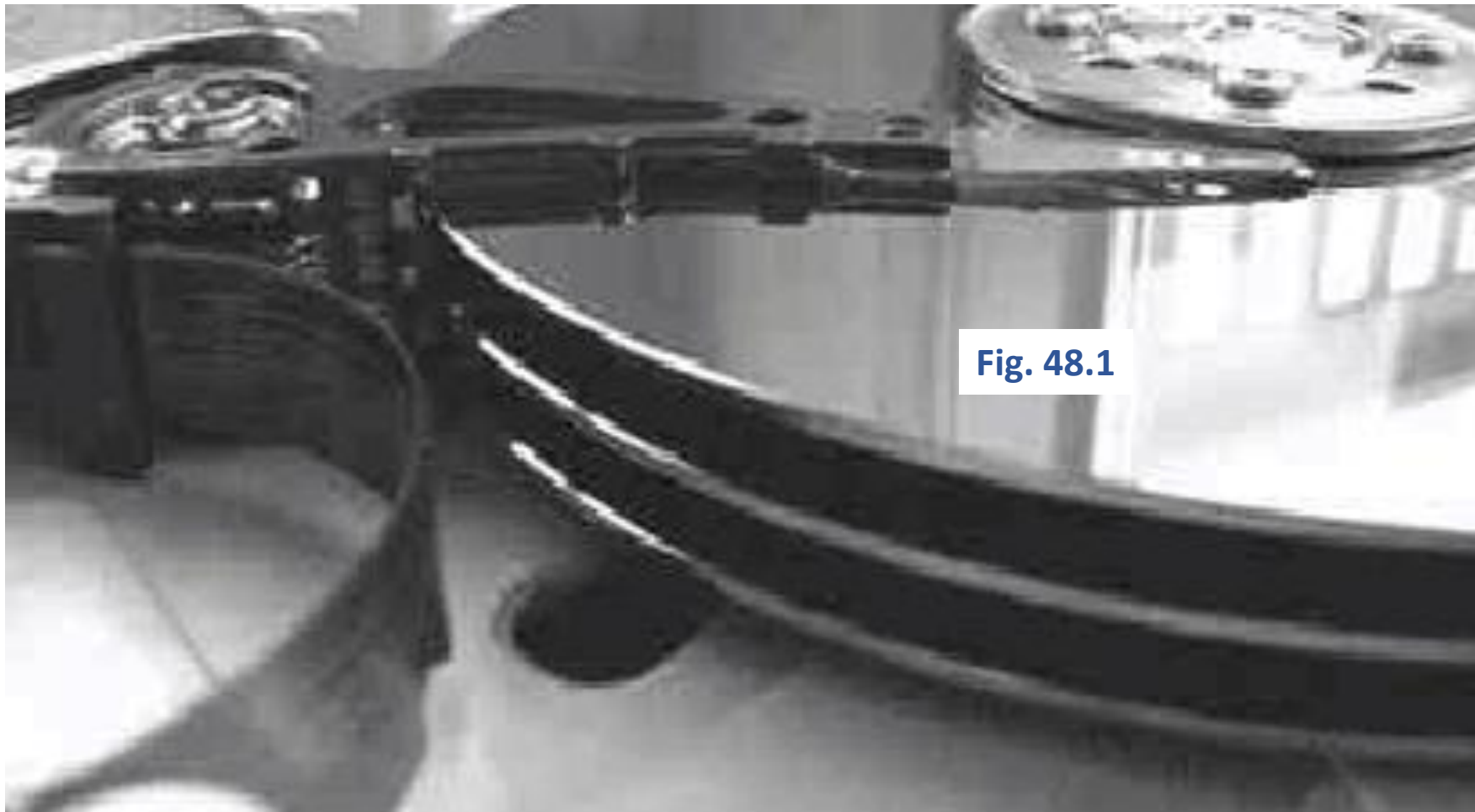
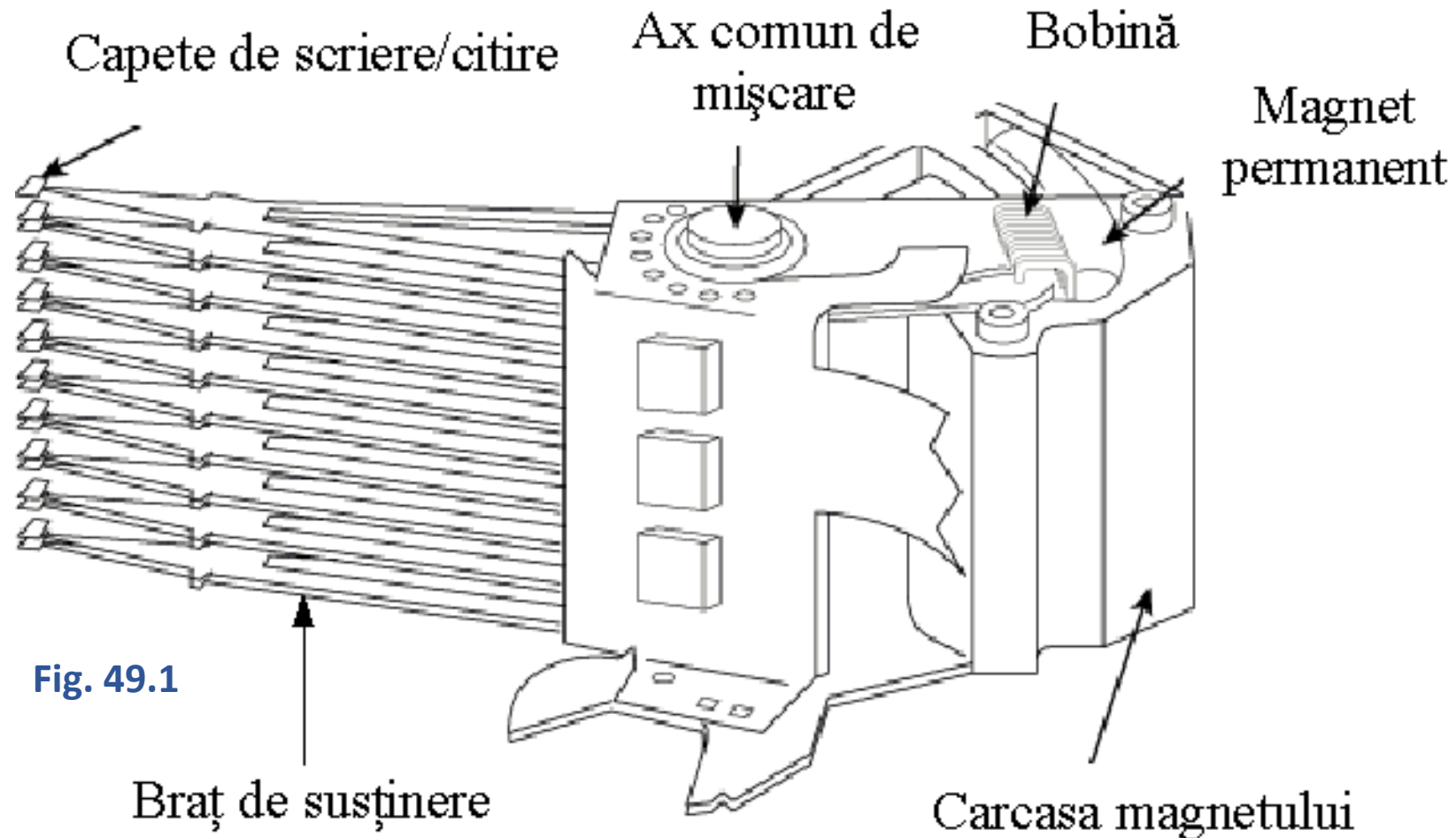


Fig. 48.1

HDD. Blocul de capete de scriere/citire:



HDD. Organizarea datelor. Servomecanismul cu informații incluse:

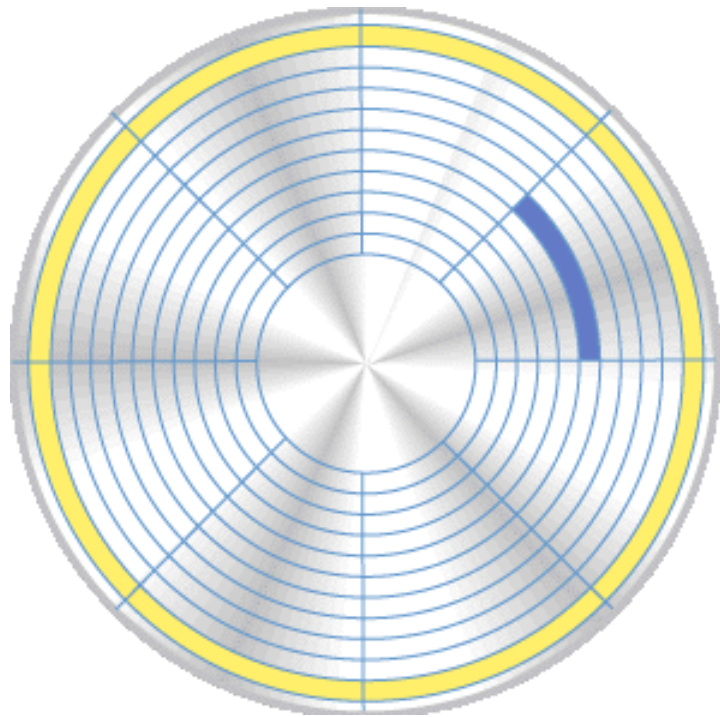


Fig. 50.1

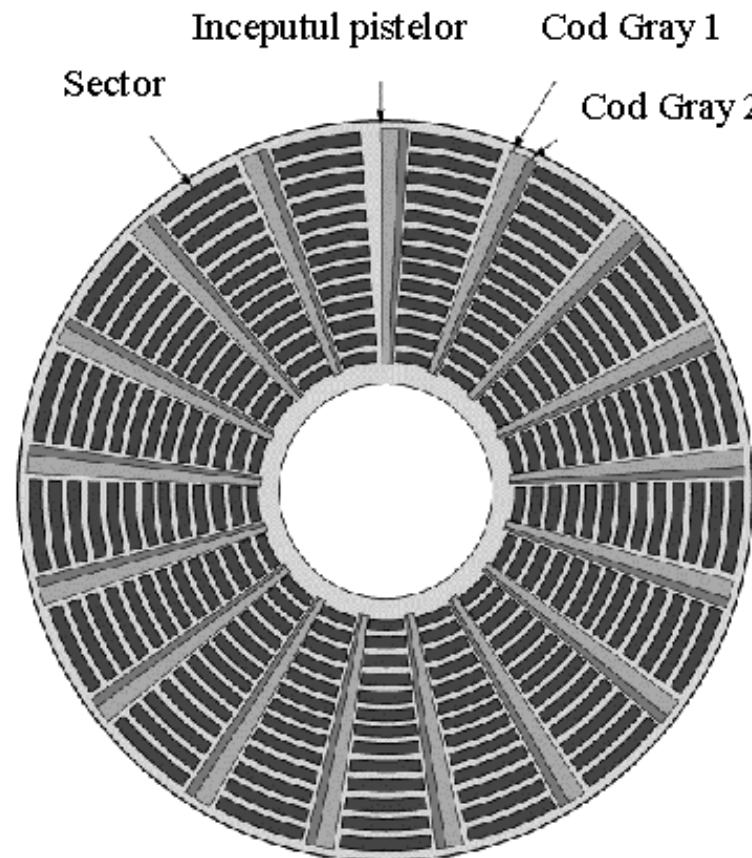


Fig. 50.2

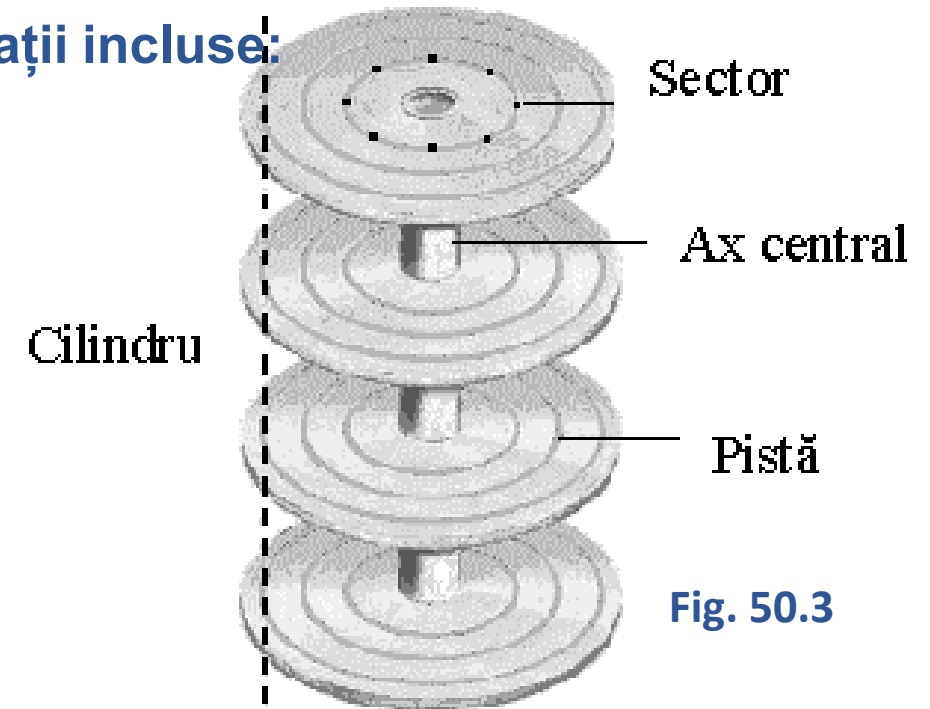


Fig. 50.3

HDD. Organizarea datelor. Prezentare 3D:

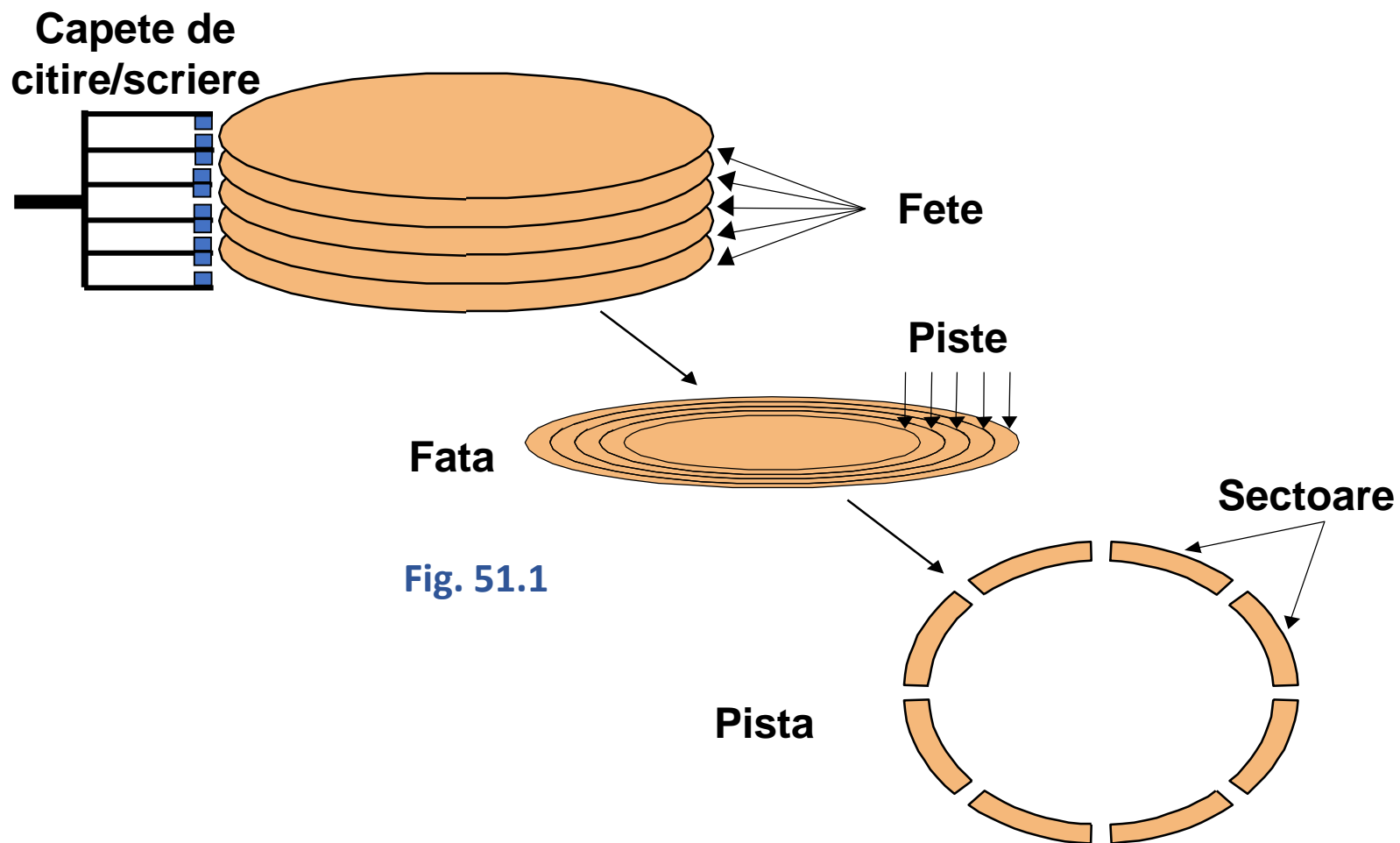


Fig. 51.1

HDD. Zona logică de identificare a începutului pistei:

Nr. octeți	Valoare	Descriere
80	4EH	Buffer la început de pistă
13	00H	Octeți pentru sincronizarea ceasului
1	A1H	Informează controlerul că urmează date
1	FE	Informează că urmează zona de identificare a pistei
1	**	Numărul pistei (HEXA)
1	**	Numărul feței (HEXA)
1	**	Numărul de sectoare pe pistă (HEXA)
1	**	Lungimea fiecărui sector
2	**	Octeți pentru codul CRC detector de eroare
16	4EH	Spațiu de separație între octeții de identificare a pistei și sectoare de date

HDD. Zona logică de identificare a sectorului:

Nr. octeți	Valoare	Descriere
13	00H	Octeți pentru sincronizarea ceasului
1	A1H	Informează controlerul că urmează date
1	FE	Informează că urmează un sector normal de date
	FB	Informează că urmează un sector special de date
2	**	Numărul cilindrului (HEXA)
1	**	Numărul capului de scriere/citire (HEXA)
1	**	Numărul sectorului (HEXA)
2	**	Octeți pentru codul CRC detectori de eroare
3	00H	Spațiu de separație între zona de identificare și zona de date
13	00H	Octeți pentru sincronizarea ceasului
1	A1H	Informează controlerul că urmează date
1	F8H	Informează că urmează date utilizator
512	**	Date
2	**	Octeți pentru codul CRC detector de eroare
3	00H	Spațiu de separație a datelor
15	00H	Buffer pentru variația vitezei de rotație a discului

HDD. Metode de codificare a datelor:

Date	Cod FM
0	TN
1	TT

Codarea FM este prima apărută și cea mai puțin eficientă din punct de vedere al optimizării spațiului ocupat.

Data precedentă	Data curentă	Cod MFM
*	1	NT
0	0	TN
1	0	NN

Codificarea MFM este prima codare ce diminuează numărul de tranziții necesare.

HDD. Metode de codificare a datelor:

Grup de biți	Codificare RLL2,7
10	NTNN
11	TNNN
000	NNNTNN
010	TNNTNN
011	NNTNNN
0010	NNTNNTNN
0011	NNNNTNNN

Codarea RLL asigură o reducere și mai mare a spațiului de înregistrare necesar pentru inscrierea unui bloc de date printr-o reducere suplimentară a numărului de tranziții necesare.

HDD. Exemplu de codificare X(ASCII) = 01011000b:

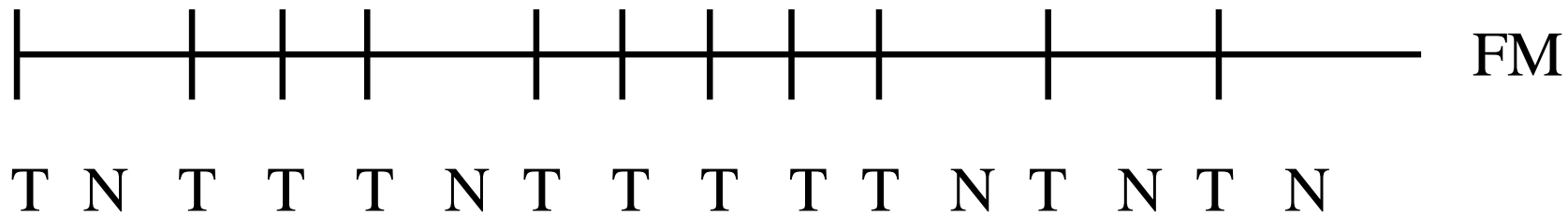


Fig. 56.1

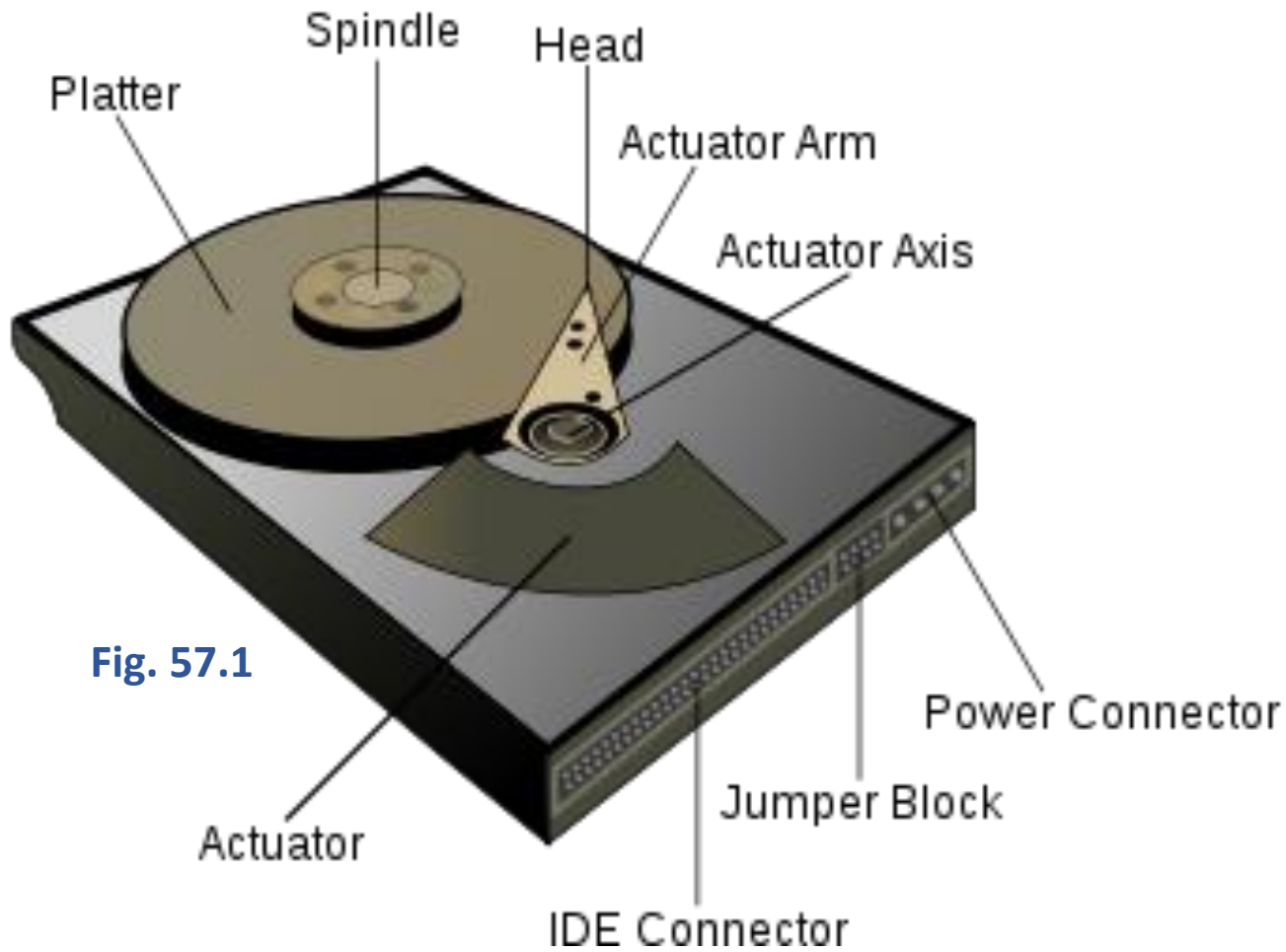


Fig. 56.2



Fig. 56.3

HDD. Exemple de dispozitive HDD:



HDD. Exemple de dispozitive SSD:

SSD vs HDD

(Solid State Drive)

(Hard Disk Drive)



Fig. 58.1

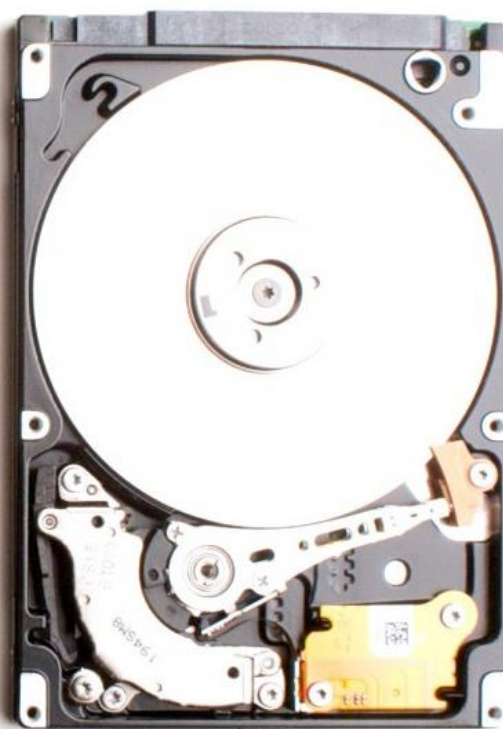


Fig. 58.2

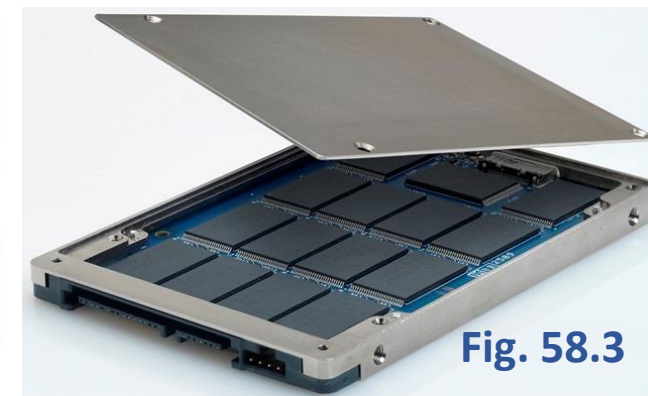


Fig. 58.3



Fig. 58.4

HDD. Interfața IDE/ATA/PATA:

Această interfață este destinată pentru conectarea unităților de hard-disc la unitatea centrală. Atât denumirea IDE cât și ATA sunt folosite, însă IDE (Integrated Drive Electronics) este mai populară. De-a lungul timpului au apărut o serie mare de variații ale acestei denumiri, fiecare dintre ele aducând o îmbunătățire. Astfel au apărut: ATA/ATAPI, EIDE, ATA-2, Fast ATA, ATA-3, Ultra ATA, Ultra DMA, etc. Practic, această interfață definește legătura dintre unitatea centrală și o unitate de hard-disc cu controler incorporat.

Standardul curent IDE este o interfață paralelă ce utilizează un cablu cu 40 de fire.

HDD. Interfața SATA. Interacțiunea Sistem - SATA:

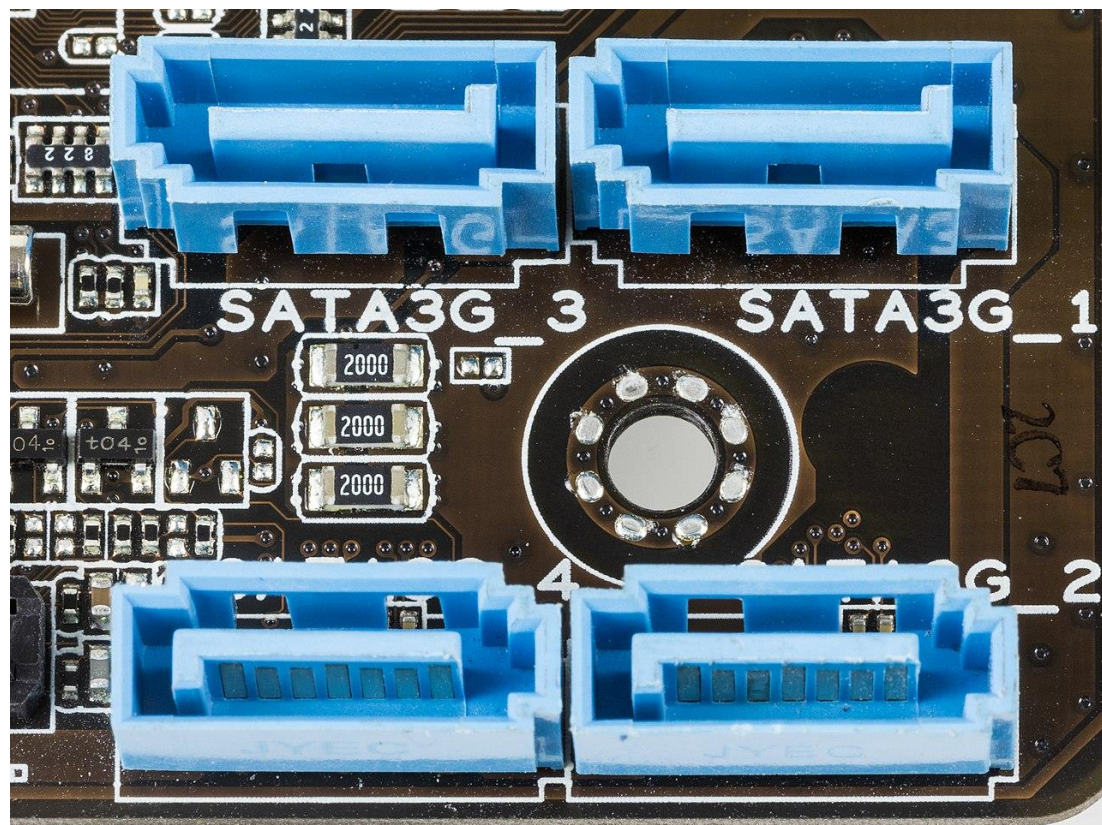


Fig. 60.1

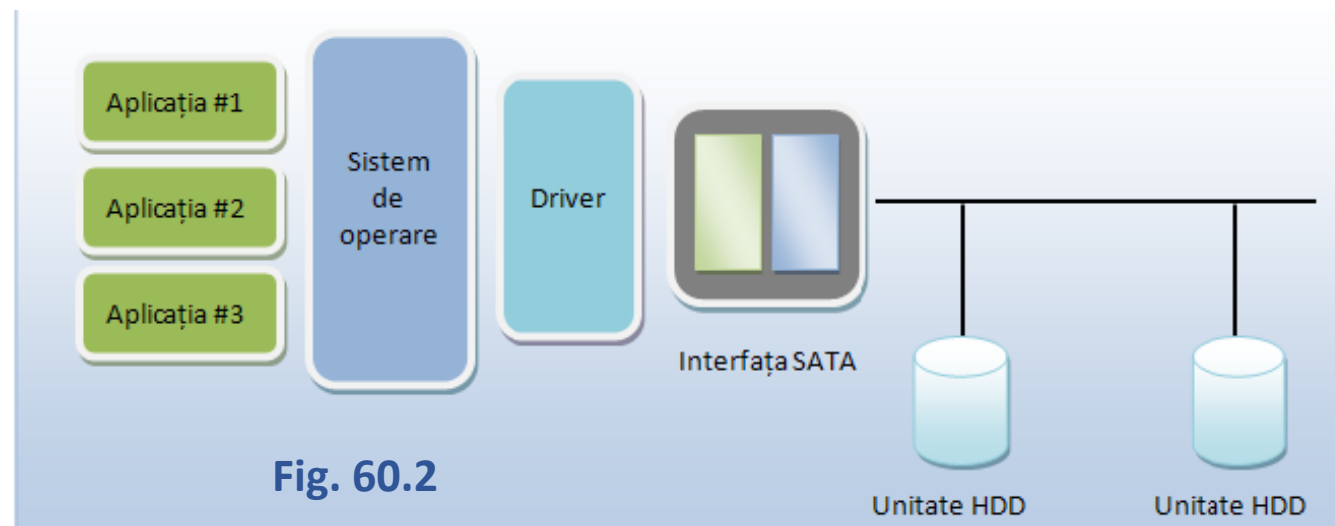


Fig. 60.2

HDD. Standarde de Interfațe SATA:

Versiune	Denumire comercială	Data lansării	Rata teoretică	Rata efectivă
SATA 1.0	Serial ATA-150	ianuarie 2003	1,5 Gb/s (192 MB/s)	1,2 Gb/s (150 MB/s)
SATA 2.0	Serial ATA-300	aprilie 2004	3 Gb/s (384 MB/s)	2,4 Gb/s (300 MB/s)
SATA 3.0	Serial ATA-600	mai 2009	6 Gb/s (768 MB/s)	4,8 Gb/s (600 MB/s)
SATA 3.1		iulie 2011		
SATA 3.2		august 2013	16 Gb/s	1969 MB/s
SATA 3.3		februarie 2016		
SATA 3.4		iunie 2018		
SATA 3.5		iulie 2020		

CD ROM. Clasificarea Discurilor Optice:

A) - Discuri R/O (Read/Only) sunt acele discuri care pot fi doar citite de către utilizator. Înscrierea lor cu informația dorită se face de către producător. Din această categorie fac parte:

- discurile video;
- compact discurile (CD-ROM fiind principal exponent);
- discurile DVD (Digital Versatil Disc sau Digital Video Disc).

B) – Discuri R/W (Read/Write) sunt acele discuri care pe lângă faptul că pot fi citite, pot fi și înscrise (o dată sau de mai multe ori) de către utilizator:

- discuri înscriptibile (pot fi înscrise o singură dată);
- discuri reînscriptibile (pot fi înscrise de mai multe ori);

Compact discurile cuprind CD-ROM-urile, compact discurile audio, compact discurile grafice, compact discurile multimedia, CD TEXT și CD EXTRA. DVD-urile sunt și ele de 3 categorii: audio, video și de date (numite și DVD-ROM).

Discurile înscriptibile se împart în: CD-R (CD-Recordable) și DVD-R (DVD-Recordable). Discurile reînscriptibile se prezintă într-o diversitate mai mare: compact discuri magneto-optice (CD-MO), compact discuri Re-Write (CD-RW), alături de seria de DVD-uri - DVD-RW, DVD-RAM și discuri +RW.

CD ROM. Standarde pentru Discurile Optice:

Red Book – descrie proprietățile fizice ale discului compact audio (CD-DA) și metoda de codificare a informațiilor audio înregistrate. La înregistrare se folosește o metodă digitală de codare numită PCM (Pulse Code Modulation).

Yellow Book – prezintă specificațiile pentru discul compact folosit la înregistrarea datelor incluzând caracteristicile discului, parametrii ansamblului optic de citire, codificarea datelor și corecția erorilor, structura datelor pe discul compact.

Green Book – prezintă compact discul interactiv (CD-I) și sistemul de operare al acestei unități, incluzând formatul discului, structura de fișiere, codificarea semnalului audio prin metoda ADPCM, codificarea și decodificarea informațiilor video.

Orange Book – definește atât compact discurile înscriptibile CD-R și cele reînscriptibile CD-RW cât și cele realizate în tehnologia magne-to-optică (CD-MO), inclusiv posibilitatea de înregistrare multi-sesiune;

White Book – definește specificațiile pentru discul video incluzând formatul discului, structura de fișiere, codificarea MPEG, etc.

Blue Book – definește specificațiile pentru discul audio îmbunătățit, cunoscut și sub denumirea de CD EXTRA (acesta este un disc multi-sesiune conținând sesiuni de date și audio, inaccesibil pentru înregistrare la utilizator);

CD ROM. Citirea datelor de pe Discuri Optice:

1. Rotirea discului;
2. Poziționarea capului de citire pe pista spiralată;
3. Focalizarea fascicolului laser;
4. Recepția fluxului de unde reflectate;
5. Reglarea vitezei de rotație a discului optic;
6. Identificarea datelor citite de pe discul optic.

CD ROM. Structura pistei. Citirea datelor de pe Discuri Optice:

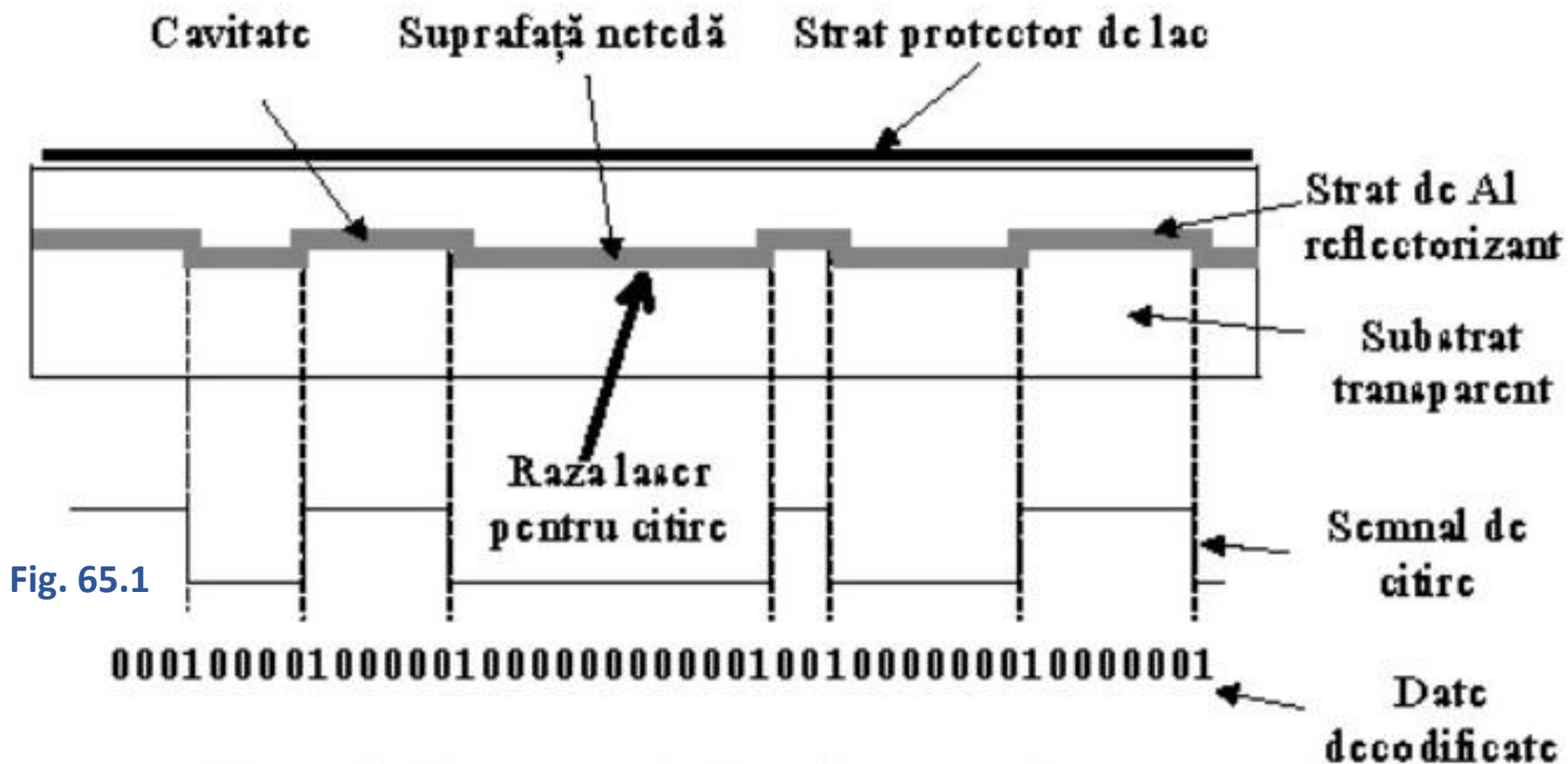


Fig. 65.1

CD ROM. Structura pistei. Citirea datelor de pe Discuri Optice:

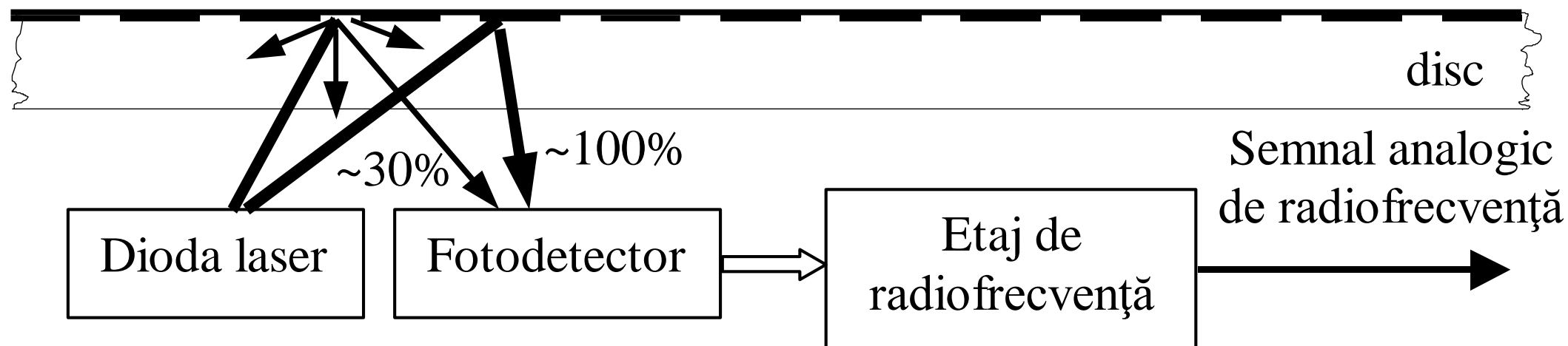
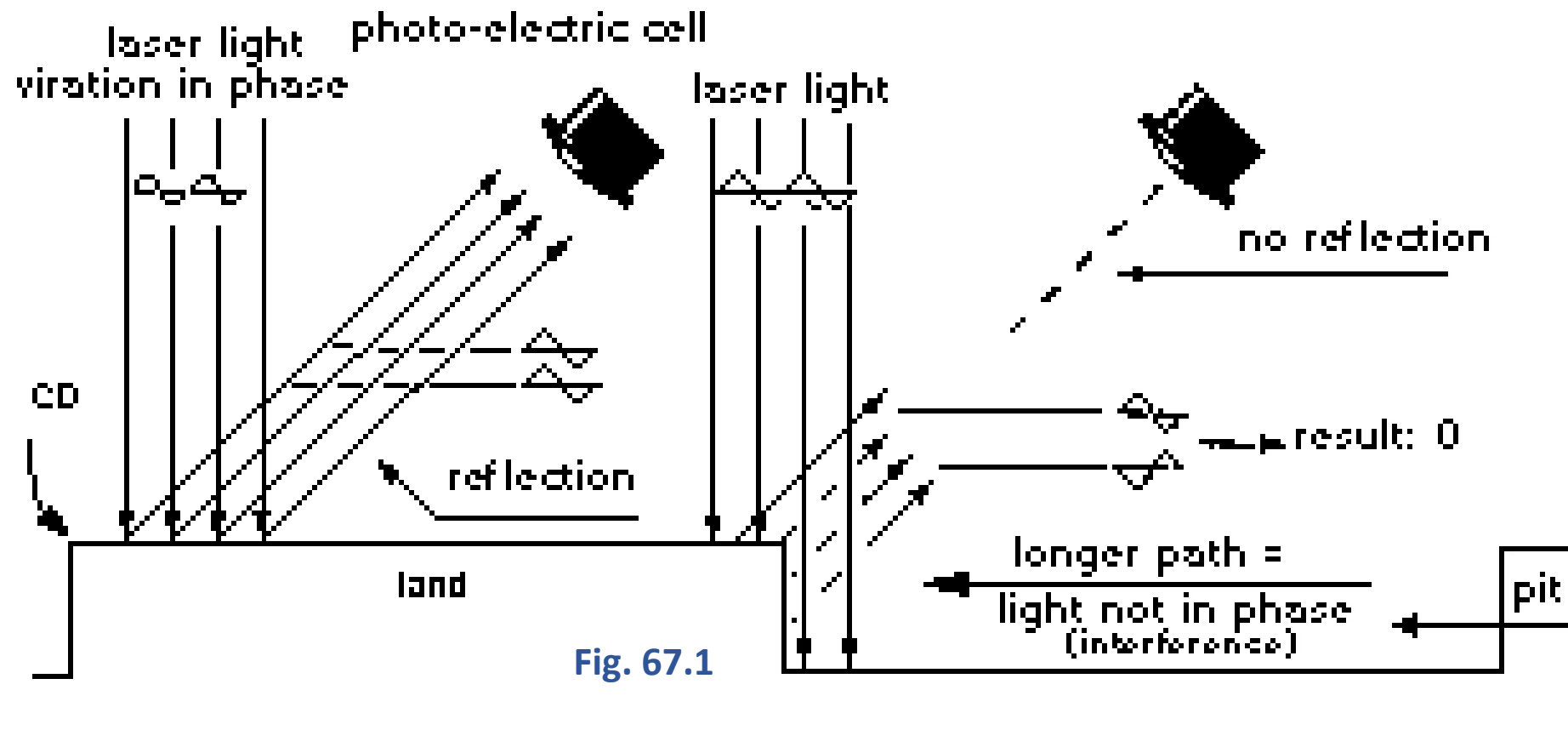


Fig. 66.1

CD ROM. Modificarea coeficientului de reflecție în procesul de citire a "1" logic:



CD ROM. Structura capului de citire a datelor:

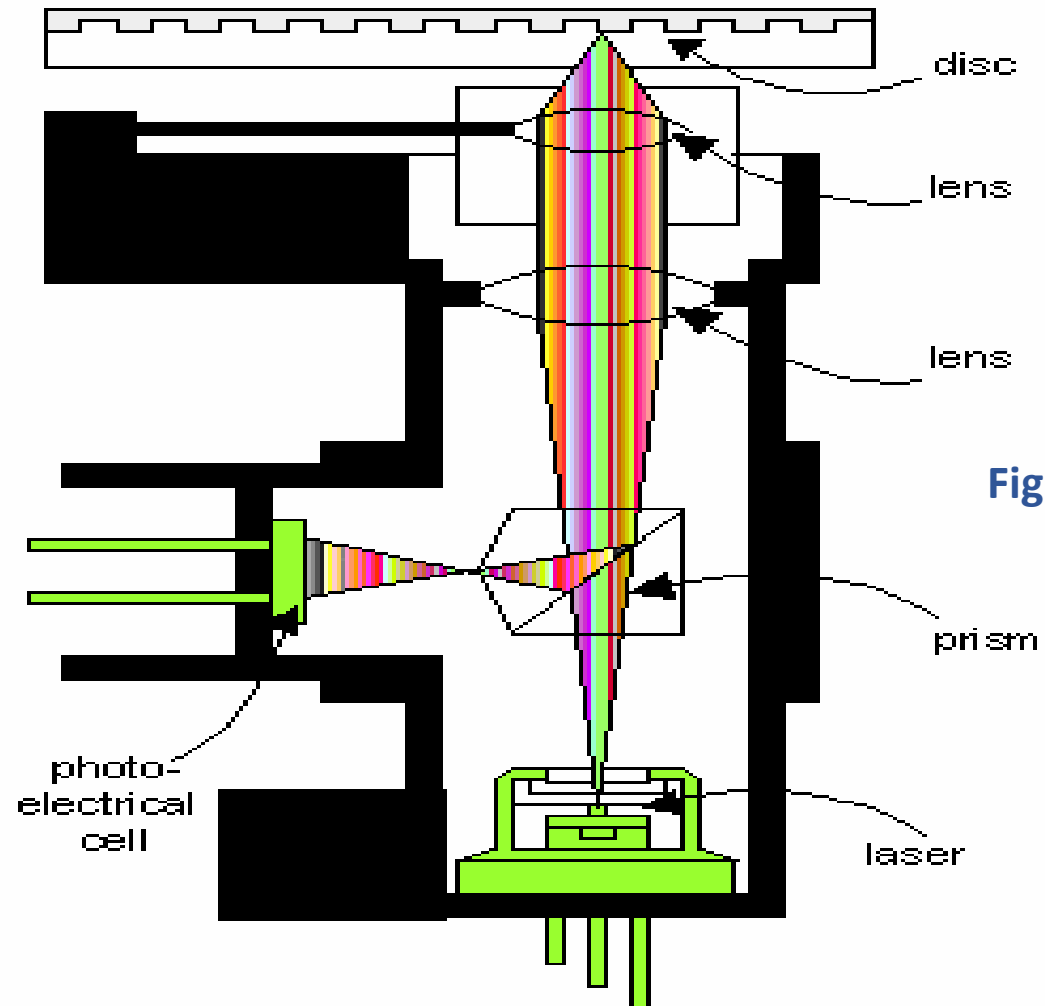


Fig. 68.1

CD ROM. Structura pistei în formă de spirală:

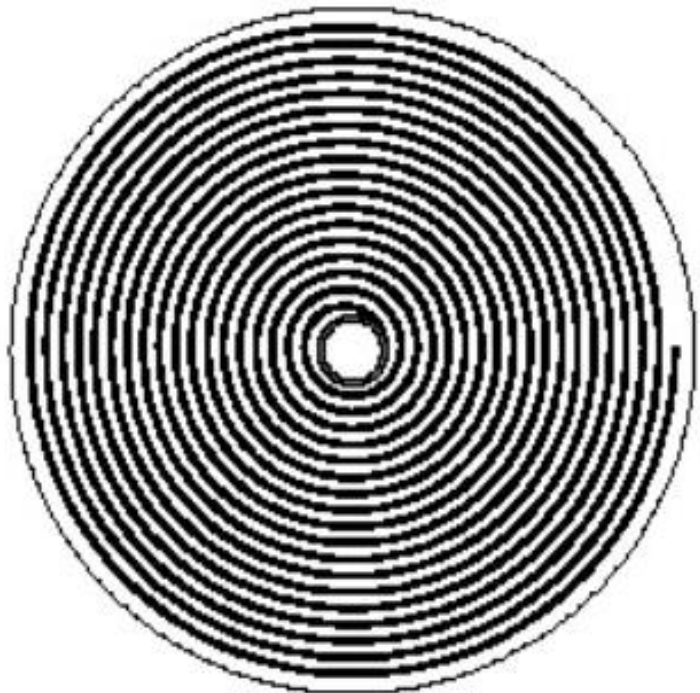


Fig. 69.1

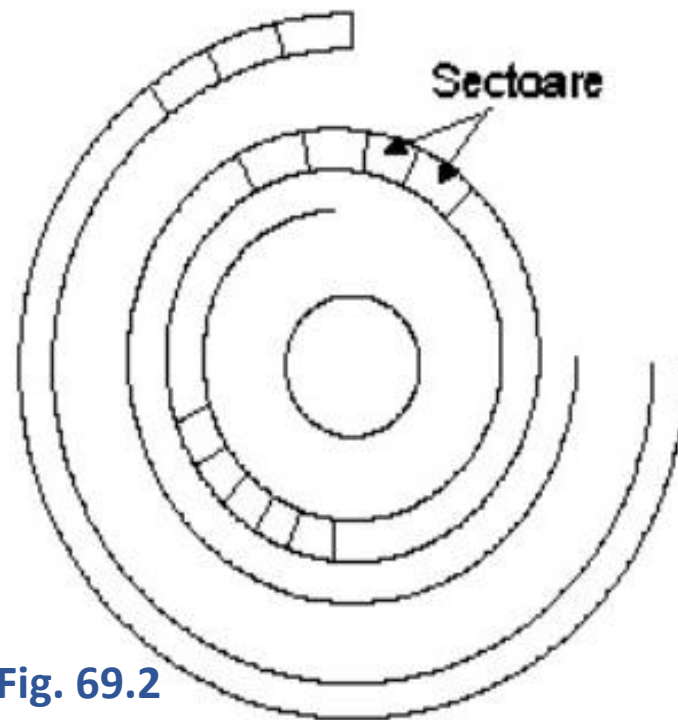


Fig. 69.2

CD ROM. Structura pistei înregistrate. Pit-ul. Dimensiunile:

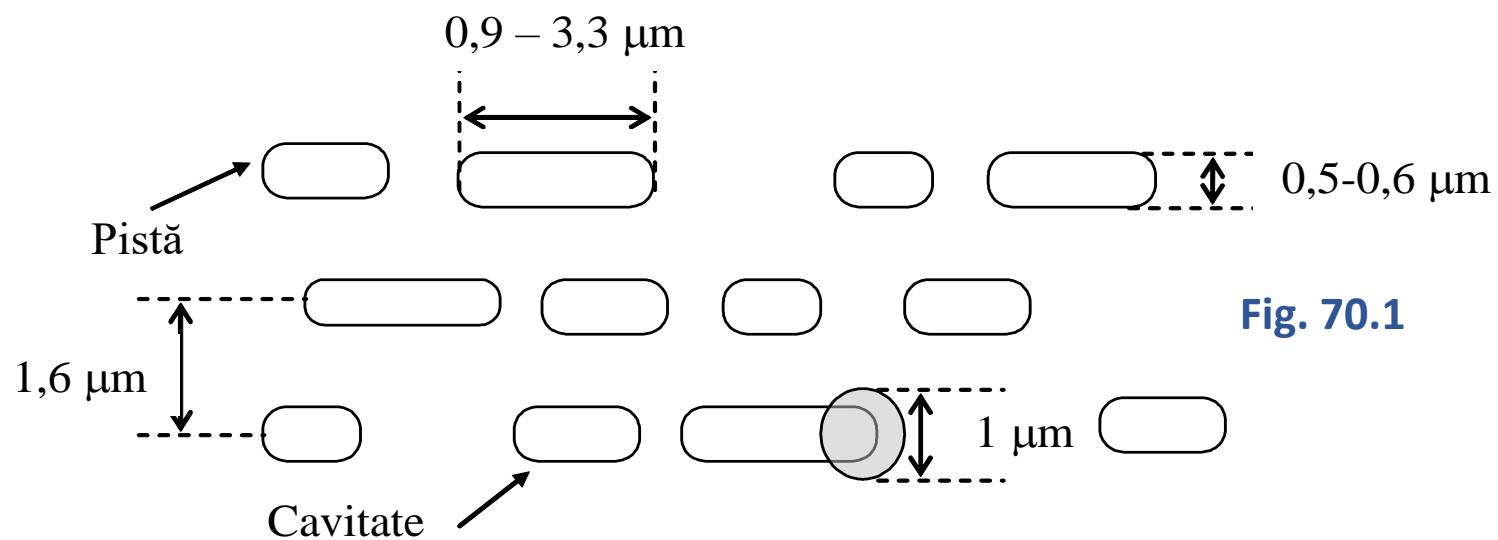


Fig. 70.1

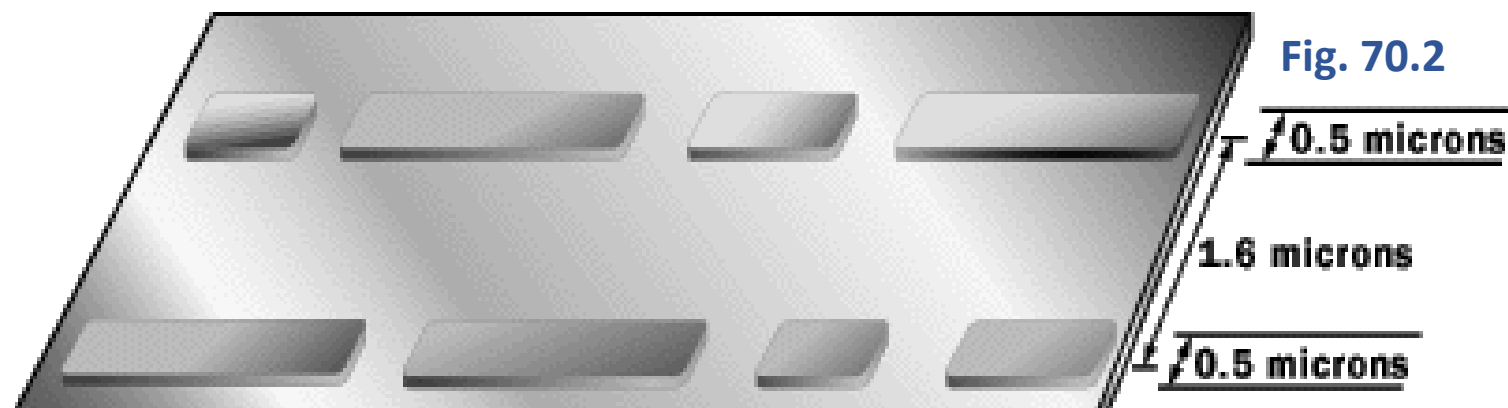
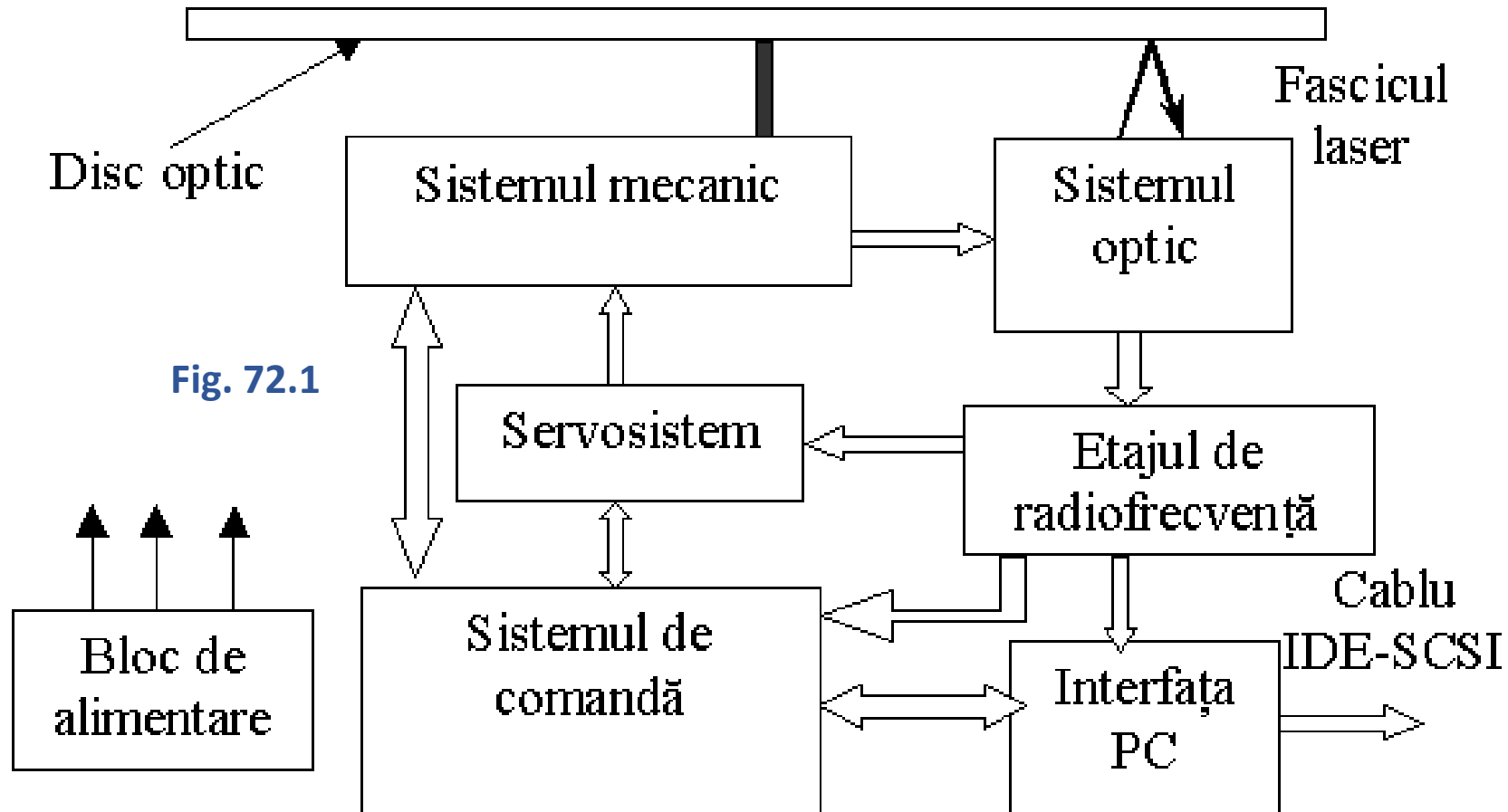


Fig. 70.2

CD ROM. Comparații pentru CD și DVD:

Caracteristica	CD	DVD
Diametrul discului (mm)	120	120
Grosimea discului (mm)	1,2	0,6
Spațiul între 2 piste adiacente (μm)	1,6	0,74
Lungimea minimă a cavității (μm)	0,9	0,4
Lungimea de undă a laserului (nm)	780	640
Capacitate pe strat (GB)	0,64	4,4
Număr de straturi	1	1-4

CD ROM. Structura dispozitivului pentru citirea Discurilor Optice:



CD ROM. Structura internă a dispozitivului pentru citirea Discurilor Optice:

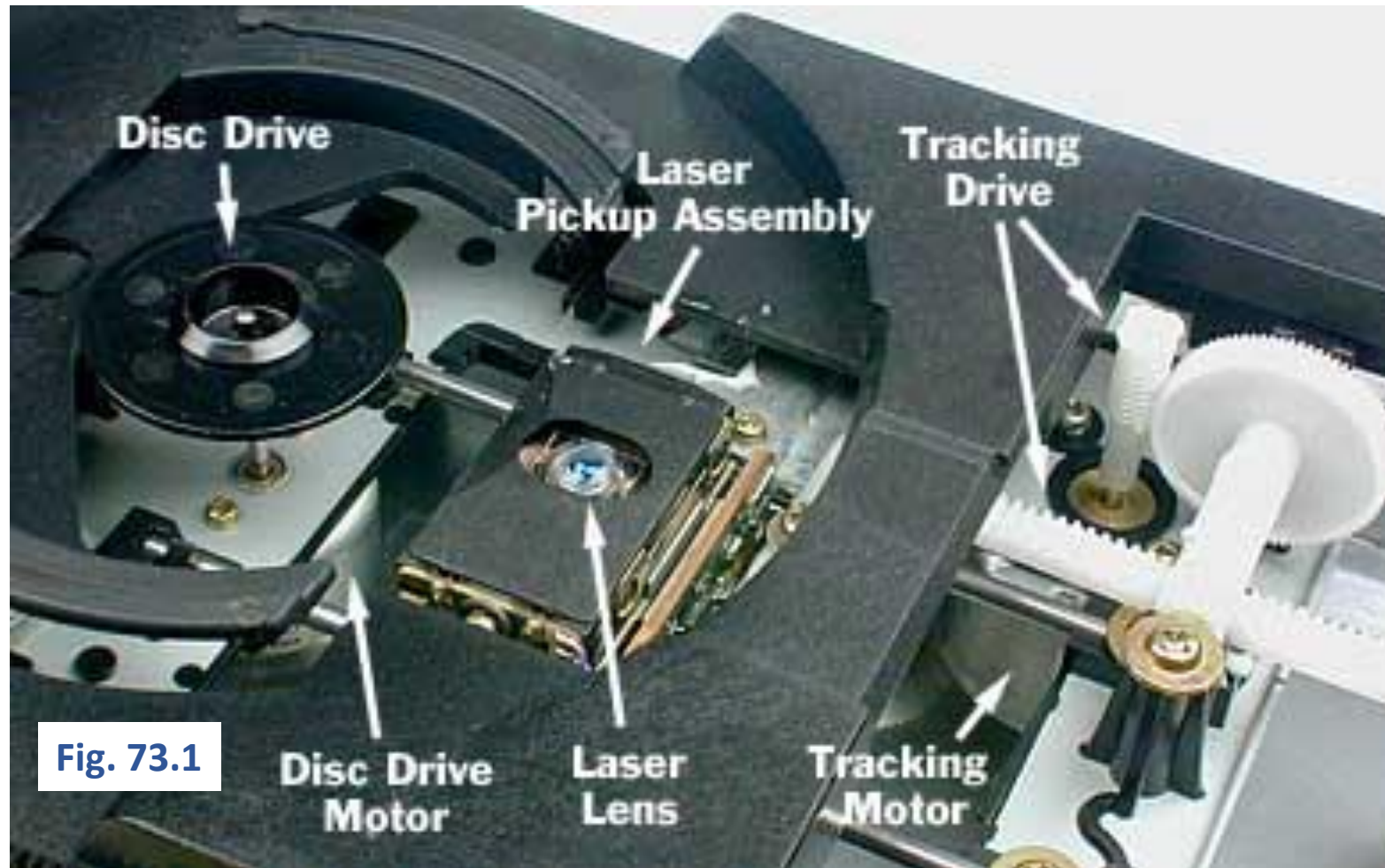


Fig. 73.1

Memorii externe: Flash, SD Card.



Fig. 74.1



Fig. 74.2

Porturi de comunicare:
Flash – USB;
SD Card – SPI.

Tehnologia RAID:

Pentru îmbunătățirea vitezei de transfer și a performanțelor unităților de hard-disc în general, pe lângă memoria cache se pot folosi și ansambluri de discuri. Există multe situații când în locul unei unități de mare capacitate se folosește un ansamblu de discuri de capacități mai mici. Acest lucru asigură posibilitatea accesării în paralel a mai multor fișiere dacă discurile sunt utilizate în mod individual sau a unui fișier împărțit pe mai multe discuri.

Conceptul RAID - *Redundant Array of Inexpensive Disks* (sau, mai nou, *Redundant Array of Independent Disks*) – apărut în anul 1988 a demonstrat că folosirea unui set de astfel de unități de hard-discuri de mică capacitate, interconectate într-un mod inteligent poate conduce la o creștere a accesibilității datelor și la o reducere a costurilor, în special în sistemele unde pierderea informației datorită defectării ar reprezenta o catastrofă.

Tehnologia RAID. Nivelul RAID:

Nivelul RAID 0 - asigură distribuirea informațiilor pe mai multe discuri dar nu asigură nici o redundanță.

Nivelul RAID 1 - asigură o rezervă egală cu capacitatea utilă de stocare. Pentru fiecare disc există un disc oglindă pe care se află aceleași date.

Nivelul RAID 2 - este inspirat din codarea Hamming (care adăuga la cuvântul util biți redundanți). Dacă dorim să protejăm informația utilă aflată pe m discuri, trebuie folosite k discuri suplimentare astfel încât să se respecte relația: $2^k \geq k + m + 1$.

Nivelul RAID 3 - este inspirat din codarea cuvintelor de date cu bit de paritate. Astfel datele sunt distribuite la nivel de bit pe discurile de date iar un disc este folosit pentru stocarea bitului de paritate.

Tehnologia RAID. Nivelul RAID:

Nivelul RAID 4 - este asemănător cu nivelul RAID 3 cu deosebirea că distribuirea datelor pe discuri se face în blocuri și nu la nivel de bit.

Nivelul RAID 5 - provine din nivelul RAID 3 și asigură o mai bună accesabilitate a informației prin distribuirea bitului de paritate pe mai multe discuri, nemaexistând astfel un disc dedicat parității.

Nivelul RAID 6 - corespunde utilizării unei matrici de discuri sau chiar dispunerii lor pe trei dimensiuni. Pe fiecare dimensiune se poate crea un bit de paritate ce poate fi stocat pe un disc dedicat sau poate fi distribuit pe discurile din dimensiunea respectivă.

; Sarcina de proiectare:

; Sa se elaboreze programul care permite functiile definite pentru un Manager de fisiere.

; Creare File / Dirr., Copiere File, Vizuare File, Stergere File / Dirr.

; Operatiile sunt efectuate in regim de dialog:

; R- Read File; W- Write File; C- Create;

; D- Delete_File; T- Input Text; F- afisare continut File;

; L- Close; P- Copieie File to File; N- Rename File;

; M - Creare Dir; I - Trecere in directoriu;

; U - Sterge directoriu;

; N - Creare unui Fisier Nou cu un nume Introdus de la KBD;

; T - Introducerea unui text pentru inscriere in File;

; Q - terminarea programului.


```
name "fileio"
org 100h
jmp start
dir1 db "c:\test3", 0
dir2 db "c:\test2", 0
dir3 db "c:\newname", 0
file1 db "c:\file1.txt", 0
file2 db "c:\newfile.txt", 0
file3 db "c:\t1.txt", 0
handle dw ?
text db "Datele pentru inscriere in fisier: Text1, Text1, Grupa CR-191, $"
text_size = $ - offset text
text2 db "hi!, Aceasta este lucrarea Nr 6. Grupa, Varianta 5, $"
text2_size = $ - offset text2
text_student db 0Dh, 0Ah, "Aceasta este lucrarea de laborator Nr 6 INT 21h: $"
text_student1 db 0Dh, 0Ah, "Studentul Grupei ...: $"
text_output db 0Dh, 0Ah, "Introduceti Comanda pentru operatiile cu HDD: $"
erroare db 0Dh, 0Ah, "S-a comis o eroare in procesul executarii programului, $"
Not_Com db 0Dh, 0Ah, "S-a introdus o comanda nedefinita. $"
Exec_Com db 0Dh, 0Ah, "Comanda indicata s-a executat. $"
S1 db 200,?, 200 dup(' ')
T2 db 200,?, 200 dup(' ')
FileName db 30,?, 30 dup(' '); Acesta este numele fisierului creat.
COM db ?
```

```

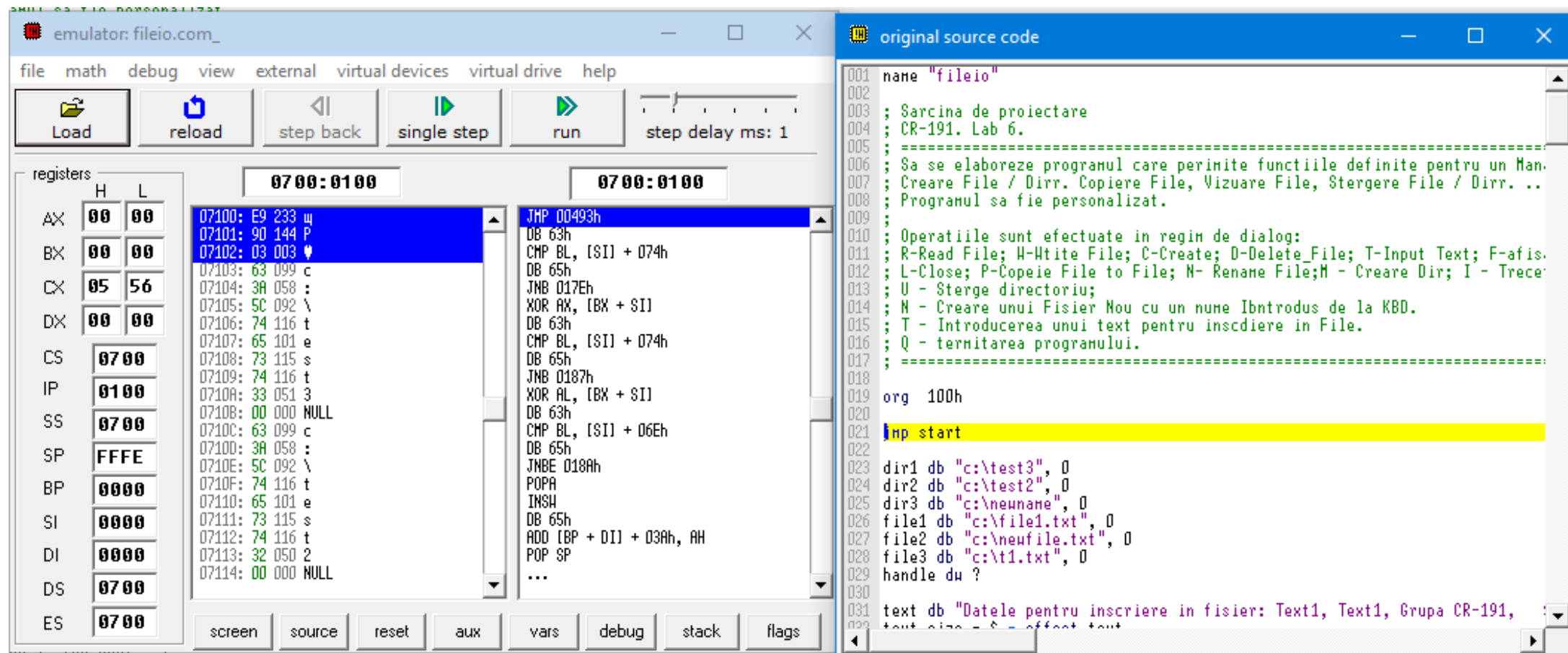
start:
mov ax, cs
mov dx, ax
mov es, ax
MOV BP, 10h
; Salutare program si invitatie
MOV DX, offset text_student
MOV AH, 9
INT 21h
MOV DX, offset text_student1
MOV AH, 9
INT 21h
Repeat:
MOV DX, offset text_output
MOV AH, 9
INT 21h
MOV AH, 00h
INT 16h
MOV COM, AL
CMP AL, "r"
JE Read_File
MOV AL, COM
CMP AL, "c"
JE Create_File
MOV AL, COM
CMP AL, "m"
JE Create_Dir
MOV AL, COM
CMP AL, "w"
JE Write_File
MOV AL, COM
CMP AL, "d"
JE Delete_File
MOV AL, COM
CMP AL, "t"
JE New_Text
MOV AL, COM
CMP AL, "n"
JE New_File
MOV AL, COM
CMP AL, "q"
JE Quit
MOV DX, offset Not_Com
MOV AH, 9
INT 21h

DEC BP
JNZ Repeat
Read_File:
; Citire din fisier
MOV DX, offset Exec_Com
MOV AH, 9
INT 21h
DEC BP
JMP Repeat
Create_Dir:
; create c:\emu8086\vdive\C\test1
mov dx, offset dir1
mov ah, 39h
int 21h
MOV DX, offset Exec_Com
MOV AH, 9
INT 21h
DEC BP
JMP Repeat
Create_File:
; create and open file: ;c:\emu8086\vdive\C\test1\file1.txt
mov ah, 3ch
mov cx, 0
mov dx, offset file1
int 21h
jc err
mov handle, ax
MOV DX, offset Exec_Com
MOV AH, 9
INT 21h
DEC BP
JMP Repeat
Write_File:
; create and open file: ;c:\emu8086\vdive\C\test1\file1.txt
mov ah, 3ch
mov cx, 0
mov dx, offset file1
int 21h
jc err
mov handle, ax
; write to file 1:
mov ah, 40h
mov bx, handle
mov dx, offset text
mov cx, text_size

int 21h
; write to file 2:
mov ah, 40h
mov bx, handle
mov dx, offset text2
mov cx, text2_size
int 21h
; close c:\emu8086\vdive\C\test1\file1.txt
mov ah, 3eh
mov bx, handle
int 21h
MOV DX, offset Exec_Com
MOV AH, 9
INT 21h
DEC BP
JMP Repeat
Delete_File:
; delete file c:\emu8086\MyBuild\t1.txt
mov ah, 41h
mov dx, offset file1
int 21h
MOV DX, offset Exec_Com
MOV AH, 9
INT 21h
DEC BP
JMP Repeat
err:
MOV DX, offset eroare
MOV AH, 9
INT 21h
nop
MOV DX, offset Not_Com
MOV AH, 9
INT 21h
New_Text:
jmp m1
msg1 db 0Dh, 0Ah, "Introduceti Textul Nou pentru inscriere in File: $"
m1: mov dx, offset msg1
mov ah, 9
int 21h
; input a string:
mov dx, offset s1
mov ah, 0ah
int 21h
DEC BP

JMP Repeat
New_File:
jmp m2
msg2 db 0Dh, 0Ah, "Introduceti Numele Fisierului: $"
m2: mov dx, offset msg2
mov ah, 9
int 21h
; input filename:
mov dx, offset filename
mov ah, 0ah
int 21h
; set 0 to the end of the filename:
xor bx, bx
mov bl, filename[1] ; get actual size.
mov filename[bx+2], 0
; create new file:
mov cx, 0
mov ah, 3ch
mov dx, offset filename+2
int 21h
jc err
mov handle, ax
; write buffer to file:
mov ah, 40h
mov bx, handle
mov dx, offset s1+2
xor cx, cx
mov cl, s1[1]
int 21h
jc err
; close file
mov bx, handle
mov ah, 3eh
int 21h
jc err
DEC BP
JMP Repeat
Quit:
ret

```



The image shows a screenshot of an x86 emulator window titled "emulator: fileio.com_". The interface includes a menu bar (file, math, debug, view, external, virtual devices, virtual drive, help) and a toolbar with buttons for Load, reload, step back, single step, run, and step delay ms: 1. Below the toolbar is a registers window showing the state of various registers (AX, BX, CX, DX, CS, IP, SS, SP, BP, SI, DI, DS, ES) with their high (H) and low (L) bytes. The main window displays assembly code with memory addresses and instructions. The current instruction is `JMP 00493h`. To the right, a separate window titled "original source code" shows the high-level assembly code with comments in Romanian, including a program name "fileio", a description of the task, and a list of operations (R-Read File, W-Write File, C-Create, D-Delete File, T-Input Text, F-afis., L-Close, P-Copeie File to File, M- Rename File, M - Create Dir, I - Trece).

Registers:

Register	H	L
AX	00	00
BX	00	00
CX	05	56
DX	00	00
CS	0700	
IP	0100	
SS	0700	
SP	FFFE	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

Assembly Code (Current View):

```

07100: E9 233 w
07101: 90 144 P
07102: 03 003 v
07103: 63 099 c
07104: 3A 058 :
07105: 5C 092 \
07106: 74 116 t
07107: 65 101 e
07108: 73 115 s
07109: 74 116 t
0710A: 33 051 3
0710B: 00 000 NULL
0710C: 63 099 c
0710D: 3A 058 :
0710E: 5C 092 \
0710F: 74 116 t
07110: 65 101 e
07111: 73 115 s
07112: 74 116 t
07113: 32 050 2
07114: 00 000 NULL
JMP 00493h
DB 63h
CMP BL, [SI] + 074h
DB 65h
JNB 017Eh
XOR AX, [BX + SI]
DB 63h
CMP BL, [SI] + 074h
DB 65h
JNB 0187h
XOR AL, [BX + SI]
DB 63h
CMP BL, [SI] + 06Eh
DB 65h
JNBE 018Ah
POPA
INSH
DB 65h
ADD [BP + DI] + 03Ah, AH
POP SP
...

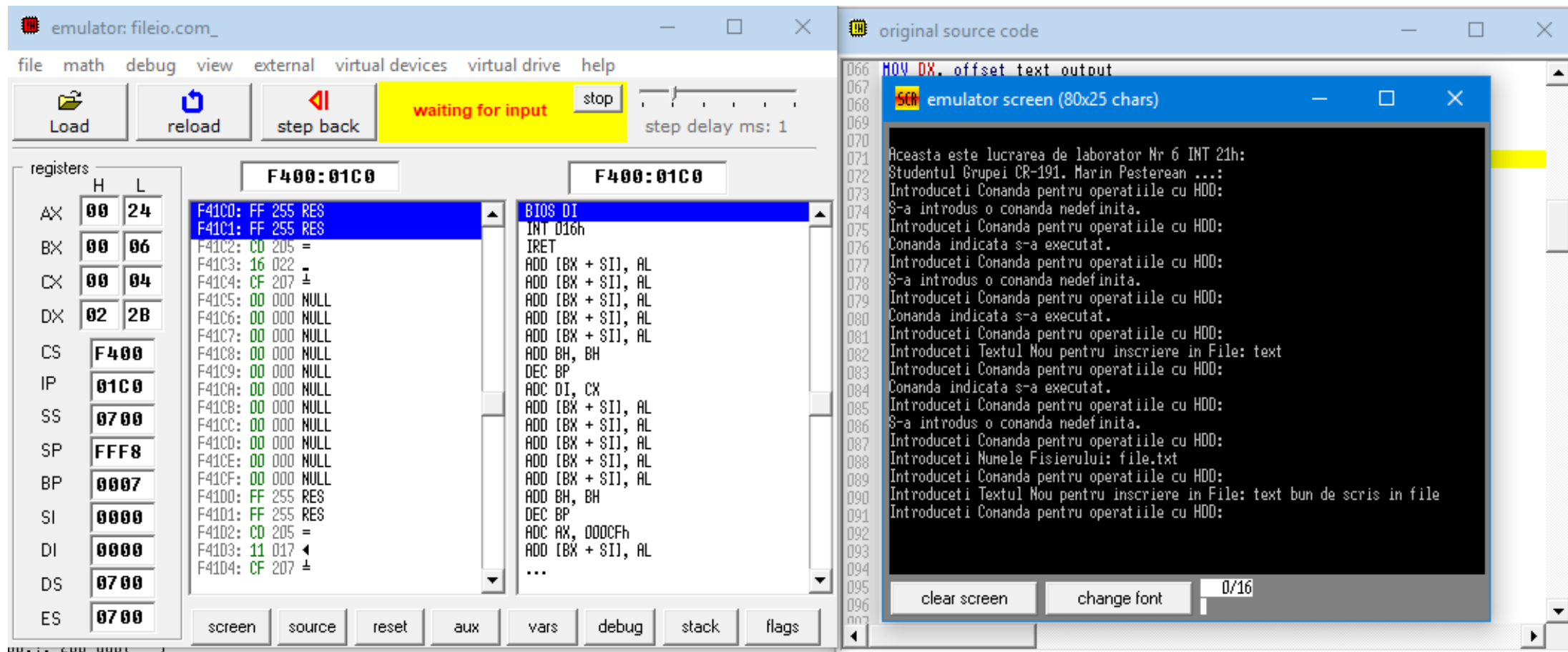
```

Original Source Code:

```

001 name "fileio"
002
003 ; Sarcina de proiectare
004 ; CR-191. Lab 6.
005 ;
006 ; Sa se elaboreze programul care permite functiile definite pentru un Man.
007 ; Creare File / Dirr. Copiere File, Vizuire File, Stergere File / Dirr. ...
008 ; Programul sa fie personalizat.
009 ;
010 ; Operatiile sunt efectuate in regin de dialog:
011 ; R-Read File; W-Write File; C-Create; D-Delete File; T-Input Text; F-afis.
012 ; L-Close; P-Copeie File to File; M- Rename File;M - Create Dir; I - Trece
013 ; U - Sterge directoriu;
014 ; N - Creare unui Fisier Nou cu un nume Ibtrodus de la KBD.
015 ; T - Introducerea unui text pentru inscdiere in File.
016 ; Q - termitarea programului.
017 ;
018
019 org 100h
020
021 jmp start
022
023 dir1 db "c:\test3", 0
024 dir2 db "c:\test2", 0
025 dir3 db "c:\neunane", 0
026 file1 db "c:\file1.txt", 0
027 file2 db "c:\newfile.txt", 0
028 file3 db "c:\t1.txt", 0
029 handle du ?
030
031 text db "Datele pentru inscriere in fisier: Text1, Text1, Grupa CR-191, ..."
032 text size = $ - offset text

```



The image shows a screenshot of an x86 emulator interface with three main windows:

- emulator: fileio.com_ (Main Window):**
 - Menu: file, math, debug, view, external, virtual devices, virtual drive, help
 - Buttons: Load, reload, step back, stop, step delay ms: 1
 - Status: waiting for input
 - Registers:

Register	H	L
AX	00	24
BX	00	06
CX	00	04
DX	02	2B
CS	F400	
IP	01C0	
SS	0700	
SP	FFF8	
BP	0007	
SI	0000	
DI	0000	
DS	0700	
ES	0700	
 - Memory: F400:01C0
 - Source Code Window (BIOS.DI):


```

INT 016h
IRET
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD BH, BH
DEC BP
ADC DI, CX
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD BH, BH
DEC BP
ADC AX, 000CFh
ADD [BX + SI], AL
          
```
 - original source code (Background):


```

066 MOV DX, offset text output
067
068
069
070
071 Aceasta este lucrarea de laborator Nr 6 INT 21h:
072 Studentul Grupei CR-191. Marin Pesterean ...:
073 Introduceti Comanda pentru operatiile cu HDD:
074 S-a introdus o comanda nedefinita.
075 Introduceti Comanda pentru operatiile cu HDD:
076 Comanda indicata s-a executat.
077 Introduceti Comanda pentru operatiile cu HDD:
078 S-a introdus o comanda nedefinita.
079 Introduceti Comanda pentru operatiile cu HDD:
080 Comanda indicata s-a executat.
081 Introduceti Comanda pentru operatiile cu HDD:
082 Introduceti Textul Nou pentru inscriere in File: text
083 Introduceti Comanda pentru operatiile cu HDD:
084 Comanda indicata s-a executat.
085 Introduceti Comanda pentru operatiile cu HDD:
086 S-a introdus o comanda nedefinita.
087 Introduceti Comanda pentru operatiile cu HDD:
088 Introduceti Numele Fisierului: file.txt
089 Introduceti Comanda pentru operatiile cu HDD:
090 Introduceti Textul Nou pentru inscriere in File: text bun de scris in file
091 Introduceti Comanda pentru operatiile cu HDD:
092
093
094
095
096
097
          
```
 - emulator screen (80x25 chars) (Foreground):


```

Acesta este lucrarea de laborator Nr 6 INT 21h:
Studentul Grupei CR-191. Marin Pesterean ...:
Introduceti Comanda pentru operatiile cu HDD:
S-a introdus o comanda nedefinita.
Introduceti Comanda pentru operatiile cu HDD:
Comanda indicata s-a executat.
Introduceti Comanda pentru operatiile cu HDD:
S-a introdus o comanda nedefinita.
Introduceti Comanda pentru operatiile cu HDD:
Comanda indicata s-a executat.
Introduceti Comanda pentru operatiile cu HDD:
Introduceti Textul Nou pentru inscriere in File: text
Introduceti Comanda pentru operatiile cu HDD:
Comanda indicata s-a executat.
Introduceti Comanda pentru operatiile cu HDD:
S-a introdus o comanda nedefinita.
Introduceti Comanda pentru operatiile cu HDD:
Introduceti Numele Fisierului: file.txt
Introduceti Comanda pentru operatiile cu HDD:
Introduceti Textul Nou pentru inscriere in File: text bun de scris in file
Introduceti Comanda pentru operatiile cu HDD:
          
```

Tema 5. Dispozitive pentru stocarea datelor :

1. Clasificarea dispozitivelor pentru stocarea datelor.
2. Sisteme de fișier: FAT, NTFS, EXTx.
3. Funcții BIOS și DOS pentru managementul datelor.
4. FDD. Programarea operațiilor de intrare/ieșire.
5. HDD. Programarea operațiilor de intrare/ieșire.
6. CD ROM.
7. Memorii externe: Flash, SD Card, SDD.
8. Tehnologii RAID.

Tematica disciplinei Arhitectura Calculatoarelor:

Tema 1. Introducere. Bazele fundamentale ale Arhitecturii Calculatoarelor;

Tema 2. Microprocesoare și Microcontrolere. Limbajul de programare Assembler;

Tema 3. Dispozitive pentru achiziția datelor;

Tema 4. Dispozitive pentru afișarea și imprimarea datelor;

Tema 5. Dispozitive pentru stocarea datelor.

Mulțumesc pentru atenție!