

LOGIČKA STRUKTURA TVRDOG DISKA

Sadržaj

1.	Općenito o tvrom disku	2
2.	Unutarnja struktura diska	2
3.	Operacijski sustavi	3
3.1.	DOS(MS-DOS)	3
3.2.	Windows 3.1x	3
3.3.	Windows 95	3
3.4.	Windows NT	3
3.5.	Windows 2000	4
4.	Datotečni sustavi.....	4
5.1.	FAT	4
5.2.	VFAT	4
5.3.	NTFS	4
5.4.	Manje poznati	4
6.	Logička struktura.....	5
7.	MBR(Master Boot Record).....	6
7.1.	MBR code	7
7.2.	Particijska tablica	8
7.3.	Magična riječ.....	11
8.	RT0 (Reminder of Track 0)	12
9.	Particije	12
9.1.	Primarna particija.....	13
9.1.1	Boot Sector primarne particije.....	13
9.1.2.	FAT1 i FAT2.....	16
9.1.3.	Root Directory	16
9.1.4.	Data Area.....	17
9.2.	Proširena particija	17
10.	Clusteri i alokacijska datoteka	18
11.	Defragmentacija	19
12.	PC – Boot proces	21
13.	Formiranje particije.....	22
13.1.	Microsoft Fdisk.....	22
13.2.	Linux Fdisk	22
13.3.	Partition Magic	23
13.4.	Formatiranje	23

1. Općenito o tvrom disku

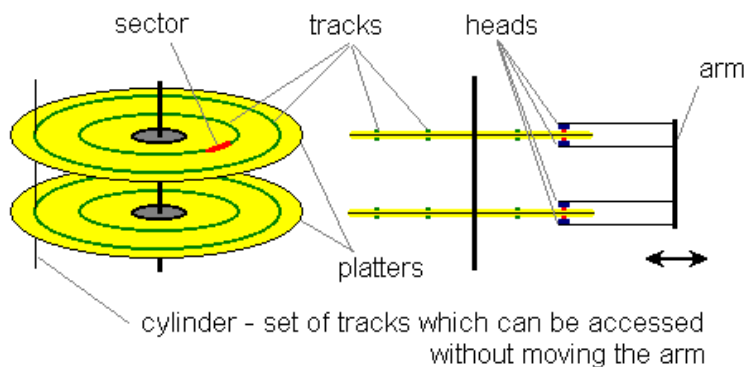
Tvrđi disk je vrsta medija koja služi za pohranu podataka. Budući da je potražnja tržišta sve veća kapacitet im se konstantno povećava, ali osnovni princip rada je isti. Podaci koji se pohranjuju na tvrdi disk pohranjuju se trajno, odnosno tako dugo dok ih korisnik ne odluči izbrisati. Ali i nakon toga oni ostaju na tvrdom disku tako dugo dok korisnik preko njih ne pohrani druge informacije. Tvrđi disk služi za pohranu operacijskog sustava (jednog ili više njih ako je disk particioniran u više od jedne particije). Mnogi kupci ponekad imaju problema prilikom kupnje tvrdog diska jer im kadkad operacijski sustav ne podržava puni kapacitet tvrdog diska (npr. Operacijski sustav DOS 6.22 i ranije verzije podržavaju diskveličine jednake ili manje od 2048 MB), dakle ukoliko imamo veći disk potrebno ga je particionirati. Ali može postojati problem i u BIOS –u. U tom se slučaju FLASH ROM napuni novim sadržajem koji će podržavati puni kapacitet tvrdog diska.



Slika 1: Tvrđi disk

2. Unutarnja struktura diska

Tvrđi disk čine ploče (*platters*) koje su učvršćene na rotor. Sastoje se od supstrata koji je nemagnetičan i obično se radi od aluminijske ili mješavine stakla i keramike. Kako bi se omogućilo zapisivanje podataka, obje strane ploče premazane su magnetskim medijem. Nekada je to bio željezni oksid, a danas je to gotovo isključivo tanki magnetski metalni film. Ploče su najčešće veličine 3.5 inča i 5.25 inča. Ploče (*platters*) diska učvršćene su na rotor koji ih može rotirati nekoliko tisuća puta u minuti. Između ploča nalazi se prostor za magnetske glave za čitanje i pisanje tako da se sa svake strane ploče nalazi se po jedna glava za čitanje i pisanje. Glave su toliko blizu pločama da bi i najmanja nečistoća na glavi mogla uzrokovati da pri rotiranju dođe do trajnog uništenja magnetskog materijala na pločama i do uništenja podataka. Glave za čitanje i pisanje zapravo lebde između ploča a drži ih protok zraka koji stvaraju rotirajuće ploče. Za svaku stranu ploče postoji jedna glava za čitanje i pisanje i sve one fiksirane su na jednoj pomičnoj ručici koja ih može pomicati od ruba diska do rotora i natrag. Ta pomična ručica naziva se i *aktuator*,



Slika 2: Unutarnji prikaz strukture HDD –a

3. Operacijski sustavi

Za svaki operacijski sustav karakterističan je datotečni sustav koji on koristi. Neki operacijski sustavi mogu koristiti više datotečnih sustava, dok drugi operacijski sustavi rade samo sa svojim datotečnim sustavom

3.1. DOS(MS-DOS)

Dos verzije 6.22 i ranije koriste FAT datotečni sustav, dakle tu spadaju FAT12 i FAT16 sustavi. Windows95 dolazi sa ugrađenim DOS v7.0. Windows 95 OEM *Service Release 2* (OSR2) podržavaju i datotečni sustav FAT32 i zbog toga koriste DOS v7.1. Kod sustava FAT12 koristi se 12 –bitni broj za adresiranje clustera, dok se kod FAT16 koristi 16 –bitni broj za adresiranje clustera.

3.2. Windows 3.1x

Windows 3.1 se ne može nazvati prvim višezadačnim sustavom, on je neka ljuska koja se izvršava iznad DOS –a, i za većinu stvari koristi DOS. Budući da njegov DOS podržava FAT16, takav datotečni sustav koristi i sami Windows 3.1. Zadnja verzija Windowsa 3.11 for Workgroups sadržavala je u sebi mogućnost nazvanu *32-bit file access*, dakle pristup podacima uz pomoć 32-bitnih rutina. To je zapravo bila prva verzija VFAT sustava kojeg koriste Windows 9x operacijski sustavi.

3.3. Windows 95

Windows 95 slijedi iza windows 3.11. On na neki način ima svoj pristup disku, a s druge strane još uvijek koristi DOS verzije 7.0. To je razlog zbog kojeg Windows ima lošije performanse, dok je bolja strana što je kompatibilan sa starijim softverom. Stvarni datotečni sustav koji koristi Windows 95 jest VFAT, koji je podržan i od strane Windows 95 i od strane DOS-a v7.0 s kojim se Windows 95 isporučuje. Počevši od Windows 95 OSR2 također Windows podržava i datotečni sustav FAT32 koji omogućuje korištenje većih particija na disku.

3.4. Windows NT

Windows NT je verzija Windows –a koja je rađena iznova i nije zasnovana na DOS–u. Može koristiti dva datotečna sustava. Prvi je *NTFS (New Tehnology File System)* koji je vrlo sličan UNIX –ovom sustavu, dok je drugi standardni *FAT16*. Podrška za FAT je uključena iz razloga kompatibilnosti i radi konfiguracije PC sustava na kojem se izvršava više operacijskih sustava. Starije verzije Windows –a NT podržavale su i sustav *HPFS(High Performance File System)*, koji se više ne podržava od verzije 4 na dalje.

3.5. Windows 2000

Windows 2000 podržavaju čak tri datotečna sustava. To su FAT16, FAT32, te sustav NTFS. FAT16 i FAT32 podržavaju se i ovdje radi kompatibilnost, no ovdje je i jedna nova struktura nazvana dinamički disk. Uvedena je radi modernijih načina rada kod kojih se postiže veća otpornost na nepravilnosti (*fault tolerant disk structure*).

4. Datotečni sustavi

Datotečni sustav (*File System*) je pojam koji se odnosi na logičke strukture i programske rutine kojima se podaci zapisuju i čitaju sa diska. Razičiti operacijski sustavi imaju razičite datotečne sustave, neki koriste samo jedan datotečni sustav

tj. samo svoj datotečni sustav dok drugi podržavaju više datotečnih sustava radi kompaktibilnosti s drugim operativnim sustavima.

5.1. FAT

Prijevod FAT-a (File allocation table) je alokacijska tablica podataka. Ova tablica ne predstavlja cijeli datotečni sustav, ona je dio datotečnog sustava koji operacijski sustav koristi za pronalaženje podataka na disku. Kad se kaže FAT sustav često se misli na FAT12 odnosno FAT16 sustav kojeg koriste tvrdi diskovi. FAT32 je poboljšanje FAT16 sustava, a napravljen je zbog povećanja kapaciteta diskova. Dakle sustav FAT16 podržava tvrdi disk ili [particiju](#) do 2GB, dok FAT32 može upravljati i sa 8GB ali i više koristeći svoju osnovnu strukturu. Uz sve to FAT32 sustav podržava manje clusterne, pa je tako iskoristivost diska veća, ali to može smanjiti brzinu rada jer se treba pročitati veći broj adresa clustera za neku datoteku. FAT32 i FAT16 datotečni sustavi su po strukturi i načinu rada isti samo što FAT32 koristi 32 bitni zapis u FAT tablici za svaki cluster na disku, a FAT16 koristi 16 bitni zapis o clusterima.

5.2. VFAT

VFAT ili Virtual FAT je sustav koji podržava dugačka imena datoteka. On je zapravo *driver* za rad s dugim imenima unutar FAT sustava. Prema tome u ovakvom sustavu pristup podacima omogućen je pomoću 32-bitnih drivera koji rade u zaštićenom modu. VFAT zapisuje imena na klasični DOS –ov način (8+3), dok operacijski sustav koristi dodatnu transformacijsku tablicu za prikaz dugačkih imena.

5.3. NTFS

NTFS (New Tehnology File System) je sustav koji je u potpunosti nekompatibilan sa FAT sustavom. Koriste ga Windows NT i u zadnje vrijeme Windows XP. Do sada se uglavnom koristio na serverima i na računalima u tvrtkama.

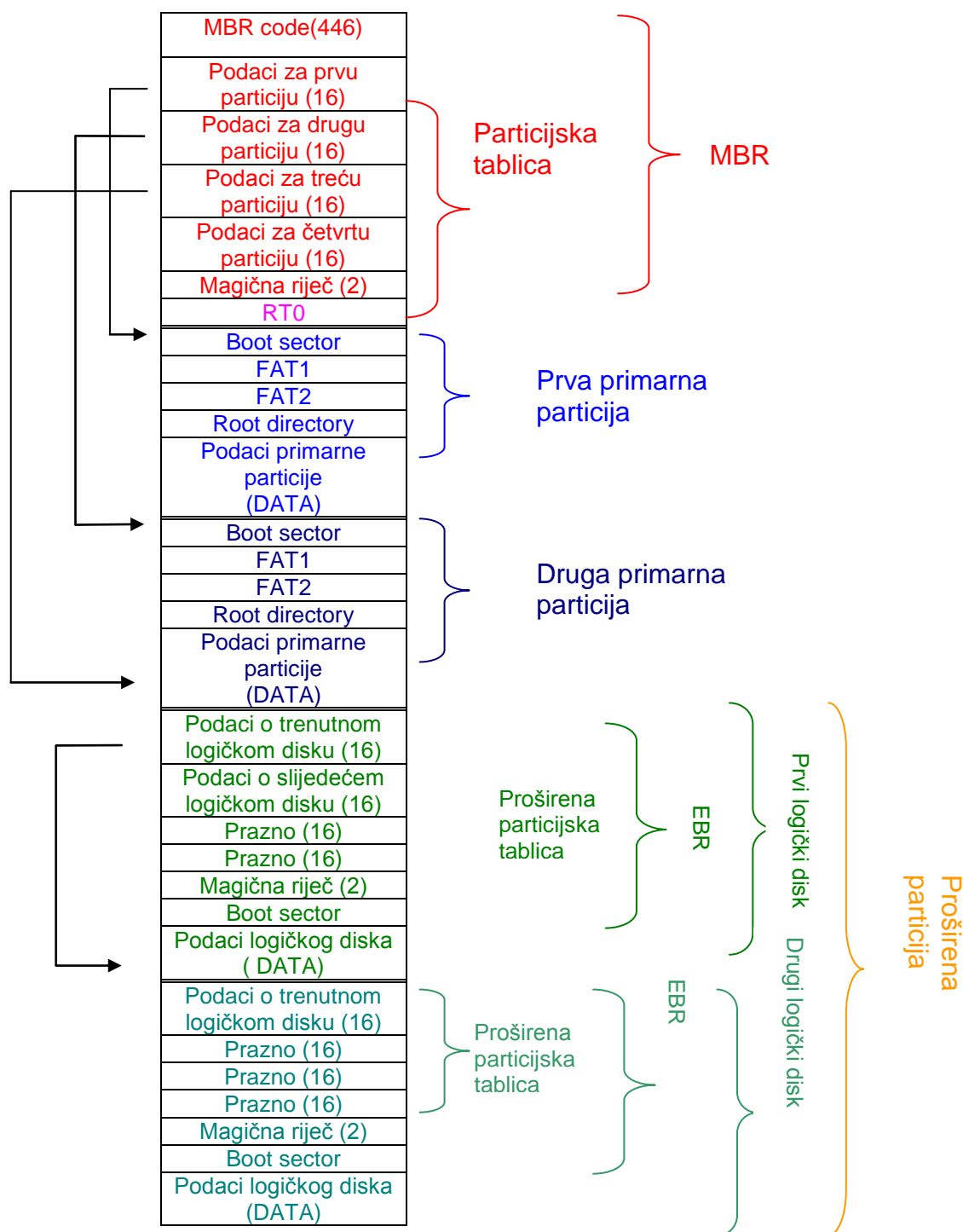
5.4. Manje poznati

Postoje još mnogi datotečni sustavi od kojih su neki nepoznati običnim korisnicima, a to su:

- a) Ext2 (Second Extended File System) * Linux
- b) Ext3 (Third Extended File System) * Linux
- c) ReiserFS (Reiser File System) * Linux
- d) UFS (Unix File System) * Unix
- e) FFS (FreeBSD File System) * FreeBSD
- f) HPFS (High Performance File System) * IBM OS/2, Win NT <v.3.51
- g) BFS (BeOS File System) * BeOS

6. Logička struktura

Logička struktura diska daje nam uvid u podjelu diska na logičke cjeline ili [particije](#), te prikazuje dijelove koji su neophodni za normalno funkcioniranje particije i podizanje sustava. Na slici imamo prikazanu strukturu diska sa dvije primarne particije i jednom proširenom sa dva logička diska



Objašnjenje: Na ovoj slici imamo logičku strukturu tvrdog diska sa dvije primarne i jednom proširenom particijom koja sadrži dva logička diska. Uz nazive pojedinih dijelova strukture u zagradi je upisana i veličina područja koje zauzima taj podatak.

7. MBR(Master Boot Record)

Master Boot Record je područje veličine 512 byte –a, koje se nalazi na početku prvog diska na CHS adresi 0,1,1. MBR se kreira pri formiranju particije na tvrdom disku programom FDISK. Master Boot Record je napravljen iz tog razloga kako bi se mogao učitati i podići operacijski sustav ili sustavi sa tvrdog diska ukoliko ih ima više. Master Boot Record sadrži tri glavna dijela, to su *MBR code*(*MBR kod*), *Partition table*(*particijska tablica*), i *Magic Word*(*Magična riječ*)

<u>MBR code</u> (446 bytes)
<u>Particijska tablica</u> (64 bytes)
<u>Magična riječ</u> (2 bytes)

- Ovo je jedan primjer MBR –a.

[illegible]

7.1. MBR code

Prvi dio MBR –a je *MBR code* dužine 446 byte –a. To je mali program u MBR –u koji se izvršava nakon BIOS –a. U prvom koraku tog programa on kopira sam sebe sa memoriskog prostora 7C00h -7DFFh (na koji ga je kopirao BIOS) na memoriski prostor 0600h-07FFh (tu se radi o adresama u RAM –u). U sljedećem koraku taj program provjerava [particijsku tablicu](#). Ukoliko u tablici primjeti nepravilnosti (npr. da ne sadrži bootabilnu particiju ili ukoliko sadrži više bootabilnih particija), tada će ispisati poruku o greški tipa

- Invalid partition table
- Error loading operating system
- Missing operating system

Ukoliko nema nepravilnosti program traži aktivnu particiju koju prepoznaje prema Boot indikatoru 80h i tada program nastavlja podizati operacijski sustav tako da se prebaci na boot sector aktivne particije. MBR code može biti veličine do 446 byte –a, dakle može biti i manje veličine. Postoji više vrsta MBR koda koji su različite veličine. Postoji način kreiranja MBR koda upisom naredbe *fdisk /mbr*. Ta naredba briše stari kod i piše novi, ali se ostali dijelovi MBR –a ne diraju. Ali postoji način da se obriše particijska tablica, tako da umjesto magične riječi (55 AA h) upišemo (00 00h) i upotrijebimo tu naredbu. Nakon toga program neće naći magičnu riječ i pobrisati će particijsku tablicu ali će ispisati ponovno magičnu riječ. U daljnjem primjeru biti će prikazan MBR kod napisan u Assembleru.

- Ovaj dio kopira sebe sa adrese 7C00h na adresu 0600h

```

7C00 FA          CLI          ; briši IF flag - onemogućiti prekide
7C01 33C0        XOR AX,AX    ; stavi stack segment na 0000
7C03 8ED0        MOV SS,AX
7C05 BC007C      MOV SP,7C00   ; postavi stack na adresu 7C00
7C08 8BF4        MOV SI,SP     ; kopiraj odavde 7C00 u SI
7C0A 50          PUSH         AX
7C0B 07          POP ES        ; ES je sada 7C00
7C0C 50          PUSH         AX
7C0D 1F          POP DS        ; DS je sada 7C00
7C0E FB          STI          ; omogući prekide
7C0F FC          CLD
7C10 BF0006      MOV DI,0600    ; DI je sada 0600
7C13 B90001      MOV CX,0100    ; prebaci 512 bajtova (256 words)
7C16 F2          REPNZ        ; prebaci MBR s adrese 7C00 na 0600
7C17 A5          MOVSW
7C18 EA1D060000  JMP 0000:061D  ; skok na adresu 061D

```

- Sljedeći korak je pronalaženje boot indikatora (80h), ukoliko je više particija bootabilno isisuje se poruka «Invalid partition table». A ukoliko nije niti jedna particija bootabilna pokreće se ROM Basic.

```

061D BEBE07      MOV SI,07BE    ; početak particijske tablice, zapis prve
                                ; particije
0620 B304        MOV BL,04      ; stavi brojač na maksimalno 4 zapisa (4
                                ; particije u tablici)
0622 803C80      CMP BYTE PTR [SI],80 ; provjera da li je boot_indicator=80h
                                ; particija aktivna?
0625 740E        JZ 0635        ; DA (ako je aktivna) skoči na adresu 0635
0627 803C00      CMP BYTE PTR [SI],00 ; ispitaj boot_indicator-može biti samo 00h or
                                ; 80h
062A 751C        JNZ 0648       ; ako nije 00h poruka -> "Invalid partition
                                ; table"
062C 83C610      ADD SI,+10     ; povećaj pointer za 16bajtova-sljedeći zapis
                                ; particije
062F FECB        DEC BL         ; smanji vrijednost brojača
0631 75EF        JNZ 0622       ; ukoliko nije kraj tablice vrati se na adresu
0622             ;
0633 CD18        INT 18         ; niti jedna od particija nije aktivna-pokreće
                                ; se ROM-BASIC

```

- U sljedećem dijelu još se uvijek provjerava [particijska tablica](#)

```

0635 8B14      MOV DX,[SI]          ; stavi u registar DH/DL broj drive/head za
                                INT19
0637 8B4C02     MOV CX,[SI+02]      ; stavi u registre CH/CL broj cylinder/sector
                                za INT19
063A 8BEE      MOV BP,SI           ; snimi pokazivač na poziciju aktivnog zapisa
063C 83C610     ADD SI,+10          ; sljedeći zapis u partijskoj tablici
063F FECB      DEC BL              ; da li je ovo posljednji zapis partijske
                                tablice?
0641 741A      JZ 065D             ; ukoliko je ovo kraj tablice skoči na 065D ->
                                boot
0643 803C00     CMP BYTE PTR [SI],00 ; nije bootabilna?
0646 74F4      JZ 063C             ; ukoliko nije nastavi dalje

```

- Ovaj dio služi za ispis pogreške i zatim odlazi u petlju nakon čega dolazi do ponovnog poretanja računala

```

0648 BE8B06     MOV SI,068B        ; poruka -> "Invalid partition table"
064B AC         LODSB              ; uzimaj po karakter poruke
064C 3C00       CMP AL,00          ; vidi da li se došlo do 00 - kraj poruke
064E 740B       JZ 065B            ; ako je kraj skoči na adresu 065B
0650 56         PUSH              SI ; pohrani SI
0651 BB0700     MOV BX,0007        ; atributi ekrana
0654 B40E       MOV AH,0E          ; ispiši 1 karakter poruke
0656 CD10       INT 10             ; ispisuje znak iz AL na ekran
0658 5E         POP SI             ; obnovi SI
0659 EBFO       JMP 064B           ; vrći petlju dok ne očita sve znakove poruke
065B EBFE       JMP 065B           ; beskonačna petlja - stoj i čekaj restart

```

- Sljedeći dio kopira Boot sector an adresu 7C00h. Ukoliko ta particija koja je bootabilna nije formatirana tj. nema boot sektora ispisuje se poruka «Error loading operating system».

```

065D BF0500     MOV DI,0005        ; pokušaj INT13h (INT19) 5 puta
0660 BB007C     MOV BX,7C00        ; učitaj boot secot na adresu 0000:7C00
0663 B80102     MOV AX,0201        ; pročitaj 1 sektor
0666 57         PUSH              DI ; pohrani DI
0667 CD13       INT 13             ; učitaj sektor na adresu 7C00
0669 5F         POP DI             ; obnovi DI
066A 730C       JNB 0678           ; ukoliko nije bilo problema s INT13h skoči na
                                0678
066C 33C0       XOR AX,AX          ; else
066E CD13       INT 13             ; izvrši prekid INT13 i resetiraj disk
0670 4F         DEC DI             ; umanji DI
0671 75ED       JNZ 0660           ; ukoliko DI nije 0 pokušaj ponovo
0673 BEA306     MOV SI,06A3        ; poruka -> "Error loading operating system"
0676 EBD3       JMP 064B           ; skoči na adresu 064B - petlja za prikazivanje
                                poruke

```

- Zadnji dio MBR programa je provjera [magične riječi](#) boot sektora, ukoliko ona nedostaje ispisuje se poruka «Missing operatig system»

```

0678 BEC206     MOV SI,06C2        ; nedostaje operativni sustav
067B BFFE7D     MOV DI,7DFE        ; stavi u DI pokazivač na magičnu riječ
067E 81D55AA    CMP WORD PTR [DI],AA55 ; provjera magične riječi
0682 75C7       JNZ 064B           ; poruka -> "Missing operating system"
0684 8BF5       MOV SI,BP          ; SI = BP = offset of active partition entry
0686 EA007C0000 JMP 0000:7C00      ; skok na boot sektor

```

7.2. Particijska tablica

Particijska tablica je dio MBR –a u kojem su zapisane sve informacije o particijama tvrdog diska. Budući da je taj dio ograničen na veličinu od 64 byte –a, dakle 16

byte –a po particiji, istovremeno možemo vidjeti samo četiri particije. Od te četiri particije jedna mora biti primarna kako bi mogla biti bootabilna, dok ostale mogu biti proširene. Možda ovo sa četiri particije predstavlja problem za onoga tko želi imati više particija npr. 8 particija. Općenito na tvrdom se disku može imati proizvoljan broj particija samo što one ne mogu biti istovremeno zapisane u particijskoj tablici. Dakle za onog tko želi imati npr. 8 particija može kombinirati 4 i 4 particije, ali i nemora. Može imati svaku particiju posebno, a u tom će slučaju dio tvrdog diska koji se ne nalazi u particijskoj tablici za neke programe biti prikazan kao slobodno mjesto na tvrdom disku (*free space*). Sljedeći primjer prikazuje particijsku tablicu sa dvije primarne particije.

Adresa:	Zapis:
0000001B0 :	00 00 00 00 00 00 00 00 – 00 00 00 00 00 00 80 01
0000001C0 :	01 00 0B FE BF A0 3F 00 – 00 00 22 F9 A4 00 00 00
0000001D0 :	C1 31 0C FE FF FF F1 45 – C8 00 66 E3 B5 00 00 00
0000001E0 :	00 00 00 00 00 00 00 00 – 00 00 00 00 00 00 00 00
0000001F0 :	00 00 00 00 00 00 00 00 – 00 00 00 00 00 00 55 AA

Na ovom primjeru je prikazana particijska tablica tvrdog diska sa dvije [primarne particije](#), prva je označena crvenom bojom dok je druga particija označena plavom bojom.

Zapis o svakoj particiji u particijskoj tablici sastoji se od više dijelova, ti dijelovi su na particijskoj tablici označeni posebnom bojom.

0000001B0 : 80 01
 0000001C0 : 01 00 0B FE BF A0 3F 00 - 00 00 22 F9 A4 00

U donjoj tablici pojašnjeni su svi dijelovi nekog zapisa o particiji u particijskoj tablici. Konkretno uzeta je prva particija koja je bootabilna počinje na CHS adresi 0,0,1, sa datotečnim sustavom FAT32, završava na CHS adresi 672,254,63. Od početka diska do particije ima 63 sektora, a ukupni broj sektora je 10811682.

Adresa polja	Dužina polja	Vrijednost polja	Opis polja
--------------	--------------	------------------	------------

01BEh	8 bita	80h (1000-0000)	Boot flag ili Boot indicator: Označava da li je neka particija aktivna., no pritom valja imati na umu da samo jedna particija može biti označena kao aktivna tj. bootabilna u particijskoj tablici. Moguće su samo dvije vrijednosti : 80h = aktivna particija 00h= nije aktivna
<i>Sljedeća tri polja predstavljaju početnu CHS adresu particije</i>			
01BFh	8 bita	01h (0000-0001)	Starting Head (početna glava): označava početnu adresu glave particije. Budući da je ovo polje duljine 8 bitova tj. 2 bytea možemo adresirati 2 ⁸ glava tj. 256 glava (0-255). U ovom slučaju to je glava 01.
01C0h	6 bita	01h (0000-0001)	Starting sector(početni sektor) : označava adresu početnog sektora particije. U ovi osam bitova koristi se samo 5 bitova niže težinske vrijednosti dok su preostala dva bita dio adrese cilindra. Pomoću ovih 6 bitova može se ukupno adresirati 2 ⁶ =64 sektora (0-63). U ovom slučaju to je prvi sektor (01)
01C1h	10 bita	00h 00 (00000000)	Starting Cylinder (početni cilindar): označava početnu adresu cilindra particije. Za ovu adresu koristi se ovaj byte i 2 bita iz prethodnog bytea. Kako se koristi 10 bitova moguće je adresirati 2 ¹⁰ =1024 cilindra (0-1023). Ovdje je adresiran prvi cilindar (01)
<i>Sljedeće polje predstavlja system flag</i>			
01C2h	8 bita	0Bh (0000-1011)	System flag (ID) : definira datotečni sustav . U ovom slučaju to je FAT 32
<i>Sljedeća tri polja predstavljaju završnu CHS adresu particije</i>			
01C3h	8 bita	FEh (1111-1110)	End Head (završna glava): označava završnu adresu glave particije. Kako je dužina ovog polja byte=8 bitova moguće je adresirati 2 ⁸ =256 glava (0-255). Ovdje je to glava FEh= 254
01C4h	6 bits	BFh (1011-1111)	Ending sector (završni sektor): označava završnu adresu sektora particije. U ovom byteu koristi se samo 5 bitova niže težinske vrijednosti dok su preostala dva dio adrese cilindra. Pomoću ovih 6 bitova može se ukupno adresirati 2 ⁶ =64 sektora (0-63). Ovdje je to sektor 63
01C5h	10 bits	A0h 10 (1010-0000)	End cylinder (završni cilindar): označava završnu adresu cilindra particije. Za ovu adresu koristi se ovaj byte i 2 bita iz prethodnog bytea. Kako se koristi 10 bitova moguće je adresirati 2 ¹⁰ =1024 cilindra (0-1023). Ovdje je adresiran cilindar 672
01C6h 01C9h	DWORD	3F000000h 0011-1111-00...	Relative Sectors ili Sectors offset: Broj sektora od početka diska do particije. To je adresa particije označena pomoću LBA adresiranja. Vrijednost ovog polja je 3Fh=63 što znači da se ispred particije nalaze 63 sektora.
Total Sectors : ukupni broj sektora koji ta particija			

01CAh 01CDh	DWORD	22F9A400h	zauzima. Ovdje je to 22F9A400h = 10811682 sektora.
----------------	-------	-----------	----------------------------------------------------

Objašnjenje veličine jedinica koje se koriste u tablici

- «BYTE» je 8 bitova
- «WORD» (riječ) : dužine 2 byte –a(16 bitova)
- «DWORD» (dupla riječ): dužine je 4 byte –a(32 bita)
- «LONGLONG» : dužine 8 byte –a(64 bita)

Napomena: Vrijednosti polja većeg od jedog byte –a odnosno osam bita zapisane su u «*Little endian*» formatu, tj oburnutim redoslijedom. Dakle prava vrijednost zapisa 22F9A400 zapisana heksadecimalno bila bi 00A4F922. To pravilo se ne odnosi na zapis kojim je zapisan početni i završni cilindar, jer se jednom byte –u dodaju 2 bita iz prethodnog zapisa. Što se tiče vrijednosti koje su navedene u tablici (početne i završne CHS adrese), one nisu standardizirane, dakle nisu na svakom tvrdom disku iste nego se vrijednosti dobiju po postupku koji je opisan u koloni «Opis polja».

7.3. Magična riječ

Magična riječ (Signature Word) je zapis veličine 2 byte –a i ona označava kraj Master Boot Record, Boot Record ili Extended Boot Record. Vrijednost tog zapisa je uvijek **55 AA** heksadecimalno i ona se ne mijenja bez obzira na promjenu particijske tablice ili promjenu koda.

Ukoliko MBR program pri podizanju sustava primjeti pogrešku u magičnom broju, tada će se podizanje zaustaviti i na monitoru će biti ispisana pogreška. Ovisno o tipu BIOS –a pogreške koje se ispisuju mogu biti tipa:

Boot from CD:Disk Boot Failure

1962 Boot Sequence Error

Disk Boot Failure, Insert System Disk and Press ENTER

Svaki tvrdi disk ima samo jedan MBR, koji se nalazi na početku diska. Disketa nema MBR nego je njezin prvi sektor boot sektor.

Veličina Master Boot Record se ne može mijenjati

Dodatak:

Virusi su programi koji su napravljeni kako bi se pripojili računalu i ometali računalu ili korisnika u radu. Virus se često izvode i šire bez korisnikova znanja. Budući da je MBR česta meta napada virusa postoji rješenje da se virus makne iz MBR –a, a da pritom podaci koje se nalaze na tvrdom disku i općenito u MBR –u ne budu ni taknuti.

Virus koji se smješta u MBR smješta se u mali dio između MBR koda i particijske tablice.

000000C0:	6D 00 4D 69 73 73 69 6E - 67 20 6F 70 65 72 61 74	m.Missing operat
000000D0:	69 6E 67 20 73 79 73 74 - 65 6D 00 00 80 28 47 15	ing system..C65
000000E0:	00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00
000000F0:	00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00
00000100:	00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00
00000110:	00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00
00000120:	00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00
00000130:	00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00
00000140:	00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00
00000150:	00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00
00000160:	00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00
00000170:	00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00
00000180:	00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00
00000190:	00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00
000001A0:	00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00
000001B0:	00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 80 01
000001C0:	01 00 0B FE BF A0 3F 00 - 00 00 22 F9 A4 00 00 00
000001D0:	C1 31 0C FE FF FF F1 45 - C8 00 66 E3 85 00 00 00
000001E0:	00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00

Prazni prostor u MBR –u

Jedna metoda suzbijanja virusa je da u taj dio pomoću programa **Tbutil** stavimo imunizaciju tj. stavimo program za provjeru MBR –a pri svakom podizanju sistema. Taj program ostaje trajno tako dugo dok ne probamo drugu metodu.

Druga metoda je upisivanjem u prompt naredbe **fdisk/mbr**. Tom naredbom kreiramo novi MBR kod i tako brišemo virus iz MBR –a.

U svakom slučaju prva metoda je bolja ako je podržava operacijski sustav s kojim radimo, dok je druga metoda jednostavno «za jednokratnu upotrebu».

8. RT0 (Reminder of Track 0)

Ostatak traga nula je kod većine računala prazan, ali on može služiti kaopodručje u koje možemo staviti kopiju nekih važnih podataka. Ukoliko prva particija počinje na CHS adresi 0,1,1 tada je RT0 veličine 62 sektora. U ostatak traga nula možemo kopirati npr. Master Boot Record ili samo partijsku tablicu ili boot sektor neke particije. Tako da u slučaju da ostanemo bez partijske tablice a nemamo systemske informacije u kratkom roku možemo vratiti računalo u početno stanje.

9. Particije

Particije su logičke strukture odnosno cjeline tvrdog diska, koje operacijski sustav prikazuje kao zasebne prostore na disku. Ukoliko uzmemo [logičku strukturu](#) prethodnog primjera gdje imamo dvije primarne particije i jednu proširenu sa dva logička diska, operacijski sustavi će prikazivati četiri cjeline na tvrdom disku.

Particije mogu biti primarne (primary) i proširene (extended). Primarne particije su najčešće one na kojima se nalazi operacijski sustav i sa kojih se podiže sistem tj. znači da su bootabilne. Proširene particije najčešće služe za pohranu podataka koji nisu važni za podizanje sustava jer one ne mogu biti bootabilne. Proširena particija zauzima također prostor u partijskoj tablici kao i primarna. Postoji ograničenje na maksimalno [četiri particije](#) koje istovremeno mogu biti vidljive.

Tvrđi disk se može particionirati zbog više razloga. U prijašnje vrijeme to je bilo zbog datotečnog sustava kojeg koristi operacijski sustav DOS 6.22. Naime datotečni sustav [FAT16](#) nije podržavao veličine diska ili particije veće ili jednake kapacitetu od 2 GB. Prema tome tko je imao tvrdi disk veličine 4 GB morao ga je podijeliti na najmanje dvije particije. U današnje vrijeme najčešće se koristi više particija zbog toga što mnogi korisnici imaju više različitih operacijskih sustava. Mnogi od tih operacijskih sustava rade samo sa svojim datotečnim sustavom, pa je potrebno disk podijeliti na više cjelina sa različitim datotečnim sustavima.

Ponekad se koristi više particija zbog čuvanja nekih dokumenata ili npr. na posebnoj particiji možemo imati IMAGE diska.

Ako se koristi proširena particija u nju možemo smjesiti logičke diskove, kojih može biti maksimalno 24 ako se ne koristi primarna particija. Ako koristimo primarnu particiju tada je ograničenje na maksimalno 23 logička diska unutar proširene particije. Ograničenje se ne nalazi ni u kapacitetu diska, ni u particijskoj tablici nego u označavanju tih diskova.

Označavanje diskova počinje sa **C:**, a završava sa **Z:**, prema tome najveći broj logičkih diskova koje možemo posjedovati je 24.

9.1. Primarna particija

Primarna particija, kao što je već spomenuto služi da se podiže sustav s nje. Naravno i na nju kao i na proširenu možemo staviti podatke nevažne za sustav, ali u svakom slučaju želimo li podići sustav sa tvrdog diska moramo imati primarnu particiju koja je bootabilna. Primarna particija se sastoji od četiri djela, to su:

Boot sector
FAT1
FAT2
Root Directory
Data Area

Prvi dio primarne particije je **Boot sektor**, iza njega dolaze dva ista podatka **FAT1** i **FAT2**, te na kraju imamo **Područje s podacima**.

Ovakva struktura particije dobiva se formatiranjem particije, kada je ona formirana. Kako bi program **format.com** znao koji je tip particije on treba pogledati particijsku tablicu (točnije adresu: 01x2, gdje x može imati vrijednost od C – F), i potražiti tip particije (system flag – ID). Kada program pronađe *system flag* i odredi vrsu particije, tada čita početne i završne adrese te particije i kreira datotečni sustav. Ukoliko naiđe na system flag koji označava primarnu particiju kreira ovakvu strukturu. Ukoliko tvrdi disk sadrži primarnu particiju i proširenu tada će program pregledavati logičke diskove unutar proširene particije. Kada dođe do logičkog diska koji ima zapis o sljedećem logičkom disku, tada odlazi na drugi logički disk i na njemu formira strukturu.

Kada se jednom formira ovakva struktura njezinu je veličinu moguće jedino mijenjati programima kao što su npr. Partition Magic, u kojem je moguće mijenjati veličinu particije, a proporcionalno sa veličinom particije djelujemo na veličinu područja FAT1 i FAT2, te na područje s podacima dok Boot sektor ostaje iste veličine.

9.1.1 *Boot Sector primarne particije*

Boot sector je prvi sektor primarne particije i on je veličine 512 byte –a. Njegov sadržaj definira i pomaže podizanje sustava.

```
Physical Sector: Cyl 878, Side 0, Sector 1
_OFFSET_ 0 1 2 3 4 5 6 7 8 9 A B C D E F
00000000: EB 58 90 4D 53 44 4F 53 35 2E 30 00 02 08 20 00 .X.MSDOS5.0...
00000010: 02 00 00 00 00 F8 00 00 3F 00 FF 00 EE 39 D7 00 .....?....9..
00000020: 7F 32 4E 00 83 13 00 00 00 00 00 00 02 00 00 00 2N.....
00000030: 01 00 06 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000040: 80 00 29 8B 93 6D 54 4E 4F 20 4E 41 4D 45 20 00 ..).mTNO NAME
```

```

00000050: 20 20 46 41 54 33 32 20 20 20 33 C9 8E D1 BC F4 FAT32 3....
00000060: 7B 8E C1 8E D9 BD 00 7C 88 4E 02 8A 56 40 B4 08 {...}.N..V@..
00000070: CD 13 73 05 B9 FF FF 8A F1 66 0F B6 C6 40 66 0F ..s.....f...@f.
00000080: B6 D1 80 E2 3F F7 E2 86 CD C0 ED 06 41 66 0F B7 ....?.....Af..
00000090: C9 66 F7 E1 66 89 46 F8 83 7E 16 00 75 38 83 7E .f.f.F.~..u8.~
000000A0: 2A 00 77 32 66 8B 46 1C 66 83 C0 0C BB 00 80 B9 *.w2f.F.f.....
000000B0: 01 00 E8 2B 00 E9 48 03 A0 FA 7D B4 7D 8B F0 AC ...+.H...}.}...
000000C0: 84 C0 74 17 3C FF 74 09 B4 0E BB 07 00 CD 10 EB ..t.<.t.....
000000D0: EE A0 FB 7D EB E5 A0 F9 7D EB E0 98 CD 16 CD 19 ...}....}.....
000000E0: 66 60 66 3B 46 F8 0F 82 4A 00 66 6A 00 66 50 06 f f;F...J.fj.fP.
000000F0: 53 66 68 10 00 01 00 80 7E 02 00 0F 85 20 00 B4 Sfh.....~.....
00000100: 41 BB AA 55 8A 56 40 CD 13 0F 82 1C 00 81 FB 55 A..U.V@.....U
00000110: AA 0F 85 14 00 F6 C1 01 0F 84 0D 00 FE 46 02 B4 .....F..
00000120: 42 8A 56 40 8B F4 CD 13 B0 F9 66 58 66 58 66 58 B.V@.....fXfXfX
00000130: 66 58 EB 2A 66 33 D2 66 0F B7 4E 18 66 F7 F1 FE fX.*f3.f.N.f...
00000140: C2 8A CA 66 8B D0 66 C1 EA 10 F7 76 1A 86 D6 8A ...f.f....v....
00000150: 56 40 8A E8 C0 E4 06 0A CC B8 01 02 CD 13 66 61 V@.....fa
00000160: 0F 82 54 FF 81 C3 00 02 66 40 49 0F 85 71 FF C3 ..T.....f@l.q..
00000170: 4E 54 4C 44 52 20 20 20 20 20 20 0D 0A 4E 54 4C NTLDR ..NTL
00000180: 44 52 20 69 73 20 6D 69 73 73 69 6E 67 FF 0D 0A DR is missing...
00000190: 44 69 73 6B 20 65 72 72 6F 72 FF 0D 0A 50 72 65 Disk error...Pre
000001A0: 73 73 20 61 6E 79 20 6B 65 79 20 74 6F 20 72 65 ss any key to re
000001B0: 73 74 61 72 74 0D 0A 00 00 00 00 00 00 00 00 00 start.....
000001C0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
000001D0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
000001E0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
000001F0: 00 00 00 00 00 00 00 00 00 7B 8E 9B 00 00 55 AA .....{....U.

```

9.1.1.1. Tablica na drugoj strani detaljno opisuje ovaj Boot sektor.

Pomak polja	Dužina polja	Vrijednost polja	Ime i značenje polja
00h	3 bytes	EB 58 90h	Jump Instruction (JMP) : Asemblerska instrukcija skoka na Boot code, koji se nalazi odmah iza BPB –a, tako da je moguće mijenjati BPB ovisno o datotečnom sustavu.
03h	8 bytes	4D53444F 53352E30h	OEM ID označava operacijski sustav kojim je particija formatirana. I MS-DOS 5.0 i Windowsi imaju na FAT particiji oznaku MSDOS5.0 radi kompatibilnosti
<i>BIOS Parameters Block polja:</i>			
0Bh	WORD	0002h	Bytes Per Sector : označava od koliko se byteova sastoji jedan sektor. Moguće su vrijednosti : 512 , 1024 , 2048 , i 4096 byta ali gotovo svi tvrdi diskovi koriste 512 bytea po sektoru.
0Dh	BYTE	08h	Sectors Per Cluster : označava koliko sektora ide u jedan cluster tj. veličinu clustera.Moguće veličine clustera su : 1 , 2 , 4 , 8 , 16 , 32 , 64 i 128 sektora. FAT32 datotečni sustav može adresirati najviše 2^{32} = 4294967296 clustera. Vrijednosti klastera većega od 32KB=(Bytes Per Sector)*(Sector Per Cluster) mogu uzrokovati probleme u radu diska ili softverske pogreške.
0Eh	WORD	20 00h (0020h)	Reserved Sectors : označava broj rezerviranih sektora , tj broj sektora ispred prve FAT tablice uljučujući i boot sektor. Vrijednost ovog polja je 32
10h	BYTE	02h	Number of FATs : Broj FAT tablica na

			particiji. Vrijednost ovog polja uvijek je 2.
11h	WORD	00 00h	Root Entries : Koristi samo kod FAT12 i FAT16 particija i ono označava maksimalni broj stavki (datoteka i direktorija) u root direktoriju. Kod FAT32 particija, ovo polje mora biti 0.
13h	WORD	00 00h	Small Sectors : I ovo se polje koristi samo kod FAT12/FAT16 particija. Kod FAT32 particija, ovo polje mora biti 0.
15h	BYTE	F8h	Media Descriptor : označava medij koji se koristi. Vrijednost F8h označava tvrdi disk, a F0h označava HD 3.5" disketnu jedinicu. Te vrijednosti specifične su za FAT16 sustav i ne koriste se kod Windowsa 2000.
16h	WORD	00 00 h	Sectors Per FAT : Ovo se polje koristi samo kod FAT12/FAT16 particija. Kod FAT32 particija, ovo polje mora biti 0.
1Ah	word	FF00h =00FFh= 255 glava	Number of Heads : Broj glava. Ova vrijednost je prividna
1Ch	dword	EE39D700h	Hidden Sectors : Broj skrivenih sektora
18h	WORD	3F 00 h = 003Fh = 63 sektora po stazi	Sectors Per Track : Ova vrijednost je dio prividne (translatirane) geometrije diska koja se koristila kada je disk bio formatiran na niskoj razini (<i>low-level formatted disk</i>). Sadrži geometriju (broj sektora po stazi) za diskove koji koriste stari INT13h prekid.
20h	DWORD	7F324E00h	Large Sectors: Ukupan broj sektora FAT32 particije.

Napomena: Donja tablica odnosi se samo na FAT32 datotečni sustav.

24h	DWORD	83130000h =00001383h= 4995 sektora	Sectors Per FAT : Broj sektora koji čine FAT tablicu na particiji. Operacijski sustav koristi ovu vrijednost, broj FAT tablica na particiji i broj skrivenih sektora (<i>Hidden Sectors</i>), kako bi se odredilo gdje počinje korijenski direktorij. OS također može prema broju zapisa u korijenskom direktoriju odrediti gdje na disku počinje područje na kojem su zapisani korisnikovi podaci.
28h	word	0000h	Extended Flags : Značenje bitova : bitovi 0–3: broj aktivne FAT tablice (brojanje počinje od 0). To je valjano ukoliko je isključeno preslikavanje (<i>mirroring</i>); bitovi 4–6: rezervirano; bit 7: 0 znači FAT je preslikan za vrijeme rada u sve FAT tablice, 1 znači da je jedan FAT aktivan (definirano bitovima 0-3); bitovi 8–15: rezervirano.
2Ah	word	0000h	File System Version : Viši bajt označava glavni broj revizije, a niži bajt označava podverziju. Ovo polje omogućuje u budućnosti razvoj i proširivanje mogućnosti FAT32 sustava i razlikovanje starih verzija FAT32 na disku.
2Ch	dword	02000000h	Root Cluster Number : Broj prvog klastera korijenskog direktorija (root directory). Ova je vrijednost obično 2 , ali ne mora biti uvijek.
			File System Information Sector Number :

30h	word	0100h	Broj sektora <i>File System Information</i> (FSINFO) strukture na rezerviranom dijelu FAT32 particije. Ova vrijednost je tipično 1. Kopija FSINFO strukture drži se u <i>Backup Boot Sector</i> -u, ali nije redovno ažurna.
32h	word	0600h	Backup Boot Sector : Označava broj sektora u rezerviranom dijelu particije na kojem je pohranjena kopija <i>boot sector</i> . Tipična vrijednost ovog polja je 6 i ne preporučuje se druga vrijednost

Extended BIOS Parameters Block polja FAT32 particije			
pomak	dužina	vrijednost	ime polja i opis
40h	BYTE	80h 00h	(<i>Physical Drive Number</i>). Odnosi se na oznaku diska koju koristi BIOS. Disketna jedinica A: identificirana je brojem 00h, a prvi tvrdi disk C: brojem 80h. Neovisno o tome koliko ima diskova. Tipično se ova vrijednost postavlja prije nego se izvrši prekid INT13h čime se označava BIOS-u kojem se disku želi pristupiti odnosno s kojega se diska želi podići sistem. Ova vrijednost ima ulogu jedino ako je disk bootabilna.
41h	BYTE	00h	(<i>Reserved</i>). Za FAT32 particije ovo polje je uvijek 0.
42h	BYTE	28h 29h	(<i>Extended Boot Signature</i>). Polje koje mora imati vrijednost 28h ili 29h kako bi ga Windows NT/2000 prepoznao.
43h	DWORD	A88B3652h	(<i>Volume Serial Number</i>). Slučajan serijski broj kreiran tijekom formatiranja diska koji omogućuje razlikovanje diskova.
47h	11 bytes	NO NAME	(<i>Volume Label</i>). Polje koje se koristi za pohranjivanje imena particije (<i>volume label</i>). No, sada se ime particije snima kao specijalna datoteka u korijenskom direktoriju.
52h	LONGLONG	FAT32	(<i>System ID</i>). Polje s vrijednošću "FAT32" čime se označava da je to FAT32 particija.

9.1.2. FAT1 i FAT2

Tvrdi disk je podijeljen na manje dijelove (512 byte sectore) u koje se pospremaju podaci. Ali zasebi se sektori ne organiziraju u datoteke zbog performansnih razloga, pa je prema tome napravljena logička podjela. Tako je disk podijeljen na veće komade – Clustere. Klasteri (clusters) su veličine od 512 byte –a do 64 Kbyte –a.

FAT je zapravo mjesto gdje se pospremaju podaci koji se klaster veže s kojim da bi stvorio datoteku. Kao što je vidljivo iz prijašnjeg prikaza strukture Boot Recorda vidimo da imamo FAT1 i FAT2 koji se nalaze iza boot sektora. FAT2 je zapravo kopija FAT1. U slučaju da se ošteti dio diska moguće je lako izgubiti podatke koji su na disku, u tom slučaju pomaže nam kopija FAT2.

9.1.3. Root Directory

Korjenski direktorij ili root je direktorij koji definira stablo direktorija na disku. Na svakoj particiji je moguće imati samo jedan korjenski direktorij na kojega se svi ostali dodaju. Za razliku od svih ostalih direktorija na disku on slijedi posebna pravila koja ne vrijede za regularne direktorije. Uz to on se nalazi na početku particije točno ispod dvije kopije FAT –a.

To je vrlo važan detalj prema kojem se on razlikuje od drugih direktorija. Broj stavaka koje se nalaze u korjenskom direktoriju ovisi o tipu particije. Diskete mogu imati 224 stavke u root – u. Ukupan broj stavki određuje se iz **BPB** –a. Novije verzije FAT32 sustava nemaju ograničenje o postavljanju i veličini korjenskog direktorija. To poboljšanje donosi mogućnosti da korjenski direktorij možemo proširivati i premješati kao i ostale direktorije. Jedna stvar je ista: **Korjenski direktorij se ne može izbrisati niti postoji direktorij koji je iznad njega.**

Sljedeća je tablica prikaz strukture svakog ulaza.

Pomak	Sadržaj
00h	Ime datoteke
08h	Ekstenzija datoteke
0Bh	Atribut datoteke
0Ch	Rezervirano
16h	Vrijeme zadnje promjene datoteke
18h	Datum zadnje promjene datoteke
1Ah	Prvi klaster datoteke
1Ch	Dužina datoteke

Ime datoteke je klasično DOS zapisivanje imena od osam znakova.

Ekstenzija nemora postojati, ali ako postoji ona je dužine od 1-3 znaka, ekstenzija ima važan utjecaj u radu. Naime ona govori što se može raditi sa podatkom koji ima određenu ekstenziju, tj. podatak sa ekstenzijom «TXT» možemo otvoriti u bilo kojem editoru i izmijeniti sadržaj. Programi imaju ekstenzije «EXE». Kada upišemo ime programa sa tim nastavkom on se pokreće, dakle ekstenzija može na neki način prikazati tip datoteke.

Atributi datoteka koriste se kako bi prikazali karakteristike određenih datoteka. Npr. atribut «Hidden» prikazuje da je datoteka skrivena i nemožemo je vidjeti u ispisu direktorija ili datoteka. Datoteka može imati više od jednog atributa koji će je karakterizirati.

9.1.4. Data Area

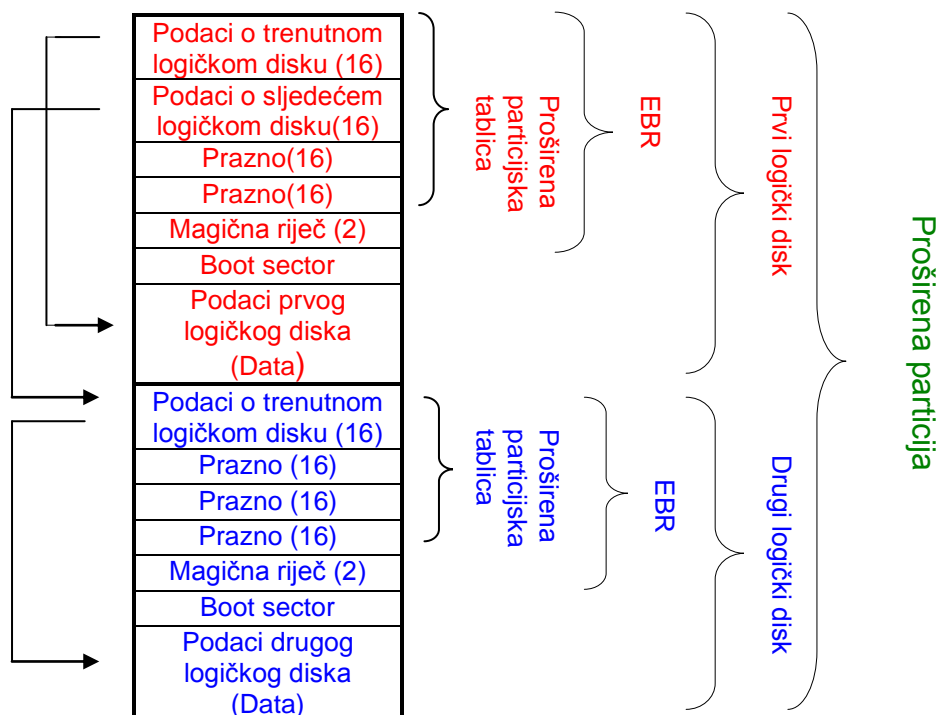
Data Area ili područje sa podacima kao što samo ime govori. Svaka particija ima svoje područje u koje sprema podatke. Veličina tog područja određena je veličinom particije.

9.2. Proširena particija

Proširena particija se u mnogočemu razlikuje od primarne. U prvom slučaju to je da ne može biti bootabilna. Sljedeće svojstvo je da u proširene particije stavljamo logičke diskove. Logički diskovi su također prikazani kao odvojene particije sa svojim prostorom na disku. Jedino što ograničava broj logičkih diskova jest označavanje (zbog slova od C :– Z:). Logički diskovi na svom početku imaju *extended boot record* koji sadrži adresu sljedećeg logičkog diska (ako postoji), te podatke o geometriji tvrdog diska. Na taj način logički su diskovi povezani u jednu cjelinu.

9.2.1. EBR(Extended Boot Record)

EBR se sastoji od proširene partijske tablice i oznake kraja zapisa (signature word). Svi EBR –ovi su povezani u lanac.



Na ovoj je slici prikazana struktura proširene particije sa dva logička diska. Brojevi koji se nalaze u zagradama označavaju veličinu koju to područje zauzima. Kao što je vidljivo iz slike to je da su podaci o diskovima povezani u jedan lanac. Zapisi unutar partijske tablica proširene particije su jednaki zapisima partijske tablice u MBR –u.

10. Clusteri i alokacijska datoteka

Jedna od najvažnijih stvari koje operacijski sustav čini je sposobnost upravljanja diskom i drugim medijima za pohranu podataka. Različiti operacijski sustavi različito upravljaju organizacijom i alokacijom diskovnog prostora za datoteke. Kako je klaster najmanja jedinica diskovnog prostora koji se može adresirati, svaka datoteka koja se zapisuje može se zapisati na određeni cijeli broj klastera. Ukoliko je datoteka manja od veličine klastera, ona zauzima cijeli klaster i ostatak prostora u klasteru ne može se iskoristiti. Veličina clustera može biti od 512 byte –a do 64 Kbyte –a, ovisno o tipu i veličini particije.

Veličina particije	broj sektora po klasteru	Veličina klastera
0MB - 128MB	4	2KB
128MB - 256MB	8	4KB
256MB - 512MB	16	8KB
512MB - 1GB	32	16KB
1GB - 2GB	64	32KB
(2GB - 4GB)	(128)	(64KB)

Veličina klastera se određuje kod particioniranja diska i ima važan utjecaj na performanse i korisnost diska. U nekim slučajevima postoje programi (npr. Partition Magic) koji mogu promijeniti veličinu klastera već na gotovoj particiji.

Svaka datoteka mora biti alocirana sa cjelobrojnomo vrijednošću klastera. Na primjer : Neka su klasteri na disku veličine 4 kB (4096 byte –a), a neka datoteka neka je veličine 4000 byte –a, tada će ta datoteka zauzeti jedan klaster. Ali ako imamo datoteku od 5000 byte –a tada ona mora biti zapisana u dva klastera. Prvi će klaster biti popunjen dok će drugi biti gotovo prazan (4 Byte –a popunjeno), te će imati veliku neupotrebljivost diska. Kada bi koristili klastere veličine 1 kByte, tada bi prva četiri klastera bila popunjena dok bi peti klaster bio djelomično popunjen. U tom slučaju nemamo mnogo gubitka na tvrdom disku, ali disk zbog toga mora pročitati više informacija o klasterima te time gubimo na vremenu.

FAT se koristi za čuvanje informacija o tome koji klasteri su dodijeljeni pojedinim datotekama, pa tako operacijski sustavi mogu ustvrditi gdje su podaci iz datoteke smješteni korištenjem stavki u direktoriju i FAT –u. FAT također sadrži informacije o tome koji su klasteri popunjeni, a koji su još slobodni i mogu se koristiti. Kada neki program želi povećati veličinu neke datoteke on zatraži to od operacijskog sustava koji pogleda u FAT. Kad u FAT –u pronade slobodne klastere on ih dodijeli aplikaciji koja je to zatražila. Svaki klaster koji se koristi u FAT –u ima informaciju o sljedećem klasteru sa kojim se treba povezati. Posljednji klaster kojega neki podatak zauzima završava sa FFFFh, ta vrijednost govori operacijskom sustavu da je to posljednji klaster i da je to kraj podatka.

Svaki klaster se označava nekim kodom npr. kod «0» označava da je taj klaster neispunjen odnosno prazan. Zatim postoji kod koji označava klastere kao nepouz dane za daljnji rad, ti se klasteri označavaju kao *bad*.

Pristup datoteci kao cjelini ostvaren je pomoću gore navedenog primjera, tj. svi klasteri nekog podatka povezani su u lanac. Klasteri nekog podatka ne moraju biti jedan iza drugog, oni se mogu nalaziti po cijelome disku. Ali pri tome nastaje problem, jer tvrdi disk mora potrošiti više vremena da bi se prikupili sve informacije o nekom podatku. Zbog toga se vrši defragmentacija.

Jedna od mnogih prednosti datotečnog sustava FAT je lakoća kojom možemo u kratkom djelovanju, brzo vratiti izbrisane podatke sa tvrdog diska. Dakle odlučimo «trajno» izbrisati neki podatak. On će i nakon tog našeg brisanja ostati na tvrdom disku, a sustav će takvim podacima prvi znak imena podatka zamijeniti sa E5h i time označiti da je podatak «izbrisan». Taj podatak će stvarno biti pobrisan sa tvrdog diska kada ga sa drugim podacima pregazimo. Dakle ukoliko djelujemo u kratkom vremenu dok još nismo ništa novo spremali na tvrdi disk lako je moguće da sa nekim programima poput DOS –ovog UNDELETE –a ili Norton Utilities UNERASE –a vratimo pobrisani podatak.

11. Defragmentacija

Budući da su podaci na tvrdom disku povezani lancem klastera, ti klasteri se mogu nalaziti bilo gdje na disku jer prethodni klaster sadrži adresu sljedećeg klastera. Tako je moguće da datoteka bude po čitavom tvrdom disku, i tu nastaje problem. To i nije problem nego veliki gubitak vremena jer sustav mora pomicati glave za čitanje i pisanje po čitavom tvrdom disku kako bi pronašao jednu datoteku, a to jedaleko od željene situacije. Takve probleme ne možemo izbjeći, ali ih možemo s vremena na vrijeme otkloniti. Najbolje rješenje bi bilo kada bi se određena datoteka u cijelosti nalazila na istom mjestu, dakle klasteri koji su povezani u jednu datoteku trebali bi se nalaziti jedan iza drugog. Tako bi tvrdi disk obavio u relativno malom vremenu doista velik posao. Za rješavanje tih problema postoje mnogi programi, najpoznatiji je Disk Defragmenter koji dolazi ugrađen sa Windows –ima 95/98... , a sama procedura naziva se defragmentiranje.

Objasnimo sada defragmentiranje na temelju sljedećeg primjera:

Cluster1	Cluster2	Cluster3	Cluster4	Cluster5	Cluster6
Cluster7	Cluster8	Cluster9	Cluster10	Cluster11	Cluster12

Neka je gornja slika prikaz tvrdog diska koji sadrži 12 clustera.

Ukoliko na tom disku stvorimo četiri datoteke:

A –veličine 1 klaster

B –veličine 4 klastera

C –veličine 2 klastera

D –veličine 3 klastera

A	B	B	B	B	C
C	D	D	D		

Na ovakvoj strukturi odlučimo izbrisati datoteku C koja nije potrebna, nakon toga slika diska je sljedeća

A	B	B	B	B	
	D	D	D		

Odlučimo sada da nam je potrebna datoteka E koja zauzima tri klastera. Budući da na disku nema niza blokova od tri klastera datoteka E će se razlomiti. Prvo će popuniti mjesta gdje je bila datoteka C, a zatim tražiti sljedeći slobodni klaster. Tvrdi će disk nakon toga izgledati ovako:

A	B	B	B	B	E
E	D	D	D	E	

Sada ćemo obrisati datoteku A i D te napraviti novu datoteku F koja će zauzimati 5 klastera. Tvrdi disk izgleda ovako:

F	B	B	B	B	E
E	F	F	F	E	F

U ovakvom stanju potraga za klasterima na kojim je zapisan neki podatak stvarno je gubljenje vremena. Želimo li podatke na tvrdom disku tako posložiti da su u neprekinutom nizu moramo pokrenuti neki program za defragmentaciju. Kada se ovaj disk defragmentira trebao bi izgledati ovako:

B	B	B	B	E	E
E	D	D	D	D	D

Postoje programi u DOS –u koji defragmentiraju disk. Sa tim programima ne preporuča se defragmentirati disk ako na njemu postoje Windows –i ili bilo koji operacijski sustav koji koristi

duga imena. U protivnom kada ti programi defragmentiraju disk oni prodacima dodaju klasična imena (8+3). U tom slučaju ne bismo imali direktorij Program Files nego Progra~1 što operacijski sustav Windows ne bi mogao prepoznati kao Program Files.

12. PC – Boot proces

Boot proces ili podizanje sustava (System Boot Sequence) je niz koraka koje sustav odrađuje nakon uključenja napajanja računala ili nakon ponovnog pokretanja (reset). Boot proces će ovdje biti prikazan u koraka.

Koraci:

1. Uključenje napajanja – Nakon uključenja napajanja potrebno je neko vrijeme kako bi se napon stabilizirao, nakon što je napon stabiliziran chipset prestaje generirati signale koji odlaze na procesor i on je spreman za rad. Naime ukoliko napon nije postavljen na vrijednost koju procesor zahtijeva za rad bez oštećenja, može doći do vrlo ozbiljnih posljedica za procesor neovisno o tome da li je napon prenizak ili previsok.
2. Budući da je RAM memorija prazna procesor negdje mora početi sa izvršavanjem instrukcija. Procesor prvo odlazi na adresu FFF0h na kojoj se nalazi instrukcija za skok na stvarni početak BIOS –a
3. BIOS izvodi POST (*Power On Self Test*) tj. testira kompletni hardware. Ukoliko naiđe na pogrešku zaustaviti će se, i budući da grafička kartica nije inicijalizirana, prijaviti će pogrešku zvučnim signalima.
4. BIOS traži BIOS grafičke kartice i pokreće ga. On se najčešće nalazi na memorijskoj lokaciji C000h. Kada se pokrene BIOS grafičke kartice on inicijalizira grafičku karticu te na vrhu ekrana ispisuje podatke o BIOS –u te kartice.
5. BIOS potražuje druge kartice, te pronalazi njihove ROM –ove. Najčešće se BIOS tvrdog diska nalazi na adresi C800h.
6. Izvođenju testova, prebrojavanje memorije, ispis količine memorije na ekranu. Ukoliko BIOS pronađe pogrešku o je ispisuje.
7. BIOS pregledava hardware koji je priključen na računalao. Današnji BIOS –i mogu dinamički odrediti parametre tvrdog diska CD–ROM –a, CD snimača...BIOS također traži i serijske i paralelne portove.
8. Ukoliko BIOS podržava P'n'P (Plug and Play) standard, konfigurirati će P'n'P uređaje i za svaki će uređaj ispisati poruku.
9. Ispisuje se tablica sa popisom svih komponenata priključenih na računalo. U njoj se nalaze adrese gdje se nalaze upravljački programi za te komponente i broj linije za prekid (*IRQ*). Ta tablica pomaže pri krivo postavljenim parametrima u BIOS Setup –u. U njoj su navedene sve.
10. BIOS traži pogon sa kojeg se podiže sustav. Današnji BIOS –i mogu sustav podići sa CD –a ili ZIP drive –a. Parametri Boot sekvence se podešavaju u CMOS –u. Kada procesor pronađe pogon sa kojeg se podiže operacijski sustav tada počinje rad programa pod nazivom Bootstrap.
11. Bootstrap je program za traženje [MBR](#) –a na disku. Kada ga pronađe tada slijedi pokretanje programa u MBR –u, točnije [MBR koda](#) .
12. Program u MBR –u prvo pogleda postoji li magični broj 55h AAh, ako postoji tada se vraća na početak partijske tablice.

13. U partijskoj tablici potražuje se aktivna particija tj. ona koja sadrži boot indikator ili boot flag (80h).
14. Kada program pronađe aktivnu particiju on odlazi na početni sektor (boot sektor) te particije.
15. Program u boot sektoru se pokreće te taj program podiže operacijski sustav.
16. Prvi korak u podizanju operacijskog sustava je učitavanje jezgre operacijskog sustava (kernel). Kernel čine datoteke *IO.SYS* i *MSDOS.SYS*. Datoteka *IO.SYS* sadrži adrese osnovnih upravljačkih programa koji se koriste za pogone (disketni, CD, tvrdi disk, ZIP...), te za portove (COM i LPT).
Datoteka *MSDOS.SYS* sadrži jegru operacijskog sustava.
17. Sljedeća datoteka koja se učitava u memoriju je *CONFIG.SYS*. Ona sadrži dodatne upravljačke programe za grafičku karticu, za skup znakova koji se koriste, zadefiniranje formata vremena i datuma. U datoteci *CONFIG.SYS* određuje se veličina RAM memorije koja će se koristiti za privremeno čuvanje veće količine podataka koji se prenose na disk.
18. Datoteka koja slijedi je *COMMAND.COM*. Ta datoteka kad se napuni u memoriju ima tri dijela. Ona je ustvari komandni procesor ili interpreter i to je dio operacijskog sustava sa kojim je korisnik stalno u vezi. Ta datoteka prihvaća unos niza znakova sa tipkovnice prevodi ih u naredbe koje se izvršavaju.
19. I posljednja datoteka (ako se radi o operacijskom sustavu DOS) je *AUTOEXEC.BAT*. Ta datoteka je zapravo batch datoteka koja sadrži naredbe.

13. Formiranje particije

U ovom poglavlju će djelomično biti pojašnjeni programi za izradu particija.

13.1. Microsoft Fdisk

Fdisk koji se nalazi na gotovo svakoj sistemskoj disketi ili na tvrdom disku je DOS –ov Fdisk. Taj program ima namjenu da pomoću njega izradujemo particije, ali ima mnogo nedostataka. Fdisk može napraviti particiju primarnu, proširenu ili logički disk, zatim može particiju postaviti aktivnom, izbrisati particiju ili logički disk te prikazati informacije o particijama. Prvi nedostatak je što DOS –ov Fdisk radi samo sa FAT datotečnim sustavom. Sljedeći problem je što na možemo napraviti particiju u određenim granicama (od cilindra do cilindra). Jedina prednost ovog programa je lako postavljanje particije aktivnom.

Program pokrećemo upisom naredbe **fdisk** u prompt, nakon čega nam se pojavi osnovni izgled programa. Ako želimo sa ovim programom napraviti particiju moramo znati veličinu ili postotak ukoliko želimo napraviti više particija. Problem postoji u tome da ako već imamo jednu primarnu particiju da nam proram ne dozvoljava da napravimo još jednu primarnu particiju. Dakle preostaje nam samo napraviti proširenu particiju i u njoj logičke diskove s time da opet moramo znati ili veličinu ili postotak diska koji će particija zauzimati.

Ovaj program može dobro doći onima koji nisu previše zahtjevni u radu sa tvrdim diskom.

13.2. Linux Fdisk

Linuxov fdisk se u mnogočemu razlikuje od DOS –ovog. Kao prvo u njemu je moguće particiju napraviti znajući ili početne i završne cilindre ili početni cilindar i veličinu. Ovaj program ima jednu veliku prednost nad DOS –ovim Fdisk –om, a to je da dozvoljava prikaz particija koje sadrže različite datotečne sustave. U ovome programu je omogućena i izmjena system flag –a, također boot flag se isto lako mijenja kao i u DOS –ovom programu. Nakon

Što je program napravi particiju on joj sam postavi system flag u Linux Native, pa se može dogoditi da pri podizanju sustava operacijaki sustav ne prepozna tu particiju. Jedna od najvećih pogodnosti ovog programa je da ne zapisuje odmah na disk kao DOS –ov program nego to moramo mi učiniti. Ukoliko načinimo pogrešku pri izradi particije ona nije odmah učinjena, pa tako da možemo vratiti staro stanje diska i početi iznova.

13.3. Partition Magic

Partition Magic je program koji ima grafičko sučelje lijepog izgleda. Partition Magic je program koji ne služi za izradu, brisanje i prikaz particija. Taj program može mijenjati veličinu particije i micati tu particiju. Jako dobra stvar pri micanju je da ukoliko imamo i podatke na toj particiji i oni će biti premješteni zajedno sa particijom. PQM ima mogućnost prikaza početne i završne adrese particije, veličinu particije, zauzetost. On ima mogućnost konvertiranja datotečnih sustava FAT i FAT32. Može mijenjati veličinu klastera neke particije. Ima prikaz iskoristivosti diska.

Za onog tko želi izvući maksimum iz svog tvrdog diska koristit će ili Linux Fdisk ili Partition Magic.

13.4. Formatiranje

Nakon svakog formiranja particije potrebno je formirati [datotečni sustav](#) particije i [logičku strukturu diska](#).

Program **Format.com** ima dvije mogućnosti. Formatiranje na niskoj i formatiranje na visokoj razini.

13.4.1. Formatiranje na niskoj razini

Formatiranjem na niskoj razini označavamo položaje staza i sektora na tvrdom disku. Prije prvog formatiranja na niskoj razini ploče diska su prazne i to je zadnji put da su prazne za čitavi radni vijek tvrdog diska. Jedino ukoliko ponovno formatiramo disk na niskoj razini tada će svi podaci biti trajno izbrisani. Formatiranje na niskoj razini jedino rješava *Bad sectore* na disku. Formatiranje na niskoj razini briše cijelu strukturu tvrdog diska, sve particije i datotečne sustave. Zbog kompleksnosti u radu sa novim diskovima velikog kapaciteta *Low Level Format* je načinjen u tvornici od strane proizvođača.

13.4.2. Formatiranje na visokoj razini

Nakon što je dovršeno formatiranje na niskoj razini i imamo disk sa sektorima i stazama potrebno je na taj disk upisati datotečni sustav. Formatiranje na visokoj razini se vrši kada je particija formirana. Da bi se očistio disk nije potrebno formatirati disk na niskoj razini već je dovoljno formatirati ga na visokoj razini i tako stvoriti novu upravljačku strukturu.

Različiti operacijski sustavi koriste različite programe za formatiranje na visokoj razini zbog različitih datotečnih sustava. Formatiranje na niskoj razini pri kojem vršimo upis staza i sektora za sve je datotečne sustave isti.

Ukoliko formatiranje vršimo na visokoj razini programom format možemo koristiti više parametara. Npr Format c: - normalno formatiranje

Format c: /u - unformat

Format c: /q - brzo formatiranje

Format c: /s - koristi se da bi se nakon formatiranja na disk kopirale sistemske datoteke

Literatura: www.open.hr/~sbrbot

