

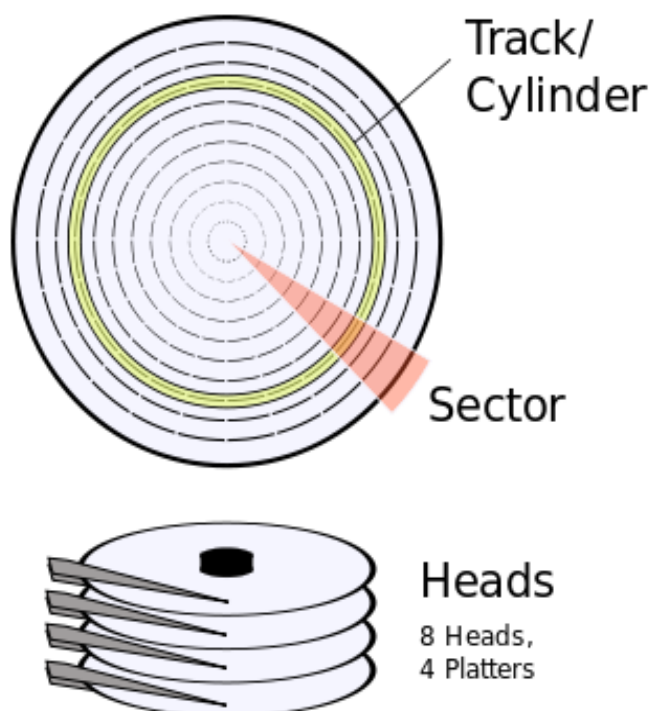
Adresiranje sektora na diskovima



U članku o naredbi [blockdev](#) [1] obećali da ćemo se pozabaviti "komplikacijama" vezanim uz parametre za iščitavanje/mijenjanje sektora/blokova na disku. Da bismo ih naučili razlikovati i razumijevati, vratit ćemo se u povijest, u prošli milenij, dok su računala još bila u "pelenama". U to doba se podacima na diskovima pristupalo preko CHS geometrije, koja je krajem osamdesetih zamijenjena LBA adresiranjem. Tada su iskovani i termini koje još uvijek koristimo, poput sektora, bloka, klastera, cilindra.

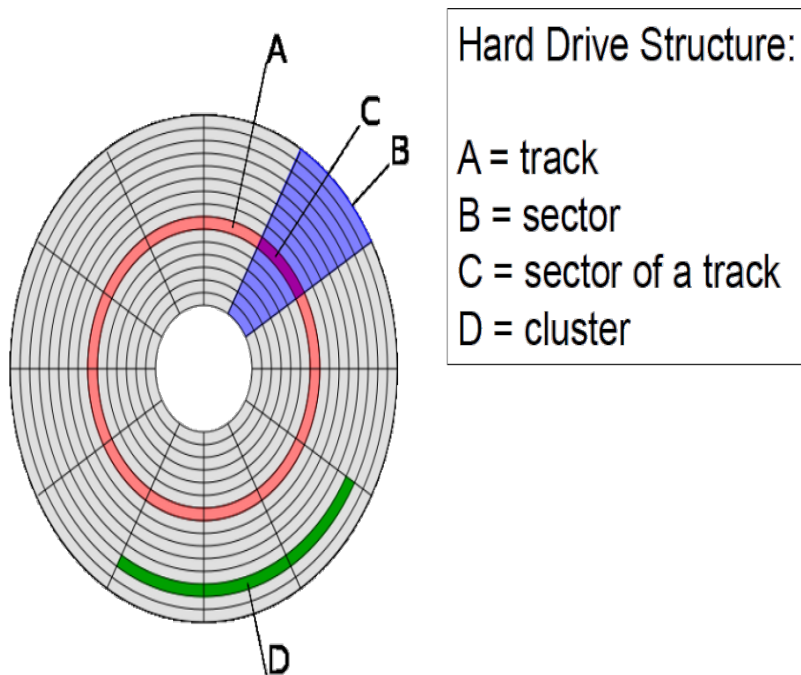
CHS

CHS je kratica za **cylinder-head-sektor**. Tvrdi disk se sastoji od nekoliko metalnih ploča premazanih magnetnim slojem, na koje se podaci zapisuju obostrano, pa svaka ploča ima dvije glave za čitanje/pisanje. Na svakoj strani ploče prostor je podijeljen u koncentrične trake, podijeljene na sektore. Sektor je najmanja količina podataka koji se mogu zapisati. Sve je to zorno prikazano na sljedećoj ilustraciji.



CHS ide od većeg prema manjem, od cilindra preko glave do sektora. Izgleda jednostavno, bar na prvi pogled. Ali nije baš tako.

Pogledajmo narednu ilustraciju.



Ovdje se izraz sektor pojavljuje u dva značenja: najprije kao "**geometrijski sektor**", u obliku kružnog isječka (kriška torte), a drugi put kao "**sektor trake**", presjek isječka i trake, prostor na kojem ćemo spremati podatke. U ovom drugom značenju se izraz sektor najčešće upotrebljava, kao najmanja jedinica na koju disk zapisuje podatke. Ponekad ćemo ga zvati **fizički sektor**, da bi se znalo da se odnosi na zapis na fizičkom mediju.

Veličina sektora kod tvrdih diskova se mijenjala, ali s vremenom je uspostavljena standardna veličina od 512 bajtova, a kod CD-a 2048 bajtova. Kako raste kapacitet medija, mijenja se i standard, kod novijih diskova sektor je 4096 bajtova i još će rasti.

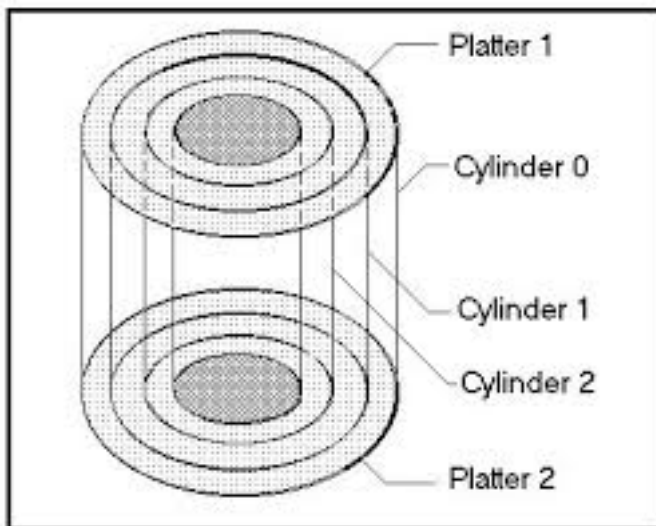
Slika 2 pokazuje i grupu od nekoliko sektora kao **cluster**. To je izraz iz Microsoftova svijeta koji se odnosi na najmanju jedinicu koju koristi datotečni sustav (FAT, NTFS) kad pristupa disku. To bi bio blok u terminologiji iz Unixova univerzuma. Ako je veličina bloka 512 bajtova, onda je blok jednak sektoru. Česće će blok biti veličine 4096 bajtova, dakle grupa od osam fizičkih sektora.

Prva stvar koja upada u oči na ovim ilustracijama jest da sektori bliži vanjskom rubu zauzimaju veći prostor od onih u središtu. A na svaki sektor možemo upisati istu količinu podataka! Sukladno tome razlikuje se i veličina traka, one vanjske su veće, ali sadrže isti broj sektora kao i one u sredini. Kakvo razbacivanje prostora!

Druga "komplikacija" jest u tome da ćemo najprije pomisliti da je cilindar isto što i ploča. Jer ploča je zapravo spljošteni cilindar, zar ne? No prva ilustracija nam sugerira da je cilindar isto što i traka!? To je neobično jer cilindar zapravo znači valjak, todimenzionalno tijelo, a traka je dvodimenzionalna. Dok sam mozgao o tome stariji me kolega podsjetio na činjenicu da je prvi IBM-ov hard disk zaista izgledao kao veliki cilindar, pravi pravcati valjak!



Cilindar nije nešto što možemo vidjeti očima na današnjim diskovima, morat ćemo ga zamisliti. Sve se glave kreću zajedno, na istoj su osovini, pa je cilindar skup traka na svim stranama ploča koje su jednako udaljene od centra. Podaci se zapisuju po vertikali, istovremeno na svim glavama, tvoreći trodimenzionalni presjek kroz sve ploče koji tvori oblik valjka, odnosno cilindra.



Većina ovih standarda potiče od IBM-a, što nije čudno jer je IBM (ako ste zaboravili) tvorac IBM-PC računala, koja su danas praktički standard, pa ih zovemo samo PC. IBM-ovi inženjeri iskoristili su svoja postojeća znanja, standarde, tehnike i terminologiju razvijenu za velika računala. Tako su CHS adresiranje koristili još 1960-tih na svojim mainframe računalima.

Radi unificirane geometrije diska u to smo doba koristili disk kontrolere koji su zauzimali jedan od utora za proširenja. Diskovi su bili klasificirani kao [MFM](#) [2] ili [RLL](#) [3] što ukazuje na način kodiranja podataka.

Hajdemo još malo zakomplicirati stvari. Adresu u CHS sustavu određuje takozvani "tuple" (uređen niz brojeva). Maksimalne vrijednosti u CHS adresi od 24 bita su 1023/255/63. C je broj traka na jednoj strani ploče, a istovremeno i broj (zamišljenih) cilindara. Ako ga pomnožimo s brojem glava (H), puta

broj sektora po traci (S) - dobit ćemo veličinu diska u sektorima. Ako rezultat pomnožimo još i s veličinom sektora, dobit ćemo kapacitet diska u bajtovima. To bi izgledalo ovako: $1023 \times 255 \times 63 \times 512 = 8,414.461.440$. Dakle najveći mogući disk je sa CHS adresiranjem ograničen na otprilike 8 GB podataka.

Ali onda primjećujemo još jednu nelogičnost: kako broj glava za čitanje može biti neparan? I nije neparan, tako se samo čini na prvi pogled jer CHS adresiranje počinje od 0/0/1, (cilindri i glave broje se od nule, ali sektori od jedinice!) Radi toga maksimalan broj glava za čitanje/pisanje zapravo nije 255 nego 256. U tom slučaju trebalo bi korigirati i izračun kapaciteta diska. Broj cilindara i glava treba povećati za jedan, jer kod njih brojanje počinje od 0, a broj sektora ne dirati: $1024 \times 256 \times 63 \times 512 = 8455716864$.

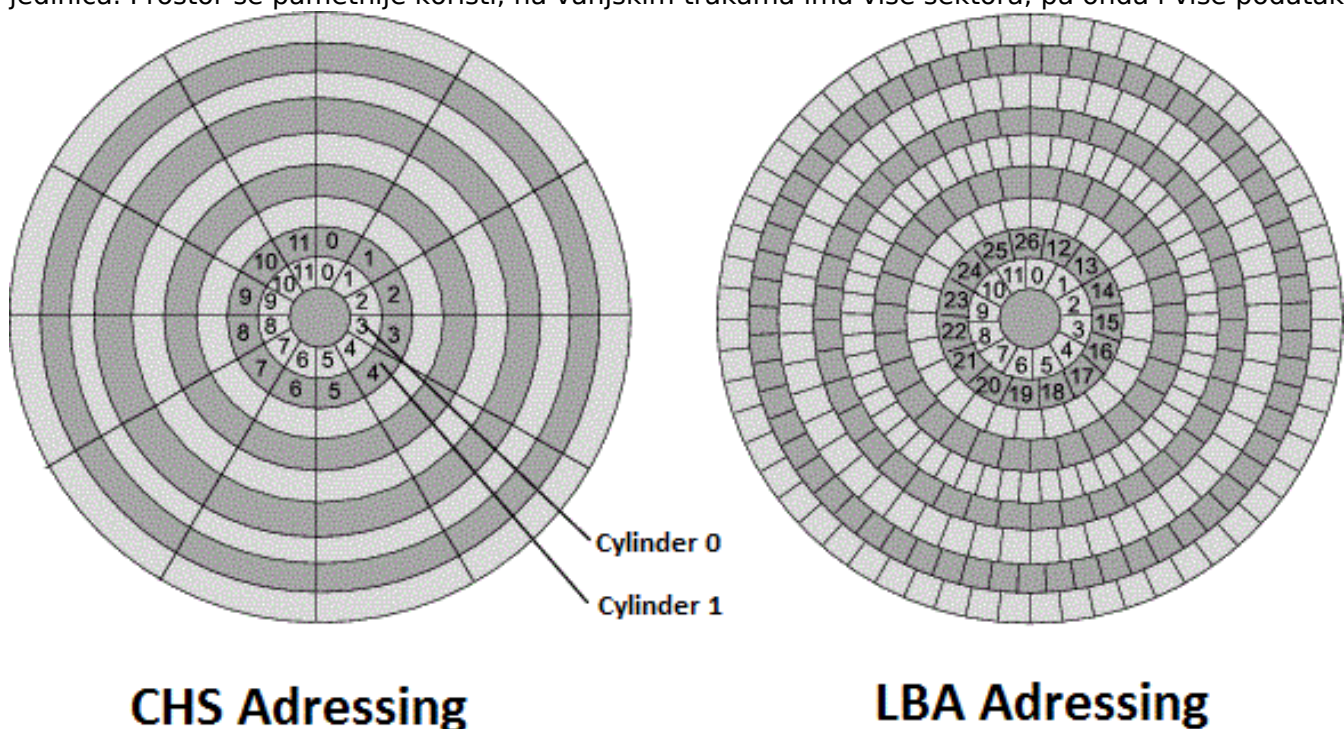
Za jako radoznale koji žele znati zašto se sektori broje od 1, a ne od 0, odgovor treba opet potražiti u naslijeđu prošlosti: autor prvog BIOS-a u vrijeme kad je IBM-PC još bootao OS s diskete rezervirao je sektor 0 za provjeru pozicioniranja glave. Kad se prešlo na tvrde diskove, pozicioniranje glave više nije bilo problem, pa bi bilo šteta ne iskoristiti i taj nulti sektor.

Ta je iznimka ili nedosljednost kasnije stvarala probleme programerima, pa i Microsoftovim, jer su verzije DOS-a do 7.0 krahirale kod bootanja ako je ako disk bio prijavljen s 256 glava. :)

Kod tehnologije koja se brzo razvija stalno su u igri ograničenja koja treba prevladati, na primjer ona u BIOS-u, zatim broja bitova u adresi itd. Kako se povećavala glad za memorijom, nužno je bilo smisliti način da se bolje iskoristi raspoloživi prostor, a onda i omogućiti adresiranje diskova većih od 8 GB. Tako je nastala nova generacija IDE diskova s LBA adresiranjem.

LBA

LBA je skraćenica od **Logical Block Addressing**. Novi je standard uveden kao apstrakcijski sloj, s namjerom da s vremenom zamijeni CHS adresiranje. Koristi jednostavno linearno adresiranje, od 0 nadalje, pa su adrese u jednodimenzionalnom nizu logičkih blokova, gdje je logički blok najmanja jedinica. Prostor se pametnije koristi, na vanjskim trakama ima više sektora, pa onda i više podataka.



Ilustracija zorno pokazuje kako je svaki sektor i fizički jednake veličine, pa vanjske trake sadrže više sektora.

Da bi diskovi različitih generacija i različitih proizvođača mogli bez problema raditi s PC računalima,

više se ne koristi univerzalni vanjski kontroler, već je on integriran u disku. Uz to LBA se više ne oslanja na BIOS da bi saznao nešto o disku. Kod CHS-a diskove je trebalo prijaviti BIOS-u, preko kojeg su detalji o fizičkoj organizaciji diska bili dostupni operativnom sustavu, dok LBA uvodi sloj koji apstrahira fizičku geometriju diska.

Jedna je od zadaća integriranog kontrolera jest posredovanje, prevođenje između LBA i CHS adresiranja kojeg su tada još očekivali BIOS, operacijski sustavi, driveri, neke aplikacije. Kod LBA adresiranja kontroler od softvera skriva stvarnu fizičku geometriju. Uvode se logički sektori/blokovi, za razliku od fizičkih. Tako logički sektor nula počinje na traci 0, glavi 0 i fizičkom sektoru 1. :)

Godine 2002. ATA-6 specifikacija uvodi 48 bitno LBA adresiranje i proglašava CHS adresiranje zastarjelim, ali u praksi ćemo se još dugo susretati s cilindrima i trakama.

Danas se sve više koriste flash mediji, u obliku SSD diskova (disk bez pokretnih dijelova), USB stikova ili SD kartica. Tu nema nikakvih cilindara i traka! Ali i dalje koristimo sektore i blokove, jer su i ti novi uređaji "block devices". No povremeno vam se može dogoditi da vam neka funkcija vrati CHS tuple i za flash memorije.

Ako bismo željeli rezimirati što smo naučili (podsjetili se), evo ukratko:

Sektor je najmanja veličina podataka koju šaljemo blok uređaju. Kod tvrdih diskova razlikujemo geometrijski sektor (kružni isječak) od fizičkog sektora (presjek trake i geometrijskog sektora). LBA adresiranje uvodi pojam logičkog sektora, skrivajući od OS-a fizičku adresu. Tu se u nazvu standarda više ne spominje sektor, već blok!

- geometrijski sektor - kružni isječak na ploči tvrdog diska
- fizički sektor - presjek trake i geometrijskog sektora
- logički sektor - sektor čija je fizička lokacija skrivena od OS-a

Povremeno ćemo u literaturi naći da se izrazi sektor i blok koriste bez jasnog razlikovanja. Načelno, sektori su jedinice na mediju, a blokovima se bavi datotečni sustav pojedinog operacijskog sustava, koji objedinjuje više blokova u vlastitu najmanju jedinicu, blok. No ostaje i dalje da je sektor zapravo blok podataka. :)

Sad se možemo vratiti parametrima naredbe blockdev s boljim razumijevanjem.

--getsz	dohvati veličinu (diska/particije) u sektorima od 512 bajtova
--getsize64	dohvati veličinu diska/particije u bajtovima
--getss	dohvati veličinu logičkog bloka (sektora)
--getpbsz	dohvati veličinu fizičkog bloka (sektora)
--getbsz	dohvati veličinu bloka u bajtovima

Kad smo tako raščistili osnovne pojmove i nostalgičarski se vratili u prošlost, u narednom nastavku možemo se vratiti naredbi blockdev i njein parametrima --setxxx pomoću kojih mijenjamo zadane vrijednosti blokova, sektora.

sub, 2018-03-31 14:04 - Aco DmitrovićKuharice: [Linux](#) [4]

Kategorije: [Operacijski sustavi](#) [5]

Vote: 5

Vaša ocjena: Nema Average: 5 (1 vote)

story_tag: [disk](#) [6]

[sektor](#) [7]

[blok](#) [8]

[CHS](#) [9]

[LBA](#) [10]

Source URL: <https://sysportal.carnet.hr./node/1799>

Links

[1] <https://sysportal.carnet.hr./node/1797/>

[2] https://en.wikipedia.org/wiki/Modified_Frequency_Modulation

[3] https://en.wikipedia.org/wiki/Run_Length_Limited

[4] <https://sysportal.carnet.hr./taxonomy/term/17>

[5] <https://sysportal.carnet.hr./taxonomy/term/26>

[6] <https://sysportal.carnet.hr./taxonomy/term/166>

[7] <https://sysportal.carnet.hr./taxonomy/term/224>

[8] <https://sysportal.carnet.hr./taxonomy/term/225>

[9] <https://sysportal.carnet.hr./taxonomy/term/226>

[10] <https://sysportal.carnet.hr./taxonomy/term/227>