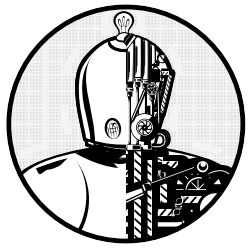
4

DISKOVI I SISTEMI DATOTEKA

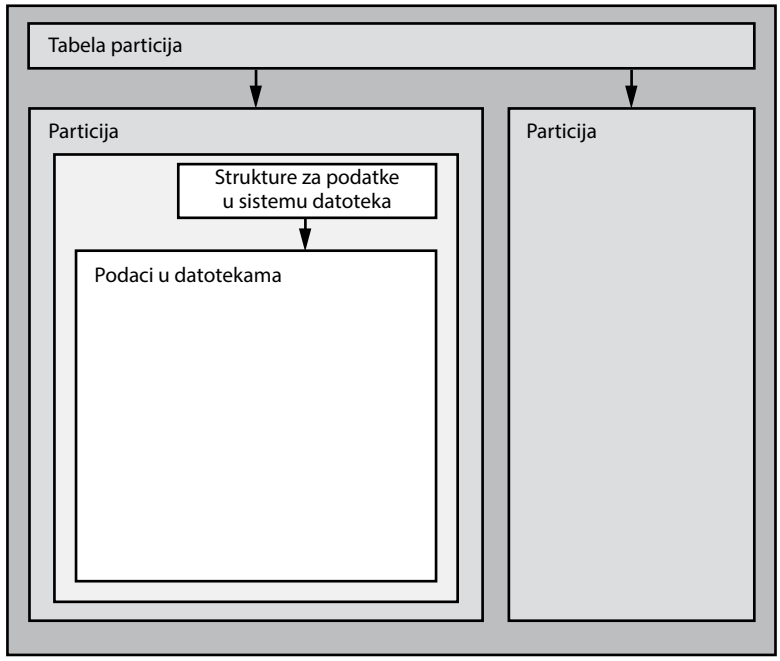


U poglavlju 3 razmatrali smo neke od disk uređaja najvišeg nivoa koje jezgro stavlja na raspolaganje u sistemu. U ovom poglavlju, detaljno opisujemo kako se radi s diskovima u Linux sistemu. Naučićete da delite diskove na particije, formirate i održavate sisteme datoteka na particijama i radite s prostorom za razmenu.

Podsećamo da disk uređaji imaju imena kao što je */dev/sda*, prvi disk u SCSI podsistemu. Ta vrsta blok uređaja predstavlja jedan ceo disk, ali unutar jednog diska postoji više raznih podela i slojeva.

Slika 4-1 ilustruje šemu tipičnog Linuxovog diska (imajte u vidu da slika ne prikazuje tačne razmere). Tokom napredovanja kroz ovo poglavlje saznaćete gde se svaki deo uklapa u celinu.

Particije (engl. *partitions*) jesu oblasti na koje je izdeljen ceo prostor diska. U Linuxu, one se obeležavaju rednim brojem iza imena blok uređaja, pa zato postoje imena uređaja kao što su */dev/sda1* i */dev/sdb3*. Jezgro predstavlja svaku particiju kao jedan blok uređaj, u istom obliku kao i ceo disk. Raspored particija je definisan u jednoj maloj oblasti diska koja se zove tabela particija (engl. *partition table*).



Slika 4-1: Tipična šema Linuxovog diska

NAPOMENA *Deljenje diska na više particija nekad je bilo uobičajeno na sistemima s velikim diskovima zato što su stariji PC računari mogli da se podižu samo sa određenih delova diska. Osim toga, administratori su koristili particije da bi rezervisali izvesnu količinu prostora u delovima namenjenim operativnom sistemu; na primer, nisu želeli da korisnici budu u stanju da zauzmu ceo sistem i tako onemoguće rad važnih sistemskih servisa. Ta praksa nije jedinstvena za Unix; i dalje ćete naći brojne nove Windows sisteme s više particija na jedinom disku. Osim toga, većina sistema ima posebnu particiju za razmenu.*

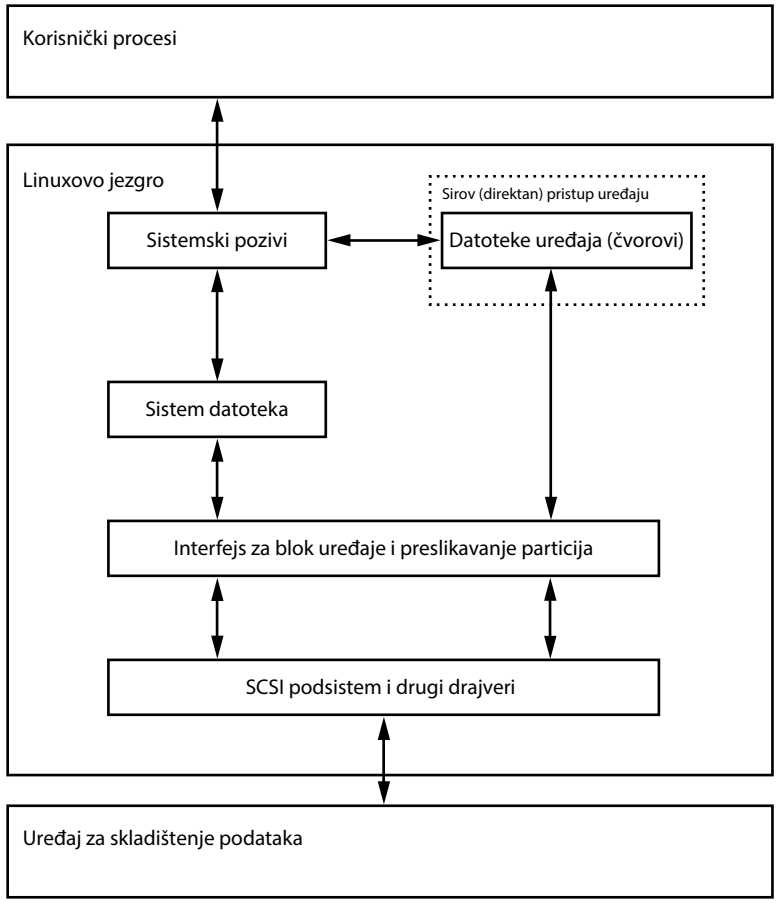
Mada jezgro pruža mogućnost da pristupate istovremeno i celom disku i jednoj od njegovih particija, to u normalnim okolnostima ne biste činili, osim kada kopirate ceo disk.

Sledeći sloj ispod particije jeste sistem datoteka (engl. *filesystem*), što je baza podataka datoteka i direktorijuma s kojom ste navikli da radite u korisničkom prostoru. Sisteme datoteka razmatramo u odeljku 4.2.

Kao što se vidi na slici 4-1, ako želite da pristupate podacima u nekoj datoteci, iz tabele particija morate pribaviti lokaciju odgovarajuće particije i zatim pretražiti bazu podataka sistema datoteka na toj particiji da biste dobili podatke iz tražene datoteke.

Kada pristupate podacima na disku, Linuxovo jezgro koristi sistem slojeva prikazan na slici 4-2. SCSI podsistem i sve ostalo opisano i odeljku 3.6 ovde je predstavljeno jednim pravougaonikom. (Imajte u vidu da s diskom možete raditi i kroz sistem datoteka i direktno pomoću disk uređaja. U ovom poglavlju radićete i jedno i drugo.)

Da biste stekli utisak kako se sve to zajedno uklapa, krenućemo od najnižeg nivoa, tj. od particija.



Slika 4-2: Šema načina na koji se disku pristupa iz jezgra

4.1 Deljenje disk uređaja na particije

Postoji više vrsta tabela particija. Tradicionalna tabela je format koji se nalazi u MBR (*Master Boot Record – glavni zapis za podizanje sistema*). Noviji standard koji počinje da stiče popularnost jeste *GPT (Globally Unique Identifier Partition Table – tabela particija s globalno jedinstvenim identifikatorima)*.

Evo kratkog pregleda brojnih alatki za deljenje diskova na particije u Linuxu:

`parted` Tekstualna alatka koja podržava i MBR i GPT.

`gparted` Grafička verzija alatke `parted`.

`fdisk` Tradicionalna tekstualna Linuxova alatka za izradu particija koja ne podržava GPT.

`gdisk` Verzija programa `fdisk` koja podržava GPT ali ne MBR.

Budući da podržava i MBR i GPT format, u ovoj knjizi ćemo koristiti program `parted`. Međutim, mnogi ljudi više vole interfejs programa `fdisk`, u čemu nema ničeg lošeg.

NAPOMENA *Mada program `parted` može da pravi nove sisteme datoteka i menja veličinu postojećih, nije preporučljivo da ga koristite za manipulisanje sistemima datoteka jer lako može doći do zabune. Postoji ključna razlika između deljenja diska na particije i manipulisanja sistemom datoteka. Tabela particija samo definiše određene granice na disku, dok je sistem datoteka znatno složeniji sistem za rad s podacima. Iz tog razloga, za deljenje diska na particije koristićemo program `parted` a sisteme datoteka pravićemo pomoću drugih alatki (videti odeljak 4.2.2). Čak se i u dokumentaciji programa `parted` savetuje da sisteme datoteka pravite pomoću zasebnih alatki.*

4.1.1 Pregledanje tabele particija

Tabelu particija u svom sistemu možete prikazati pomoću komande **`parted -l`**. Evo kako izgleda primer rezultata na računaru s dva disk uređaja i dve različite vrste tabela particija:

```
# parted -l
Model: ATA WDC WD3200AAJS-2 (scsi)
Disk /dev/sda: 320GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos

Number  Start   End    Size    Type      File system     Flags
 1      1049kB  316GB  316GB   primary   ext4            boot
 2      316GB   320GB  4235MB  extended
 5      316GB   320GB  4235MB  logical   linux-swap(v1)

Model: FLASH Drive UT_USB20 (scsi)
Disk /dev/sdf: 4041MB
Sector size (logical/physical): 512B/512B
Partition Table: gpt

Number  Start   End     Size    File system  Name        Flags
 1      17.4kB  1000MB  1000MB               myfirst
 2      1000MB  4040MB  3040MB               mysecond
```

Prvi uređaj, */dev/sda*, koristi tradicionalnu MBR tabelu particija (koju `parted` zove „msdos“), a drugi disk sadrži tabelu tipa GPT. Obratite pažnju na to da se za svaku tabelu particija prikazuju različiti parametri, zato što su u pitanju drugačije tabele. Konkretno, u MBR tabeli nema kolone Name (ime) zato što u tom formatu ne postoji ime particije. (U GPT tabeli sam proizvoljno izabrao imena `myfirst` i `mysecond`.)

U ovom primeru, MBR tabela sadrži primarnu, proširenu i logičku particiju. *Primarna* (engl. *primary) particija* standardni je oblik podele diska na manje oblasti; takva particija je particija 1. Pošto je osnovni MBR ograničen na najviše četiri primarne particije, ako vam treba više od četiri, jednu od particija definišete kao proširenu (engl. *extended) particiju*. Proširenu particiju

dalje delite na *logičke* (engl. *logical) particije* koje operativni sistem može da koristi isto kao i svaku drugu particiju. U ovom primeru, particija 2 je proširena particija koja sadrži logičku particiju 5.

NAPOMENA *Podatak o tipu sistema datoteka (system ID) koji* `parted` *prikazuje, ne mora obavezno biti sadržaj polja system ID koje je definisano u većini MBR stavki. MBR-ov system ID je običan broj; na primer, 83 predstavlja standardnu Linuxovu particiju a 82 Linuxovu particiju za razmenu. To znači da* `parted` *pokušava da sam prepozna tip sistema datoteka. Ako apsolutno morate da znate sadržaj polja system ID u MBR tabeli, upotrebite* `fdisk -l`.

Početno učitavanje u jezgro

Kada na početku učitava MBR tabelu, Linuxovo jezgro šalje sledeću dijagnostičku poruku (podsećamo da je možete videti pomoću komande `dmesg`):

```
sda: sda1 sda2 < sda5 >
```

Deo `sda2 < sda5 >` znači da je */dev/sda2* proširena particija koja sadrži jednu logičku particiju, */dev/sda5*. Proširene particije ćete uglavnom zanemarivati jer ćete, u normalnim okolnostima, uvek pristupati samo logičkim particijama unutar proširene.

4.1.2 Menjanje tabela particija

Pregledanje tabela particija relativno je jednostavna i bezopasna operacija. Mada je menjanje postojećih tabela particija takođe relativno jednostavno, tu vrstu promena na disku prati i određeni rizik. Imajte na umu sledeće:

- Menjanje tabele particija značajno otežava restauriranje podataka na particijama koje izbrišete zato što se tako menja inicijalna referentna tačka u sistemu datoteka. Obavezno napravite rezervnu kopiju podataka ako disk čije particije menjate sadrži važne podatke.
- Postarajte se da nijedna particija na ciljnom disku nije u upotrebi. To je važan korak zato što većina Linux distribucija automatski montira svaki sistem datoteka koji nađe. (Više informacija o montiranju i demontiranju diskova naći ćete u odeljku 4.2.3.)

Kada budete spremni, izaberite program za deljenje na particije. Ako biste želeli da koristite `parted`, možete upotrebiti alatku za komandnu liniju `parted` ili neki grafički interfejs kao što je `gparted`; ako želite interfejs u stilu programa `fdisk`, upotrebite `gdisk` ako radite s GPT particijama. Za te alatke postoji ugrađena pomoć i lako se uče. (Isprobajte njihovu upotrebu ne nekom fleš uređaju ili nečem sličnom ako nemate rezervne diskove.)

Međutim, postoji bitna razlika između načina na koji `fdisk` i `parted` rade. Kada koristite `fdisk`, novu tabelu particija morate osmisliti pre nego što zaista izmenite disk; `fdisk` unosi izmene tek kad izađete iz tog programa. Nasuprot tome, `parted` pravi, menja i briše particije *čim izdate komandu*. Nemate priliku da pregledate tabelu particija pre nego što je izmenite.

Te razlike su važne i za razumevanje kako te dve alatke komuniciraju s jezgrom. I `fdisk` i `parted` menjaju particije u potpunosti u korisničkom prostoru; nema potrebe da se u jezgro ugrađuje podrška za ponovno pisanje tabele particija zato što se iz korisničkog prostora može čitati i menjati sadržaj celog blok uređaja.

Međutim, jezgro mora da u nekom trenutku učita tabelu particija kako bi moglo da ih predstavi u obliku blok uređaja. Alatka `fdisk` koristi relativno jednostavnu metodu: pošto izmeni tabelu particija, `fdisk` šalje sistemski poziv kojim obaveštava jezgro da treba da ponovo učita tabelu particija. Jezgro zatim generiše dijagnostičku poruku koju možete videti pomoću alatke `dmesg`. Na primer, ako na */dev/sdf* napravite dve particije, videćete sledeće:

```
sdf: sdf1 sdf2
```

S druge strane, alatke `parted` ne koriste sistemski poziv koji se odnosi na ceo disk. Umesto toga, signaliziraju jezgru kad god izmene neku particiju na disku. Pošto obradi izmenu koja se odnosi na jedni particiju, jezgro ne generiše dijagnostičku poruku kao u prethodnom slučaju.

Postoji nekoliko načina da saznate da je neka particija izmenjena:

- Upotrebite komandu `udevadm` da biste pratili događaje o izmenama koje jezgro šalje. Na primer, `udevadm monitor --kernel` pokazaće da su uklonjeni uređaji koji su predstavljali stare particije i da su dodati novi.
- Pregledajte */proc/partitions* ako vam trebaju kompletni podaci o particijama.
- Pregledajte */sys/block/device/* da biste videli sistemske interfejse izmenjenih particija odnosno */dev* da biste videli uređaje koji predstavljau izmenjene particije.

Ako morate biti apsolutno sigurni da ste zaista izmenili tabelu particija, pomoću komande `blockdev` možete poslati sistemski poziv u starom stilu koji `fdisk` koristi. Na primer, da biste naredili jezgru da ponovo učita tabelu particija na disku */dev/sdf*, zadajte sledeće:

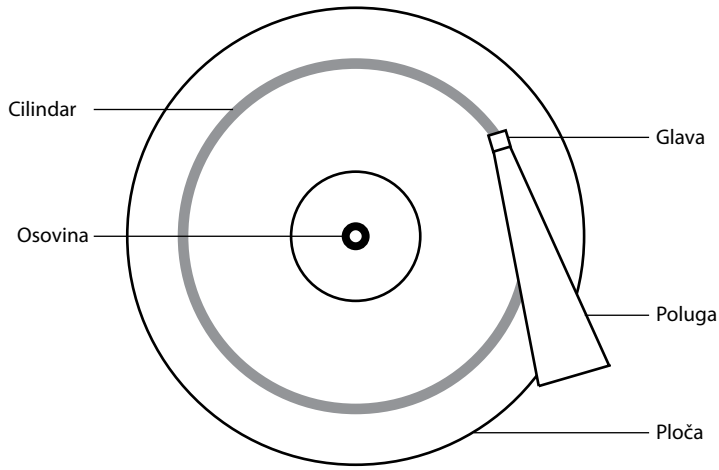
```
# blockdev --rereadpt /dev/sdf
```

Zasad je ovo sve što treba da znate o podeli diska na particije. Međutim, ako vas zanima nešto više o diskovima, nastavite čitanje. U suprotnom, pređite na odeljak 4.2 da biste naučili kako se formira nov sistem datoteka na disku.

4.1.3 Geometrija diska i particija

Svaki uređaj s pokretnim delovima uvodi dodatnu komplikaciju u softverski sistem zato što postoje fizički elementi koje je teško apstrahovati. Čvrsti disk nije izuzetak; čak i ako čvrsti disk zamislite kao blok uređaj s nasumičnim pristupanjem svakom bloku, postoje ozbiljne posledice po performanse ako ne vodite računa o tome kako raspoređujete podatke na disk. Razmotrite fizička svojstva jednostavnog diska s jednom pločom, prikazanog na slici 4-3.

Disk se sastoji od ploče koja se okreće oko svoje osovine, i od glave na pokretnoj poluzi koja može da se kreće duž radijusa diska. Kako se disk okreće ispod glave, glava učitava podatke. Kada se glava nalazi u datom položaju, može da čita podatke samo iz fiksnog kruga na disku. Taj krug se zove *cilindar* zato što veći diskovi imaju više od jedne ploče, koje su sve nanizane na istu osovinu. Svaka ploča može imati jednu ili dve glave, za početak i/ili kraj ploče, a sve glave su postavljene na istu polugu i kreću se zajedno. Pošto se poluga pokreće, na disku postoji veliki broj cilindara, od malih u blizini centra, do velikih blizu periferije diska. I najzad, cilindar se može podeliti na isečke koji se zovu *sektori*. Taj način predstavljanja geometrije diska zove se *CHS*, od *cylinder-head-sector* (cilindar-glava-sektor).



Slika 4-3: Prikaz elemenata čvrstog diska

NAPOMENA *Staza* (engl. *track) jeste ceo deo cilindra kojem može da pristupa jedna glava, pa je zato na slici 4-3, cilindar isto što i staza. Verovatno vam neće trebati da brinete i o stazama.*

Jezgro i razni programi za deljenje na particije mogu da vam kažu šta disk prijavljuje kao svoj broj cilindara (i *sektora*, što su isečci cilindara). Međutim, na savremenom čvrstom disku, te *prijavljene vrednosti ne odgovaraju ničem stvarnom*! Tradicionalan način adresiranja koji koristi CHS nije primenjiv na savremen hardver diskova, niti uzima u obzir činjenicu da možete smestiti više podataka na spoljašnje cilindre nego na unutrašnje. Hardver današnjih diskova podržava adresiranje po logičkim blokovima (*Logical Block Addressing, LBA)* gde se lokacija na disku adresira jednostavno po broju bloka, ali ostaci CHS-a i dalje žive. Na primer, MBR tabela particija sadrži i CHS podatke i njihove LBA ekvivalente, a neki programi za podizanje sistema i dalje su toliko ograničeni da veruju CHS vrednostima (ne brinite – većina programa za podizanje Linuxa koristi LBA vrednosti).

Uprkos tome, ideja cilindara je važna za podelu na particije zato što su cilindri savršene granice za particije. Tok podataka se sa cilindra učitava vrlo

brzo, zato što glave mogu da bez prekida primaju podatke tokom obrtanja ploče diska. Particija uređena kao skup susednih cilindara omogućava i brzo pristupanje podacima koji slede jedni drugima, zato što glava ne mora mnogo da se pomera između cilindara.

Neki programi za deljenje na particije „protestuju“ ako granice particija ne postavite tačno na granice cilindara. Zanemarite te proteste jer po tom pitanju možete malo toga uraditi pošto prijavljene CHS vrednosti na savremenim diskovima jednostavno nisu tačne. LBA podela diska obezbeđuje da vaše particije budu tačno tamo gde hoćete da budu.

4.1.4 Diskovi bez pokretnih delova (SSD)

Uređaji za skladištenje podataka bez pokretnih delova, kao što su elektronski diskovi (engl. *solid-state disks, SSDs)*, radikalno se razlikuju od elektromehaničkih diskova po karakteristikama pristupanja podacima. U njihovom slučaju, nasumično pristupanje nije problem zato što nema glave koja se kreće iznad ploče, ali određeni činioci ipak utiču na performanse.

Jedan od najznačajnijih činilaca koji utiče na performanse SSD uređaja jeste poravnavanje particija. Kada učitavate podatke sa SSD uređaja, to činite u blokovima – najčešće po 4096 bajtova odjednom – a čitanje mora da započne od nekog umnoška te veličine. Ako particija i podaci u njoj ne počinju na nekoj 4096-bajtnoj granici, možda će vam trebati dva učitavanja za jednu kratku i vrlo čestu operaciju, kao što je učitavanje sadržaja direktorijuma.

Budući da su mnoge alatke za deljenje na particije (na primer, `parted` i `gparted`) u stanju da novodefinisane particije postave na odgovarajuća rastojanja od početka diskova, verovatno nećete nikad morati da brinete o pogrešnom poravnavanju particija. Međutim, ako ste radoznali u vezi s time gde vaše particije počinju i želite samo da se uverite da zaista počinju na odgovarajućoj granici, do tog podatka možete lako doći ako pogledate u */sys/block*. Evo primera particije na */dev/sdf2*:

```
$ cat /sys/block/sdf/sdf2/start
1953126
```

Ova particija počinje na 1.953.126 bajtova od početka diska. Pošto taj broj nije deljiv s 4096, particija ne bi pružala optimalne performanse kada bi bila na SSD uređaju.

4.2 Sistemi datoteka

Poslednja karika lanca između jezgra i korisničkog prostora u slučaju diskova obično je sistem datoteka; to je ono s čime ste navikli da radite kada zadajete komande kao što su `ls` i `cd`. Kao što smo već pomenuli, sistem datoteka je oblik baze podataka, koja obezbeđuje strukturu pomoću koje se jednostavan blok uređaj pretvara u složenu hijerarhiju datoteka i poddirektorijuma koju korisnici mogu da razumeju.

Nekada su sistemi datoteka bili smešteni na diskove i druge fizičke medijume koji su se koristili isključivo za čuvanje podataka. Međutim, pošto su struktura u obliku stabla i ulazno/izlazni interfejs sistema datoteka prilično

raznovrsni, sistemi datoteka danas obavljaju razne vrste poslova, kao što su sistemski interfejsi koje vidite u */sys* i */proc*. Osim toga, sistemi datoteka se tradicionalno implementiraju u jezgru, ali inovacija 9P iz Plana 9 (*http://plan9.bell-labs.com/sys/doc/9.html*) inspirisala je razvoj sistema datoteka i u korisničkom prostoru. Mogućnost FUSE (*File System in User Space – sistem datoteka u korisničkom prostoru*) dozvoljava sisteme datoteka u korisničkom prostoru na Linuxu.

Sloj apstrakcije *VFS (Virtual File System – virtuelni sistem datoteka*) zaokružuje implementaciju sistema datoteka. Slično kao što SCSI podsistem standardizuje komunikaciju između pojedinih tipova uređaja i upravljačkih komandi iz jezgra, VFS obezbeđuje da sve konkretne implementacije sistema podržavaju standardizovan interfejs kako bi aplikacije iz korisničkog prostora mogle da na isti način pristupaju datotekama i direktorijumima. Podrška za VFS je omogućila Linuxu da podržava izuzetno veliki broj različitih sistema datoteka.

4.2.1 Tipovi sistema datoteka

Linuxova podrška za sistem datoteka obuhvata izvorne dizajne koji su optimizovani za Linux, spoljašnje tipove kao što je Windowsova FAT porodica, univerzalne sisteme datoteka kao što je ISO 9660 i mnoge druge. Sledeća lista sadrži najuobičajenije tipove sistema datoteka za skladištenje podataka. Imena tipova koje Linux prepoznaje navedena su između zagrada, pored imena sistema datoteka.

- *Fourth Extended filesystem* (ext4) tekuća je iteracija linije sistema datoteka koja je izvorno projektovana za Linux. Verzija *Second Extended filesystem* (ext2) dugo se podrazumevala u Linux sistemima koji su vodili poreklo od tradicionalnih Unixovih sistema datoteka kao što je UFS (Unix File System) i FFS (Fast File System). Verzija *Third Extended filesystem* (ext3) dodala je mogućnost koja se zvala žurnal (mali keš izvan normalne strukture za podatke sistema datoteka) i koja je poboljšavala integritet podataka i ubrzavala podizanje sistema. Verzija ext4 tog sistema datoteka inkrementno je poboljšanje koje je uvelo podršku za datoteke veće nego što su podržavale verzije ext2 ili ext3 i podršku za više poddirektorijuma.

  Postoji izvestan nivo kompatibilnosti unazad između članova ovog niza sistema datoteka. Na primer, svaku verziju ext2 i ext3 možete montirati kao da je ona druga, a ext2 i ext3 možete montirati i kao ext4, ali *ne možete* montirati ext4 kao ext2 ili ext3.

- *ISO 9660* (iso9660) jeste standard za CD-ROM. Većina CD-ROM uređaja koristi neku varijantu standarda ISO 9660.
- *FAT sistemi datoteka* (msdos, vfat, umsdos) razvijeni su za Microsoftove sisteme. Jednostavan tip msdos podržava vrlo primitivnu varijantu s jednom vrstom slova (samo velika) u MS-DOS sistemima. U većini savremenih sistema datoteka za Windows, trebalo bi da koristite sistem datoteka vfat kako biste imali pun pristup iz Linuxa. Sistem datoteka umsdos retko se koristi i svojstven je Linuxu. Podržava Unixove mogućnosti kao što su simboličke veze povrh sistema datoteka za MS-DOS.
- *HFS+* (hfsplus) Appleov je standard koji se koristi u većini Macintosh sistema.