Sveučilište J. J. Strossmayera u Osijeku

Fakultet elektrotehnike, računarstva i informacijskih tehnologija

Kneza Trpimira 2b

HR-31000 Osijek

www.ferit.unios.hr

Laboratorijska vježba 2:

Rad s datotečnim sustavom operacijskog sustava Linux

Sadržaj

1. UVOD

Svi računalni programi pohranjuju i dohvaćaju informacije. Proces u izvršavanju može spremiti ograničenu količinu podataka u svom adresnom prostoru. Za neke aplikacije ta je količina adekvatna, no u većini slučajevima je nedovoljna (npr. bankarske, financijske, turističke aplikacije i sl.). Osim toga, kada proces završi s izvršavanjem, briše se cijeli njegov adresni prostor, pa time i pohranjeni podaci. Također značajni problem pohranjivanja podataka u adresnom prostoru procesa se javlja kada više procesa istovremeno mora pristupiti istoj informaciji.

Kroz povijest je razvijen velik broj različitih uređaja za pohranjivanje digitalnog sadržaja, od magnetskih do SSD (engl. *solid-state drives*) diskova. Kako bi iskoristio takve uređaje, operacijski sustav stvara novu apstrakciju nekog sadržaja, koju naziva **datoteka** (engl. *file*). Uz procese i adresni prostor, datoteka predstavlja jednu od najvažnijih apstrakcija operacijskog sustava. Datoteke su logične jedinice informacija kreirane od strane procesa te su pohranjene na tvrdom disku. Procesi mogu čitati postojeće datoteke i kreirati nove, a u njima pohranjena informacija mora biti **dosljedna** (engl. *persistent).* Upravljanje datotekama provodi operacijski sustav, a ono podrazumijeva njihovo strukturiranje, imenovanje, pristup, korištenje, zaštita i implementiranje. U cjelini, dio operacijskog sustava zadužen za upravljanje datotekama se naziva **datotečni sustav** (engl. *file system*).

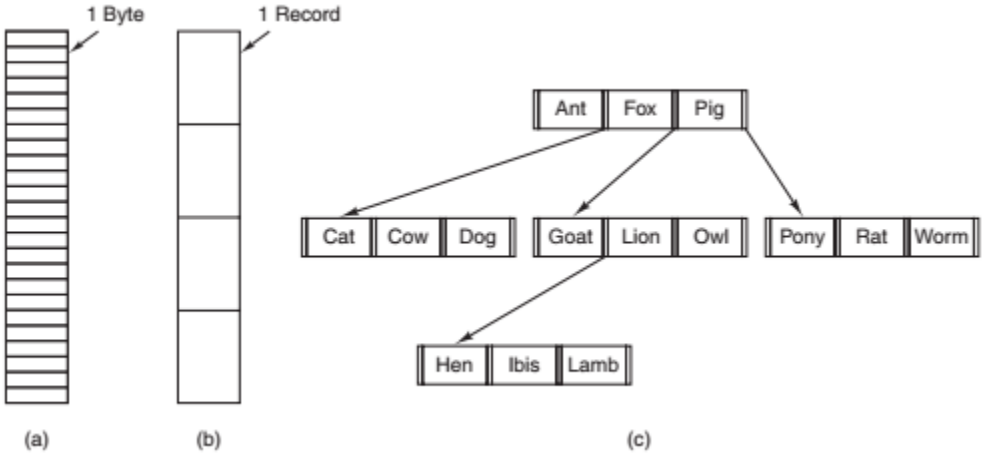
U ovoj laboratorijskoj vježbi ćemo naučiti osnovne teorijske koncepte operacijskog sustava za upravljanje datotekama. Detaljnije ćemo se upoznati s datotečnim sustavom operacijskog sustava Linux te s naredbama ljuske koje njegovim korisnicima omogućuju normalan rad s datotekama i direktorijima. Također ćemo usvojiti koncepte sigurnosti datotečnog sustava u Linuxu te prikazati naredbe za izradu sigurnosnih kopija.

1.1. Datoteke

Datoteke su apstrakcijski mehanizam operacijskog sustava. Pružaju način pohranjivanja informacija na disk te povratnog čitanja informacija s diska. Putem tog apstrakcijskog mehanizmi, operacijski sustav štiti korisnike od detalja fizičkog pristupa disku i njegovog rada općenito. Kada proces stvara datoteku pridjeljuje joj ime, a kada on završava s izvršavanjem, datoteka nastavlja postojati i mogu joj pristupiti i drugi procesi. Kad je riječ o nazivlju datoteka, većina operacijskih sustava podržava dvodijelna imena datoteka, sastavljena od imena i nastavka (ekstenzije). Neke od najčešćih ekstenzija datoteka su: .exe, .txt, .pdf, .zip, .jpg, .html, .c, .mp3, … U nekim sustavima (npr. UNIX) nastavci datoteka su samo konvencije, ali nisu podržani od strane operacijskog sustava.

1.1.1. Struktura datoteke

Datoteke mogu biti strukturirane u nekoliko načina. Najčešći načini su prikazani na Slici 1. Prvi način strukturiranja datoteka je kao slijed bajtova, pri čemu operacijski sustav ne poznaje pravi sadržaj datoteke nego samo vidi bajtove. Bilo koje značenje moraju nametnuti korisničke aplikacije. Takva struktura datoteka omogućuje maksimalnu razinu fleksibilnosti, a koriste ju UNIX i Windows. U drugom slučaju (Slika 1. b), datoteka je slijed zapisa fiksne duljine koji imaju određenu unutarnju strukturu. Centralna ideja oko ovakvog zapisa leži u načinima čitanja i zapisivanje datoteke kao jedan zapis. Ovaj način strukturiranja je u povijesti prevladavao, no danas se gotovo ne koristi. Prilikom strukturiranja datoteke kao stablo, zapisi ne moraju biti jednake duljine, a svaki sadrži ključ prema kojem se stablo sortira. Ovakav način sortiranja je prikladan za brzo pretraživanje, a koriste ga mainframe računala za komercijalno procesiranje podataka.



Slika 1. Tri strukture datoteka. (a) Slijed bajtova. (b) Slijed zapisa. (c) Stablo

1.1.2. Tipovi datoteka

Mnogo operacijskih sustava podražava različite vrste datoteka. UNIX i Windows imaju regularne datoteke i direktorije, te UNIX još sadrži i znakove i specijalne blok datoteke. **Regularne datoteke** su one koje sadrže korisničke informacije. **Direktoriji** su sustavske datoteke koje služe za održavanje strukture datotečnog sustava. **Posebne znakovne datoteke** (engl. *character special files*) su vezane uz ulaz/izlaz i modeliraju I/O uređaje, kao što su terminali, pisači i mrežni uređaji. **Posebne blok datoteke** (engl. *block special files*) modeliraju diskove.

Regularne datoteke se dijele na ASCII i binarne datoteke. ASCII datoteke čine linije teksta, a prednost im je što se lako prikazuju i ispisuju, te ih se može uređivati putem tekstualnih editora. Binarne datoteke su sve datoteke koje nisu ASCII, a imaju određenu unutarnju strukturu. Najpoznatiji primjeri takvih datoteka su izvršne datoteke i arhive.

1.1.3. Pristup datotekama

Rani operacijski sustavi su omogućavali samo jedan način pristupa datotekama: sekvencijalni pristup, prema kojem se datoteke čitaju bajt po bajt (ili zapis po zapis) po redu. Sekvencijalno čitanje datoteka je bilo prikladno za medije za pohranu kao što su magnetska traka. S pojavom diska, postalo je moguće čitati datoteke izvan određenog redoslijeda te pristupiti točnoj poziciji unutar datoteke. Datoteke čiji se bajtovi ili zapisi mogu pročitati u bilo kojem redoslijedu se nazivaju datoteke s nasumičnim pristupom (engl. *random-access files*). Ovakve datoteke su od izuzetne važnosti za različite aplikacije (npr. baze podataka).

1.1.4. Datotečni atributi

Osim imena i podataka, operacijski sustav pridružuje dodatne informacije uz datoteku. Te dodatne informacije se nazivaju metapodacima ili datotečnim atributima (engl. *file's attributes*), a najčešći su vrijeme stvaranja, vrijeme modifikacije te veličina datoteke. Tablica 1 prikazuje neke mogućnosti datotečnih atributa.

Tablica 1. Lista najčešćih datotečnih atributa

Atribut	Značenje
Protection	Tko može pristupiti datoteci i na koji način
Password	Zaporka za pristupanje datoteci
Creator	ID kreatora datoteke
Owner	ID trenutnog vlasnika datoteke
Read-only flag	0 za čitanje/pisanje; 1 samo za čitanje
Hidden flag	0 za normalno; 1 za neprikazivanje
System flag	0 za normalno; 1 za sistemske datoteke
Archive flag	0 za izrađenu sigurnosnu kopiju; 1 za potrebu za kopiranjem
ASCII/binary flag	0 za ASCII datoteku; 1 za binarnu datoteku
Random access flag	0 za sekvencijalni pristup; 1 za nasumični pristup
Temporary flag	0 za normalno; 1 za brisanje datoteke po izvršavanju
Lock flags	0 za nezaključane; 1 za zaključane datoteke
Record length	Broj bajtova u zapisu
Key position	Pomak ključa unutar zapisa
Key length	Broj bajtova za polje ključa
Creation time	Datum i vrijeme kreiranja datoteke
Time of last access	Datum i vrijeme zadnjeg pristupa datoteci
Time of last change	Datum i vrijeme zadnje promjene datoteke
Current size	Broj bajtova u datoteci
Maximum size	Maksimalni mogući broj bajtova u datoteci

1.1.5. Operacije s datotekama

Najčešće operacije s datotekama su:

- Kreiranje (engl. *create*) - datoteka se kreira bez podataka
- Brisanje (engl. *delete*) - oslobađanje mjesta na disku
- Otvaranje (engl. *open*) - dohvaćanje datotečnih atributa i memorijskih adresa na disku
- Zatvaranje (engl. *close*) - oslobađanje prostora u internoj tablici datoteka
- Čitanje (engl. *read*) - čitanje bajtova prema poziciji
- Pisanje (engl. *write*) - pisanje teksta na određenoj poziciji
- Dodavanje (engl. *append*) - dodavanje teksta na kraj datoteke
- Pretraživanje (engl. *seek*) - dohvaćanje pokazivača na datoteku na točno određenoj poziciji
- Dohvaćanje atributa (engl. *get attributes*) - dohvaćanje atributa prikazanih u Tablici 1
- Postavljanje atributa (engl. *set attributes*) - postavljanje atributa prikazanih u Tablici 1
- Promjena imena (engl. *rename*) - stvaranje kopije s novim imenom ili preimenovanje stare datoteke

1.2. Direktoriji

Za praćenje datoteka, datotečni sustavi normalno koriste direktorije ili mape (engl. *folders*), koji su također po strukturi datoteke. Najjednostavnija struktura sustava direktorija je tzv. jednorazinska struktura (engl. *single-level directory systems*) u kojoj jedan direktorij sadrži

sve datoteke. On se obično naziva korijenskim direktorijem (engl. *root*). U modernim operacijskim sustavima koji sadrže tisuće datoteke, jednorazinska struktura je neefikasna za pretraživanje. Stoga je razvijen hijerarhijski datotečni sustav (engl. *hierarchical directory system*) u kojem su povezane datoteke grupirane zajedno. S ovakvim pristupom, postoji više direktorija, a svaki korisnik sadrži svoju privatnu hijerarhiju koja se nadovezuje na korijenski direktorij.

1.2.1. Putanje

Kada je datotečni sustav organiziran u stablo direktorija, koriste se putanje za definiranje imena datoteka. U pravilu postoje dvije vrste putanja: **apsolutna** i **relativna**. Apsolutna putanja sadrži putanju od korijenskog direktorija do tražene datoteke. Relativna putanja je vezana uz koncept radnog direktorija (engl. *current/working directory*). Korisnik može označiti jedan direktorij radnim direktorijem, a u tom slučaju sve putanje koje ne započinju od korijenskog direktorija su relativne u odnosu na radni direktorij.

1.2.2. Operacije s direktorijima

Najčešće operacije s direktorijima su:

- Kreiranje - kreiranje praznog direktorija
- Brisanje - brisanje praznog direktorija
- Otvaranje - čitanje sadržaja direktorija
- Zatvaranje - oslobađanje prostora u internoj tablici datoteka
- Čitanje - dohvaćanje iduće datoteke u otvorenom direktoriju
- Promjena imena - stvaranje kopije s novim imenom ili preimenovanje stare datoteke
- Povezivanje (engl. *link*) - prikazivanje iste datoteke u više direktorija
- Prekidanje veze (engl. *unlink*) - brisanje datoteke iz nekih direktorija

1.3. Implementacije datotečnog sustava

Korisnike operacijskog sustava u suštini zanimaju samo imena datoteka i izvršavanje operacija nad njima. S implementacijske strane, operacijski sustavi imaju svoju programsku implementaciju i pohranjuju se na disk. Vjerojatno najvažniji problem u implementaciji pohrane datoteka jest praćenje koji diskovni blokovi pripadaju kojoj datoteci, tj. problem alokacije memorije diska. U različitim operacijskim sustavima koriste se različite metode alokacije:

- Neprekidna alokacija (engl. *contiguous allocation*) - najjednostavnija alokacijska shema za pohranjivanje svake datoteke kao neprekidan niz blokova na disku. Primjerice, na disku s blokovima veličine 1 KB, za datoteku veličine 50 KB se alocira 50 neprekidnih blokova. Prednosti ovakve alokacije su jednostavnost implementacije te visoke performanse (slijednog) čitanja. Nedostatak je što tijekom vremena disk postaje fragmentiran, tj. brisanjem datoteka nastaju rupe u memoriji.
- Alokacija povezanog popisa (engl. *linked-list allocation*) - svaka datoteka je povezani popis blokova diska. Prednost ovakve alokacije je što je svaki blok diska iskorišten i ne dolazi do fragmentacije. Nedostatak je kompleksnija implementacija i spori pristup datotekama.
- Alokacija povezanog popisa pomoću tablice u memoriji - oba nedostatka alokacije povezanog popisa se mogu eliminirati postavljajući pokazivače na blokove diska u tablicu u memoriji. Takva tablica se naziva tablica alokacije datoteka (engl. *file allocation table (FAT)*). Na taj način se može kretati kroz povezani popis, ali u memoriji, što je znatno brže nego referenciranje na adrese na disku.

- I-čvorovi (engl. *I-nodes*) - svakoj datoteci se asocira struktura podataka zvana I-čvor, koja sadrži atribute te datoteke i adrese njenih blokova na disku. Velika prednost ovakve alokacije je što ova struktura podatka mora biti u memoriji samo u trenutku otvaranja datoteke, pa stoga zauzima manje radne memorije.

Datotečni sustavi operacijskih sustava UNIX/Linux te Windows NTFS koriste I-čvorove kao metodu alokacije memorije diska.

2. DATOTEČNI SUSTAV OPERACIJSKOG SUSTAVA LINUX

2.1. Generalni pregled datotečnog sustava

Inicijalni datotečni sustav u Linuxu bio je MINIX 1 datotečni sustav, koji je podržavao nazive datoteka dugačke 14 znakova i bio veličine 64 MB. Prvo poboljšanje je bio *ext* datotečni sustav, a podržavao je 255 znakova i bio veličine 2 GB. Trenutna unaprijeđena verzija je *ext4*, koji omogućuje najveće datoteke i najbrži rad u odnosu na svoje prethodnike. Jednostavna rečenica koja dosljedno opisuje UNIX sustave, a također je primjenjiva i na Linux je: „*On a UNIX system, everything is a file; if something is not a file, it is a process.*“. Linuxov datotečni sustav ne pravi razliku između datoteka i direktorija, programa, usluga, tekstualnih datoteka, slika i sl. Štoviše, ulazni i izlazni uređaji te svi uređaji generalno se smatraju datotekama u datotečnom sustavu.

Međutim, određene datoteke se ipak razlikuju i nose određene specifičnosti. Većina datoteka se nazivaju **regularnim** datotekama, a sadrže normalne korisničke podatke (tekst, izvršne programe, ulaz ili izlaz programa...). Ostale specijalne datoteke u Linuxovom datotečnom sustavu su prikazane u Tablici 2.

Tablica 2. Vrste datoteka u datotečnom sustavu Linux

Simbol	Značenje	Opis
-	Regularne datoteke	Sadrže normalne korisničke podatke.
d	Direktoriji	Datoteke koje su liste drugih datoteka.
l	Poveznice (engl. *links*)	Datotečni sustav čini neke datoteke ili direktorije vidljivim u više dijelova hijerarhije stabla, a za to koristi linkove.
c	Specijalne datoteke	Mehanizam za ulaz i izlaz. Većina specijalnih datoteka je u */dev* direktoriju.
s	Sockets	Specijalna vrsta datoteke, slična TCP/IP socketima, a omogućuje međuprocesno umrežavanje zaštićeno kontrolom datotečnog sustava.
p	Cjevovodi	Ponašaju se kao socketi i pružaju način međuprocesne komunikacije, no bez mrežne povezanosti.
b	Blok uređaji	Uređaji za pohranu podataka koji podržavaju čitanje i pisanje podataka u blokovima fiksne veličine.

Naredbom *ls –l* možemo dobiti prikaz sadržaja direktorija u dugačkom obliku, u kojem na prvom mjestu stoje vrste datoteka i prava pristupa.

2.2. Particioniranje datotečnog sustava

Prije nego objasnimo hijerarhiju datotečnog sustava u operacijskom sustavu Linux, potrebno je razumjeti pojam particioniranja (engl. *partitioning*) i njegovu upotrebu u Linuxu. Particioniranje diska (engl. *disk partitioning*) je postupak kreiranja jedne ili više regije na sekundarnoj pohrani, kako bi se svaka regija mogla upravljati pojedinačno. Te regije se nazivaju particijama. Particioniranje je tipičan prvi korak prilikom instaliranja novog diska, prije stvaranja datotečnog sustava. Disk pohranjuje informacije o lokacijama i veličinama particija u tzv. tablicu particija (engl. *partition table*) koje operacijski sustav čita prije bilo kojeg dijela diska. Svaka particija operacijskom sustavu predstavlja odvojen logički disk koji koristi dio stvarnog diska. Sistemski administratori koriste program zvan uređivač particija

(engl. *partition editor*) za kreiranje, modifikaciju, brisanje i manipuliranje particijama. Koncept particija omogućuje korištenje različitih datotečnih sustava na istom disku i olakšanu izradu sigurnosnih kopija. Linux koristi više od jedne particije na istom disku, čak pri standardnoj instalacijskoj proceduri. Jedan od ciljeva particija je postizanje veće podatkovne sigurnosti u slučaju katastrofe. Dijeleći tvrdi disk u particije, podaci se mogu grupirati i odvojiti. Uslijed katastrofe, samo se podaci u particiji oštećuju, dok podaci na drugim particijama vjerojatno preživljavaju.

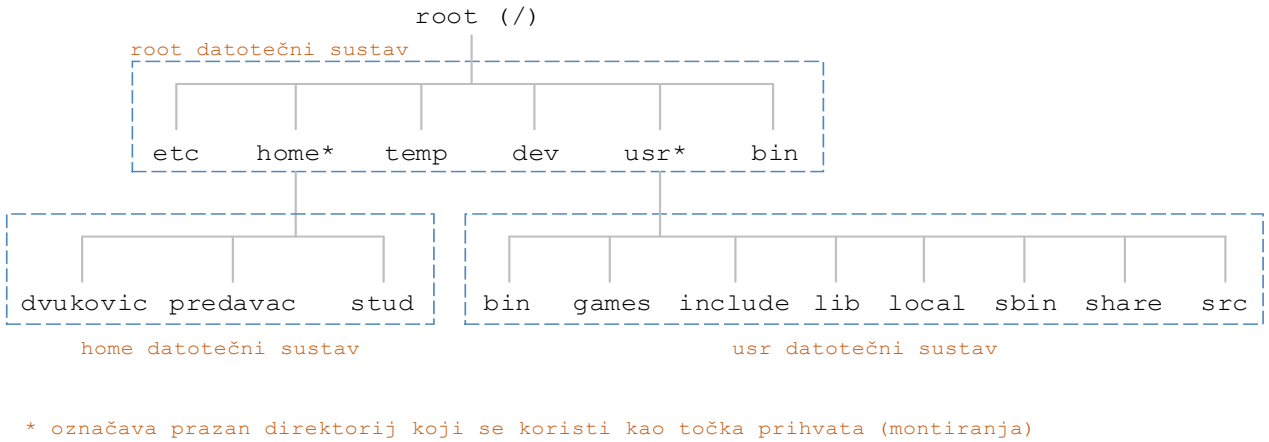
Postoje dvije vrste velikih particija u Linxu:

- **Podatkovne particije** – normalni Linux sistemski podaci, uključujući korijensku particiju koja sadrži sve podatke za pokretanje i izvršavanje operacijskog sustava
- **Particija za izmjenu** (engl. *swap partition*) – proširenje fizičke memorije računala; dodatna memorija na tvrdom disku

Većina sustava sadrži korijensku particiju, jednu ili više podatkovnih particija te jednu ili više particija za izmjenu. Standardna korijenska particija (označena s /) je veličine oko 100-500 MB i sadrži konfiguracijske datoteke sustava, osnovne naredbe i poslužiteljske programe, sistemske biblioteke, privremeni prostor i *home* direktorij administratorskog korisnika. Standardna instalacija zahtijeva oko 250 MB za korijensku particiju. Particija za izmjenu je sakrivena od normalnih korisničkih operacija, a služi za osiguravanje normalnog korisničkog rada. Zbog ove dodatne memorije, u slučaju popunjenih podatkovnih particija, dolazi do izmjene podataka u taj prostor. Generalno, veličina particije za izmjenu kod Linuxa je otprilike dvostruke veličine u odnosu na radnu memoriju. Primjerice, računalo s 1 GB radne memorije (RAM) ima 2 GB veličine particije za izmjenu.

Jezgra operacijskog sustava je zbog svoje važnosti na posebnoj */boot* particiji. Ostatak tvrdog diska je generalno podijeljen u podatkovne particije i to najčešće u: particiju za korisničke programe (*/usr*), particiju za korisničke podatke (*/home*), particiju za privremene podatke (*/var*) te particiju za softver trećih strana (*/opt*).

Sve particije u sustavu se povezuju preko **točke montiranja** (engl. *mount point*). Točka montiranja definira položaj određenih podataka u datotečnom sustavu. Obično, sve particije su povezane preko *root* particije (/), kako je prikazano na Slici 2. Na slici je prikazan primjer logičke podjele tvrdog diska s osnovnom particijom (root), koja je neophodna, i dvije dodatne particije (home i usr) koje su pridodane osnovnoj particiji prilikom pokretanja operacijskog sustava u prazne direktorije home i usr i čine jedinstveni datotečni sustav za korisnika. Particije se povezuju putem **mount** naredbe. Prilikom pokretanja operacijskog sustava, sve particije se povezuju kako je definirano u datoteci */etc/fstab*.



Slika 2. Prikaz povezivanja particija

Informacije o particijama i točkama montiranja se mogu prikazati pomoću **df** naredbe.

df
(*report filesystem disk space usage* – daje informacije o upotrebi diskovnog prostora)

Naredba **df** koristi se za prikazivanje zauzetosti diskovnog prostora po particijama. U primjeru 1, dvije particije se nalaze na dva fizička diska sda2 i sdb1. Prvo slovo particije označava vrstu diska (SCISI/SAS/SATA) a brojke označavaju particiju na fizičkom disku. Prema konvenciji, IDE diskovi dobivaju imena /dev/hda do /dev/hdd, pri čemu je HDD a prvi, a HDD c treći disk.

Primjer 1: Korištenje naredbe df

```
student00@linux:~$ df
Filesystem          1K-blocks      Used Available Use% Mounted on
/dev/sda2            17061616   1104664  15090248   7% /
tmpfs                  258316         4    258312   1% /dev/shm
/dev/sdb1            17559988     47700  17512288   1% /home
```

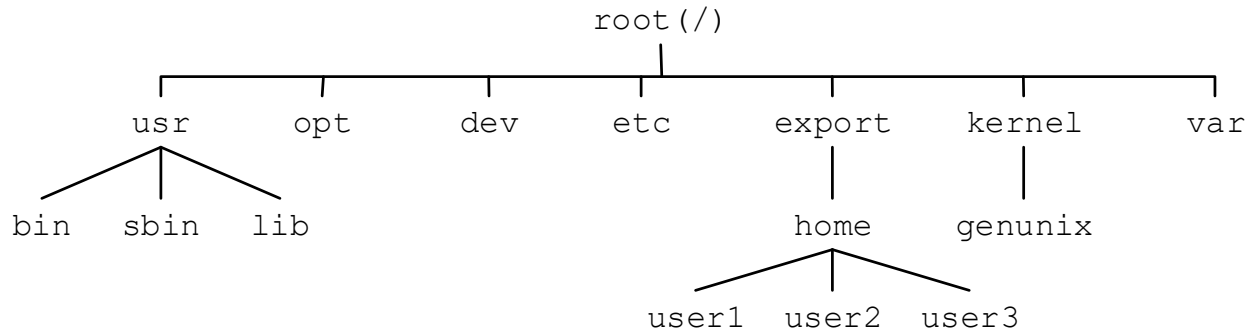
du
(*estimate file space usage* – daje informacije o zauzetosti datotekama)

Naredba **du** prikazuje zauzetost diskovnog prostora datotekama i direktorijima. Opcija –s nalaže sumarni prikaz, a opcija –k prikaz u blokovima od 1 KB. Argument naredbe može biti bilo koji direktorij ili datoteka.

2.3. Hijerarhija datotečnog sustava

Datoteka u Linuxu je slijed 0 ili više bajtova koji predstavljaju proizvoljnu informaciju. U Linuxu ne postoji razlika između ASCII, binarnih ili bilo koje vrste datoteka. Imena datoteka su ograničena na 255 znakova, a svi ASCII znakovi (osim NUL) su dozvoljeni. Prema konvenciji, imena datoteka se sastoje od naziva i ekstenzije koje operacijski sustav implementacijski ne provodi, tj. Linux ne razlikuje ekstenzije od naziva. Datoteke mogu biti grupirane u direktorije koji su također pohranjeni kao posebne datoteke. Direktoriji mogu sadržavati i poddirektorije, čime se stvara hijerarhijska struktura. Hijerarhijska struktura datotečnog sustava operacijskog sustava Linux je prikazana na Slici 3, a za njega vrijedi sljedeće:

- Korijenski direktorij (engl. *root*) datotečnog sustava je početak datotečnog sustava na disku, označava se sa oznakom '**/**',
- ispod korijenskog direktorija nalaze se ostali direktoriji,
- unutar direktorija se nalaze poddirektoriji (direktorij koji se nalazi unutar drugog direktorija).



Slika 3. Hijerarhijska struktura datotečnog sustava Linux

Uobičajeni UNIX/Linux direktoriji su:

- /bin (binary) – sadržava naredbe OS-a
- /sbin (single user binaries) – sadrži osnovne izvršne programe za pokretanje i oporavak sustava
- /boot (Linux) – sadržava datoteke nužne za pokretanje Linux sustava
- /kernel (UNIX) - sadrži osnovni OS (kernel)
- /usr (user) – sadržava izvršne naredbe, sistem-administratorske programe i rutine
- /lib (library) – sadrži zajedničke biblioteke datoteka
- /opt (optional) – sadržava nestandardne programe i programe trećih proizvođača
- /dev (devices) – sadržava datoteke koji su pokazivači na uređaje. Sukladno principu „sve je datoteka“
- /etc (etcetera) – sadržava većinu konfiguracijskih datoteka sustava
- /home (Linux) /export/home (UNIX) – sadrži korisničke direktorije
- /mnt (mount) – standardno mjesto priključenja datotečnih sustava kod Linux-a
- /proc (process) – sadrži datoteke vezane za informacije o sustavu kod Linux-a
- /var (variable) – sadrži dinamičke i promjenjive datoteke (print spooling, mail datoteke, log datoteke…)
- /tmp (temporary) – sadrži privremene datoteke koje se nakon upotrebe brišu

Dva su tipa putanja u datotečnom sustavu operacijskog sustava Linux:

- apsolutna putanja: određuje položaj datoteke ili direktorija u odnosu na cijeli datotečni sustav, uvijek započinje s root (‘/’) direktorijem
- relativna putanja: određuje položaj datoteke ili direktorija u odnosu na trenutni direktorij, pristup datoteci ili direktoriju u tekućem (radnom) direktoriju jednostavnim navođenjem imena. Ne započinje s ‘/'.

2.3.1. Varijabla okoline *PATH*

Kada u operacijskom sustavu Linux pokrećemo neku uobičajenu naredbu (npr. *ls*), gotovo nikada nije potrebno pružiti punu putanju do naredbe. Pri tome je bitno razumjeti da se pri izvršavanju neke naredbe ljuske pokreće njena izvršna datoteka, koja zapravo predstavlja kompajlirani program napisan u programskom jeziku C i asembleru koji potencijalno uključuje sustavske pozive operacijskog sustava. Primjerice, izvršna datoteka naredbe *ls* nalazi se u datoteci ***/bin/ls***. Drugim riječima, pokretanje naredbe *ls* zapravo podrazumijeva pokretanje programa, koji je kompajliran u izvršnu datoteku istog imena i smješten u direktorij *bin*.

Kako bi se izbjeglo pisanje pune putanje do izvršne datoteke prilikom korištenja svake naredbe, operacijski sustav preko varijable okoline *PATH* vodi računa o mapiranju naredbi i njihovih izvršnih datoteka. Ova varijabla izlistava direktorije u sustavu koji sadrže izvršne

datoteke, te na taj način oslobađa korisnika od upisivanja i pamćenja dugačkih naredbi. Naredbom **echo** možemo ispisati sadržaj varijable *PATH*, kako je pokazano u primjeru 2. U primjeru, direktoriji */usr/local/bin*, */usr/bin*, */bin/usr/local/games* te */user/games* se slijedno pretražuju kako bi se pronašla izvršni program upisane naredbe. Ako se nakon instalacije novog programa želi izbjeći pisanje pune putanje do njegove izvršne datoteke, moguće ju je dodati u varijablu okoline *PATH* i pokretati putem skraćenog naziva. Primjer 3 prikazuje način dodavanja nove putanje u varijablu *PATH*.

Primjer 2: Ispis sadržava varijable PATH
student00@linux:~$ **echo** $PATH
/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games

Primjer 3: Dodavanje nove putanje u varijablu okoline *PATH*
student00@linux:~$ **export** PATH=$PATH:/usr/local/bin/python

2.3.2. Varijabla *HOME*

Svaki korisnik se nakon povezivanja na operacijski sustav nalazi u unaprijed zadanom *home* odredištu, koje je specifično baš za tog korisnika. U većini slučajeva ono se nalazi u direktorij */home*, iako to nije nužno pravilo. Vaš se *home* direktorij može nalaziti i na tvrdom disku udaljenog poslužitelja. U svakom slučaju, administrator sustava definira mapiranja između korisnika i njihovih *home* direktorija, a sistemska varijabla *HOME* sadrži listu svih mapiranja. Naredbom **echo** možemo ispisati sadržaj varijable *HOME*, kako je pokazano u primjeru 4. Na putanju koja je sadržana u varijabli *HOME* datotečni sustav smješta korisnika nakon pokretanja naredbe **cd ~**.

Primjer 4: Ispis sadržaja varijable HOME
student00@linux:~$ **echo** $HOME
/home/stud/student00

2.4. Manipuliranje datotekama

U prvoj laboratorijskoj vježbi pokazane su osnovne naredbe ljuske operacijskog sustava Linux za kretanje kroz datotečni sustav te za rad s datotekama i direktorijima. Tako su pokazani primjeri korištenja naredbi: **pwd**, **cd**, **ls**, **touch**, **file**, **cat**, **more**, **cp**, **mv**, **mkdir**, **rm** te **rmdir**. Preko tih naredbi izvršava se kreiranje, brisanje, premještanje i čitanje datoteka i direktorija. U nastavku će se prikazati malo naprednije naredbe za čitanje, pretraživanje i povezivanje datoteka.

2.4.1. Čitanje sadržaja datoteke

Osim naredbe **cat**, koja samo šalje datoteke na standardni izlaz, postoji još nekoliko naredbi za čitanje sadržaja datoteke. Uobičajeni način čitanja datoteke danas je preko grafičkih alata. Ipak, čitanje je moguće i unutar alata naredbenog retka (konzole). Primjeri 5, 6 i 7 pokazuju takve naredbe.

Primjer 5: Upotreba naredbe head
student00@linux:~$ **head -2 head_tail**
1. red
2. red

Primjer 6: Upotreba naredbe tail
student00@linux:~$ **tail -3 head_tail**
13. red
14. red
15. red

Primjer 7: Upotreba naredbe wc
student00@linux:~$ **wc head_tail**
 15 30 111 head_tail
student00@linux:~$ **wc -l jedan**
3 jedan

2.4.2. Pretraživanje datotečnog sustava

Postoji nekoliko naredbi za pretraživanje sadržaja datotečnog sustava. No, prije proučavanja samih naredbi, bitno je spomenuti **metaznakove** za pretraživanje. Tri najčešće korištena metaznaka su:

- Zvjezdica (engl. *asterisk*) * - zamjenjuje nula ili više znakova osim vodeće točke kod skrivenih datoteka
- Upitnik ? – zamjenjuje jedan znak osim vodeće točke kod skrivenih datoteka
- Uglate zagrade [] – zamjenjuju jedan iz raspona ili jedan iz skupa zadanih znakova koji mogu biti na jednoj poziciji znaka. Znakovi unutar zagrada ne moraju biti poredani. Kada se definira raspon znakovi moraju biti u redoslijedu. Znak '-' definira raspon. Primjer: [a-z] traži bilo koji znak u obliku malih slova između a i z uključujući i ta dva znaka. Znakovi su osjetljivi na veličinu.
- Točka-zarez ; - omogućuje pisanje više naredbi u jednom retku bez potrebe korištenja tipke Enter, stavlja se znak ';' između naredbi.

Kada je potrebno dohvatiti datoteke unutar direktorija koji odgovaraju određenim pravilima, moguće je kombinirati naredbu **ls** i metaznakove, kako je pokazano u primjeru 8.

Primjer 8: Korištenje metaznakova
student@linux:~$ **ls**
a.txt a1.txt b.txt c.txt dir1 dir2 test.txt
student@linux:~$ **ls a***
a.txt a1.txt
student@linux:~$ **ls ?.txt**
a.txt b.txt c.txt
student@linux:~$ **ls [a-c].txt**
a.txt b.txt c.txt
student@linux:~$ **date; ls -ld dir***
Sat Mar 3 13:21:36 CET 2007
drwxr-xr-x 4 student stud 4096 Mar 3 12:43 dir1
drwxr-xr-x 2 student stud 4096 Mar 3 12:43 dir2

Osim kombinacija naredbe **ls** i metaznakova, postoje specijalizirane naredbe za pretraživanje.

find
(*search for files in a directory hierarchy* – pretražuje datoteke u hijerarhijskoj strukturi direktorija)

Naredba **find** pronalazi datoteke temeljem postavljenih kriterija i opcionalno izvršava naredbe na pronađenim datotekama. Sintaksa naredbe: **find [path] [expression (action)]**

- ***putanja*** definira mjesto u datotečnoj strukturi odakle započinje pretraga
- ***izraz*** je jedan ili više kriterija za pretragu koji opisuje što treba tražiti
- ***akcija*** predstavlja neobveznu naredbu koja se može izvršiti na pronađenim datotekama

Tablica 3. Izrazi za pretraživanje u naredbi find

Izraz za pretraživanje	Značenje	Definicija
-**name filename**	Naziv datoteke	Traženje svih datoteka čiji naziv odgovara traženom. Dozvoljeni su i metaznakovi (* i ?), ali se interpretiraju doslovno osim ako nisu omeđeni navodnicima.
-type filetype	Tip datoteke	Traženje svih datoteka čiji tip odgovara traženom (d = direktorij).
-mtime [+/-]n	Vrijeme izmjene	Traženje svih datoteka čije vrijeme izmjene je jednako, starije (+) ili je novije (-) od **n** dana.
-atime [+/-]n	Vrijeme pristupa	Traženje svih datoteka čije vrijeme pristupa je jednako, starije (+) ili je novije (-) od **n** dana.
-user loginID **-group groupID**	Korisnički ID ili ID grupe	Traženje svih datoteka čiji vlasnik odgovara identifikacijskom broju korisnika (korisnički ID) ili grupe (ID grupe)
-perm mod	Prava pristupa	Traženje svih datoteka čija prava pristupa odgovaraju traženima (samo oktalne oznake)
-size [+|-]n[c]	Veličina	Traženje svih datoteka čija veličina odgovara, veća je od (+) ili je manja od (-) **n**. **n** predstavlja blokove od 512 bajtova ili znakova ako se iza veličine nalazi **c**.

grep
(*print lines matching a pattern* – ispisuje retke datoteke koji zadovoljavaju uzorak)

Naredba **grep** pretražuje tekst unutar datoteke ili unutar izlaza neke naredbe. Radi sa riječima ili regularnim izrazima. Sintaksa naredbe: **grep [option] string filename:**

- string može biti znak, riječ ili rečenica
- string sa razmakom ili specijalnim znakovima mora biti postavljen u navodnike
- opcija *–i* ignorira velika/mala slova
- opcija *–v* ispisuje retke koji ne zadovoljavaju string
- opcija *–n* ispisuje broj retka u kojem je pronađen string

Naredba **grep** ima svoje proširene verzije: **egrep** (*extended grep*) i **fgrep** (*fast grep*). Prva naredba proširuje naredbu **grep** pomoću regularnih izraza, a druga ubrzava pretragu jer ne prihvaća regularne izraze. Primjeri 9 i 10 pokazuju upotrebu naredbi **find** i **grep**.

Primjer 9: Korištenje naredbe find
student00@linux:~$ **find . -name** "datoteka*"
./datoteka1.txt
./datoteka2.txt
student00@linux:~$ **find . -size** 10k
./file/probni.tar

Primjer 10: Korištenje naredbe grep
student00@linux:~$ **grep 2 -n head_tail**
2:2. red
12:12. red
student00@linux:~$ **ls -la |grep** linux
-rw-r--r-- 1 student stud 2819 Mar 26 08:39 linux.txt
-rw-r--r-- 1 student stud 4134 Mar 26 08:39 linux1.txt

2.4.3. Povezivanje datoteka

Poveznica (engl. *link*) je način povezivanja dva ili više imena datoteka na isti skup datoteka. U datotečnom sustavu operacijskog sustava Linux postoje dvije vrsta linkova:

- **Teški link** (engl. *hard link*) – povezuje dva ili više imena datoteka na istu datoteku na tvrdom disku. Teški linkovi dijele iste podatkovne blokove na tvrdom disku, ali se ponašaju kao neovisne datoteke. Glavni nedostatak teških linkova je nemogućnost proširenja preko više particija, jer su podatkovni blokovi jedinstveni za particiju.
- **Slabi link** (engl. *soft/symbolic link*) – mala datoteka koja pokazuje na drugu datoteku, još se naziva i simbolični link. Sadrži putanju do tražene datoteke, umjesto njene fizičke lokacije na tvrdom disku. S obzirom da nije vezan uz fizičku reprezentaciju datoteke u memoriji, simbolički link se može proširiti preko više particija.

Uklanjanje ciljane datoteke čini simbolički link beskorisnim. U principu, svaka regularna datoteka je ujedno teški link. Naredba za stvaranje linkova je **ln**.

ln
(*make links between files* – napravi poveznicu između datoteka)

Naredba **ln** se koristi za stvaranje linka (poveznice) prema datoteci. Sintaksa naredbe: **ln filename [option] new_filename.** Koliko poveznica postoji na nekoj datoteci može se vidjeti naredbom ls –l. Uvijek je minimalno jedna poveznica – tekući direktorij ('.'). Poveznica se može obrisati kao bilo koja datoteka naredbom rm:

- **rm new_filename** će obrisati samo poveznicu, ne i originalnu datoteku
- **rm filename** će obrisati datoteku i sve poveznice na nju.

Opcija –s uz naredbu **ln** kreira simboličku poveznicu umjesto čvrste.

2.5. Sigurnost datotečnog sustava

Sigurnost datotečnog sustava operacijskog sustava Linux temelji se na pravima koja određuju koji korisnici mogu pristupiti pojedinim datotekama i direktorijima datotečnog sustava i što s njima mogu raditi. Za prikaz dopuštenja u datotečnom sustavu koristi se izlistavanje u dugom obliku, tj. naredba **ls –l**. Dugi oblik započinje sa strukturom prikazanom na Slici 4. Dopuštenja datoteka (engl. *file permissions*) se sastoje od sljedećih kategorija polja:

- Tip datoteke (engl. *file type*) – prikazani u Tablici 1. Najčešće '-' za datoteke ili 'd' za direktorij

- Vlasnik datoteke (engl. *owner*) – korisnik koji je kreirao datoteku ili ima datoteku u vlasništvu
- Grupa (engl. *group*) – grupa koja ima prava nad datotekom
- Ostali (engl. *others*) – svi ostali koji nisu vlasnik datoteke i ne pripadaju grupi koja ima prava nad datotekom, a imaju prava pristupa sustavu.

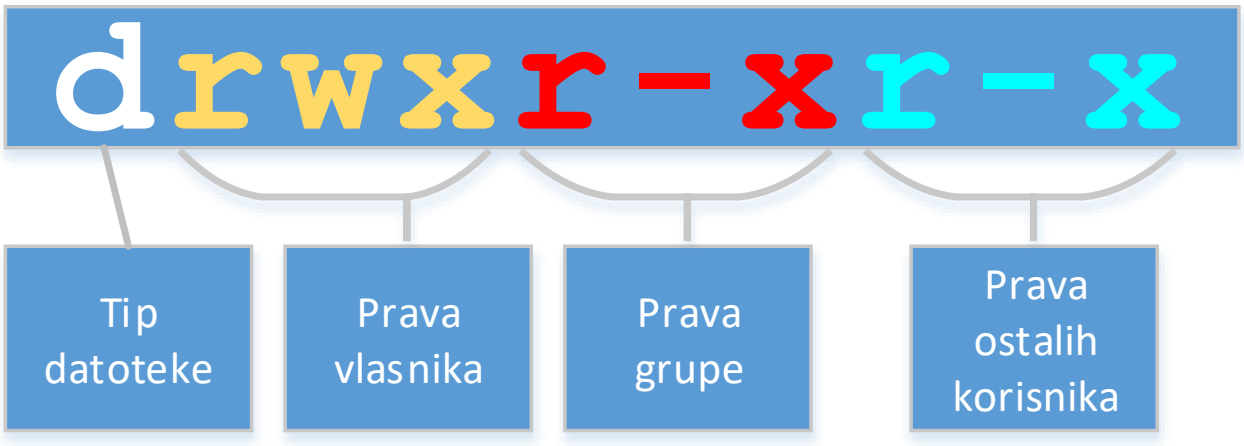
Svaka od kategorija prava (vlasnik, grupa, ostali) ima tri moguća prava pristupa:

- Čitanje (engl. ***r**ead*) – označeno slovom 'r'
- Pisanje (engl. ***w**rite*) – označeno slovom 'w'
- Izvršavanje (engl. *e**x**ecute*) – označeno slovom 'x'

Oznaka '-' ukazuje da je taj oblik pristupa uskraćen. Da bi se pristupilo direktoriju 'r' i 'x' prava moraju biti postavljena (za izlistavanje sadržaja direktorija dovoljno je 'r'). Da bi se pristupilo datoteci mora biti postavljeno pravo 'r'. Mijenjanja vlasništva nad datotekom se izvršava pomoću naredbi **chown** i **chgrp**. Prva naredba mijenja vlasnika datoteke i grupu, a druga mijenja vlasništvo grupe. Samo korisnik s administratorskim ovlastima ima mogućnost izvršavanja tih naredbi.

Primjer 11: Korištenje naredbi chown i chgrp

```
predavac@linux:~$ touch prava
predavac@linux:~$ ls -l |grep prava
-rw-r--r--  1 root    root    0 Apr  1 10:56 prava
predavac@linux:~$ chown stud prava
predavac@linux:~$ ls -l |grep prava
-rw-r--r--  1 stud    root    0 Apr  1 10:56 prava
predavac@linux:~$ chgrp stud prava
predavac@linux:~$ ls -l |grep prava
-rw-r--r--  1 stud    stud    0 Apr  1 10:56 prava
predavac@linux:~$ chown root:root prava
predavac@linux:~$ ls -l |grep prava
-rw-r--r--  1 root    root    0 Apr  1 10:56 prava
```



Slika 4. Struktura dopuštenja datoteka u operacijskom sustavu Linux

Prilikom kreiranja nove datoteke i novog direktorija uobičajeno se postavljaju prava pristupa: **-rw-r--r--** (nova datoteka) i **drwxr-xr-x** (novi direktorij). Nova prava su detaljnije pojašnjena u Tablici 4.

Tablica 4. Uobičajena prava pristupa datotekama i direktorijima nakon kreiranja

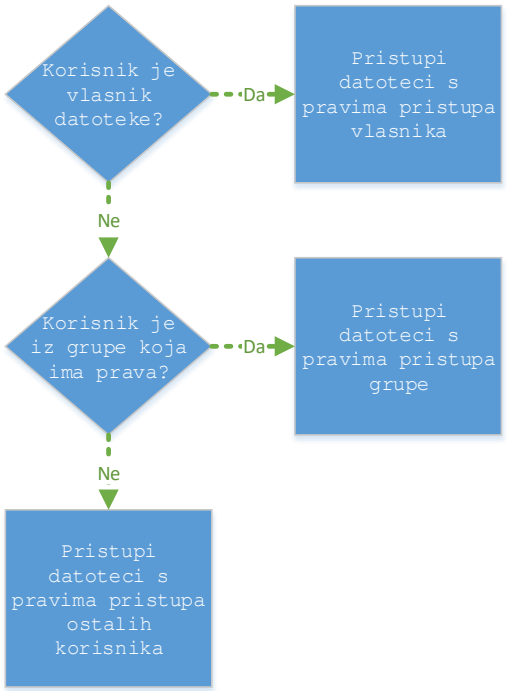
Kategorija korisnika	Korisnik (vlasnik)	Grupa	Ostali
Nova datoteka	Čitanje/Pisanje (r w -)	Čitanje (r - -)	Čitanje (r - -)
Novi direktorij	Čitanje/Pisanje/Izvršavanje (r w x)	Čitanje/Izvršavanje (r – x)	Čitanje/Izvršavanje (r – x)

U dugom izlistanju sadržaja direktorija, imena korisnika i grupe su predstavljeni identifikatorima korisnika (engl. *user identifier (UID)*) i grupe (engl. *group identifier (GID)*). Naredba **ls -n** prikazuje UID i GID umjesto njihovih tekstualnih naziva, kako je prikazano u primjeru 12.

Primjer 12: Korištenje naredbe ls

```
student00@linux:~$ ls -l
-rw-r--r--  1 student stud 2819 Mar 31 10:33 linux.txt
student00@linux:~$ ls -n
-rw-r--r--  1 2308 1001 2819 Mar 31 10:33 linux.txt
```

Naredba **id** prikazuje numerički i tekstualni UID i GID aktivnog korisnika. Naredba **groups** prikazuje sve grupe kojima korisnik pripada. S obzirom da korisnik istovremeno može pripadati različitim kategorijama koje imaju različite ovlasti nad datotekom, datotečni sustav operacijskog sustava Linux slijedi algoritam za određivanja korisničkih prava prikazan na slici 5.



Slika 5. Algoritam za određivanje korisničkih prava na datoteci/direktoriju

Normalna posljedica primjene strogih dopuštenja nad datotekama je da će se prava pristupa mijenjati iz nekih razloga. Za mijenjanje prava pristupa nad datotekom koristi se naredba **chmod**.

chmod
(*change file access permissions* – mijenja pristupna prava na datoteci)

Naredba **chmod** se koristi za mijenjanje prava pristupa datoteci od strane vlasnika ili root-a. Naredba može raditi u simboličkom (relativnom) i oktalnom (apsolutnom) načinu rada:

- Simbolički mod:
  - Mijenja prava pristupa relativno u odnosu na trenutna prava
  - Sintaksa: **chmod mod filename**
    - mod se sastoji od tri dijela
    - tko - kategorija korisnika na koju se odnosi naredba (**u - user, g - group, o - other, a - all**)
    - operator - operator koji se koristi (=, +, -)
    - pravo - pravo koje se postavlja (=), dodaje (+) ili oduzima (-): **r, w, x**
- Oktalni mod:
  - Numerički mijenja prava na datoteci za sve kategorije istovremeno
  - Sintaksa: **chmod octal_mod filename** ili **chmod -R octal_mod directory_name**

Značenja oktalnih vrijednosti za prava pristupa su prikazana u Tablici 5. Detaljnija razrada svih mogućih oktalnih vrijednosti je prikazana u Tablici 6. Primjer 13 prikazuje upotrebu naredbe **chmod**.

Tablica 5. Vrijednosti oktalnog moda za osnovna prava pristupa nad datotekom/direktorijem

Oktalna vrijednost	Prava pristupa
4	Čitanje (r - -)
2	Pisanje (- w -)
1	Izvršavanje (- - x)

Tablica 6. Vrijednosti oktalnog moda za sve kombinacije prava pristupa nad datotekom/direktorijem

Oktalna vrijednost	Zbroj vrijednosti kategorija prava pristupa	Odgovarajuća prava pristupa	Definicija
7	4 + 2 + 1	(r w x)	Čitanje, Pisanje i Izvršavanje
6	4 + 2 + 0	(r w -)	Čitanje i Pisanje
5	4 + 0 + 1	(r - x)	Čitanje i Izvršavanje
4	4 + 0 + 0	(r - -)	Čitanje
3	0 + 2 + 1	(- w x)	Pisanje i Izvršavanje
2	0 + 2 + 0	(- w -)	Pisanje

1	0 + 0 + 1	(- - x)	Izvršavanje
0	0 + 0 + 0	(- - -)	Bez prava pristupa

Primjer 13: Korištenje naredbe chmod

```
student00@linux:~$ ls -l |grep linux.txt
-rw-r--r--  1 student stud 2819 Mar 31 10:33 linux.txt
student00@linux:~$ chmod g+w linux.txt
student00@linux:~$ ls -l |grep linux.txt
-rw-rw-r--  1 student stud 2819 Mar 31 10:33 linux.txt
student00@linux:~$ chmod 545 linux.txt
student00@linux:~$ ls -l |grep linux.txt
-r-xr--r-x  1 student stud 2819 Mar 31 10:33 linux.txt
```

Prilikom kreiranja datoteka i direktorija, oni dobivaju uobičajena prava pristupa, kako je prikazano u Tablici 3. Vrijednosti uobičajenih prava pristupa se računaju iz korisničke **maske** za kreiranje datoteke ili direktorija. Predefinirana vrijednost maske u Linuxu je (0) 022 u oktalnom brojevnom sustavu. Uobičajena prava pristupa nad datotekom ili direktorijem se dobivaju odbijanjem vrijednosti maske od najvećih mogućih vrijednosti prava pristupa u oktalnom modu za direktorije (777) i datoteke (666). Primjerice, nova datoteka ima ovlasti: $622_8 - 022_8 = 644_8$, a direktorij $777_8 - 022_8 = 755_8$. Vrijednost maske se može mijenjati pomoću naredbe **umask**.

umask
(*user file creation mode mask* – korisnička maska za kreiranje datoteke)

Naredba **umask** se koristi za postavljanje predefiniranih ovlasti prilikom kreiranja datoteka i direktorija. Sintaksa naredbe: **umask umask_value.** Samo pozivanje naredbe **umask** vraća trenutnu vrijednost maske.

2.6. Izrada sigurnosnih kopija

Izrada sigurnosnih kopija podataka (engl. *backup*) omogućuje povrat izgubljenih i oštećenih podataka uslijed:

- pogrešaka na tvrdom disku (zbog mehaničkih dijelova diskovi su podložni kvarovima, RAID 0 i 5 su uobičajeni načini zaštite od ovakve vrste kvarova)
- neispravnosti datoteka
- destruktivnog utjecaji (zaposlenici, virusi, hakeri)
- prirodne katastrofe (požar, poplava,…)
- slučajnog brisanja ili prepisivanja datoteka

Tri glavna oblika backupa uključuju puni (eng. *full*), inkrementalni (eng. *incremental*), i diferencijalni (eng. *differential*) backup i najčešće uključuju 5-7 dnevnih magnetskih traka, 4-5 tjednih traka i 12 mjesečnih traka. Tračni uređaji su najčešći oblik medija za izradu arhiva. Značajke traka su veliki kapacitet, sekvencijalni pristup podacima (pisanje i čitanje) i sporost.

tar
(*archiving utility* – program za arhiviranje)

Naredba tar (skraćenica od engl. *Tape Archive*) omogućuje arhiviranje višestrukih datoteka i direktorija datotečnog sustava. Sintaksa naredbe: tar function [option] [out_file] filename. Inicijalno je napravljena za izradu sigurnosne kopije podataka na tračne uređaje, no služi i za

arhiviranje na druge medije (HDD, USB, memorijske kartice,...). Ne komprimira podatke, samo smješta datoteke unutar jedne arhive/datoteke. Preporuča se korištenje .tar nastavka za izlaznu datoteku. U tablicama 7 i 8 su prikazane oznake funkcija i opcija koje se koriste uz naredbu tar.

Tablica 7. Funkcije uz naredbu **tar**

Oznaka funkcije (malim slovom)	Značenje	Funkcija koja se izvodi
c	Izradi/kombiniraj (eng. ***C**reate/**C**ombine*)	Izradi novu **tar** datoteku
t	Sadržaj (eng. ***T**able of contents*)	Prikaži sadržaj tar datoteke
x	Otpakiraj datoteku (eng. *E**x**tract*)	Otpakiraj određene datoteke iz tar datoteke

Tablica 8. Opcije uz naredbu **tar**

Oznaka opcije	Značenje	Funkcija koja se mijenja
f	Naziv datoteke (eng. ***F**ile name*)	Odredi tar datoteku koja se izrađuje ili kao datoteka na tvrdom disku (/tmp/file.tar) ili kao datoteka uređaja na izlaznom uređaju (/dev/xxx), kao što je SD kartica i USB memorija
v	Detalji (eng. ***V**erbose*)	Izvrši u *verbose* modu

Za komprimiranje **tar** arhive se može koristiti naredba **compress** čime se značajno može smanjiti veličina arhive (ovisno o sadržaju). Sažeta datoteka ima isti naziv kao i osnovna, nekomprimirana datoteka uz dodatak nastavka **.Z**. Za dekomprimiranje datoteke/arhive komprimirane naredbom **compress** koristi se naredba **uncompress**. Ostali alati za arhiviranje, komprimiranje i dekomprimiranje u operacijskom sustavu Linux su: **gzip**, **gunzip**, **gzcat**, **jar**, **gtar**, **zip**, **unzip.** Često se primjenjuje arhiviranje datoteka sa istovremenim komprimiranjem **gzip** algoritmom, uobičajena ekstenzija takvih datoteka je **.tgz**

Primjer 14: Korištenje naredbi compress i uncompress

```
student00@linux:~$ ls -l
total 12
-rw-r--r--  1 root root 10240 Apr  5 13:22 arhiva.tar
student00@linux:~$ compress arhiva.tar
student00@linux:~$ ls -l
total 4
-rw-r--r--  1 root root 275 Apr  5 13:22 arhiva.tar.Z
student00@linux:~$ uncompress arhiva.tar.Z
student00@linux:~$ ls -l
total 12
-rw-r--r--  1 root root 10240 Apr  5 13:22 arhiva.tar
```

Primjer 15: Istovremeno arhiviranje i komprimiranje arhive

```
student00@linux:~$ tar -czvf arhiva/arhiva.tgz datoteka1 datoteka2
datoteka1
datoteka2
student00@linux:~$ ls -l arhiva/arhiva.tgz
-rw-r--r--  1 root root 130 Apr  5 13:27 arhiva/arhiva.tgz
```

3. ZADACI ZA SAMOSTALNI RAD

Pristupiti poslužitelju linux.etfos.hr pomoću Putty SSH klijenta. Odgovorite na sljedeća pitanja:

1. Na kojoj particiji se nalazi */home* direktorij? Kolika je ukupna veličina tog direktorija?
2. Pozicionirati se u vaš home (*cd ~*) direktorij. Izlistati sve retke datoteke *head_tail* počevši od 4. retka. Zatim izlistati zadnjih 6 redaka te datoteke.
3. Iz datoteke *head_tail* izlistati sve retke koji ne sadrže broj 5. Prebrojati broj redaka u datoteci *linux.txt*.
4. Koristeći metaznakove za pretraživanje datotečnog sustava, u home direktoriju pronađite sve tekstualne datoteke (završavaju na .txt) koje u svom imenu sadrže brojku 1 i obrišite ih. Zatim iz vašeg home direktorija premjestite sve datoteke koje u imenu sadrže riječ „da“ (ne uzimajući u obzir nastavak) u direktorij *dir1*. Izlistajte sadržaj direktorija sortiran po vremenu kreiranja tako da se najstarije datoteke prikažu prve.
5. Napravite simbolički link na direktorij *dir2* u vašem home direktoriju. Izlistajte sadržaj home direktorija i pokušajte ući u novokreirani link. Zatim napravite još jedan simbolički link na isti direktorij te ponovno izlistajte sadržaj direktorija. Uklonite direktorij *dir2* i provjerite što se dogodilo s kreiranim simboličkim linkovima.
6. Napisati naredbu koja će promijeniti predefiniranu masku na način da će prilikom izrade novog direktorija njegove ovlasti biti 754. Nakon toga napisati naredbu kojom se izrađuje novi direktorij naziva *test_maska*. Napisati naredbu kojom provjeravate prava pristupa nad izrađenim direktorijem. Koja su to prava?
7. Napišite chmod naredbu u oktalnom i simboličkom obliku za dodjelu sljedećih prava:
   a. -rw-r-x-w-
   b. drwx--xrw-

   Prethodno postavljena prava su 422 za datoteku i 741 za direktorij.
8. Kreirajte arhivu proizvoljnog naziva od datoteka *jedan* i *dva*. Komprimirajte arhivu te ju zatim raspakirajte u direktorij *dir2*.

3.1. Zadaci za vježbu

Ove zadatke nije potrebno riješiti tijekom laboratorijske vježbe. Oni služe isključivo za vježbanje za pismeni ispit, a svoja rješenja možete provjeriti na konzultacijama. Programe možete napisati koristeći tekstualne editore (Nano, Vim) ili možete koristiti online okruženja (Ideone, programiz, repl,…). Ukoliko kompajlirate Vaš kod unutar operacijskog sustava Linux, preporučena je upotreba *gcc* kompajlera. Upute za korištenje *gcc* kompajlera se mogu pronaći na Loomen stranici kolegija.

1. U programskom jeziku C implementirajte sljedeće naredbe komandnog sučelja koristeći odgovarajuće sustavske pozive operacijskog sustava Linux te naredbe programskog jezika C za rad s datotekama:
   a. *cat*
   b. *ls*
   c. *mv*

   Opišite koje ste biblioteke i pripadne funkcije koristili i zašto.
2. U programskom jeziku C napravite program koji otvara direktorij unutar operacijskog sustava Linux te izlistava njegov sadržaj na komandno sučelje.
3. U programskom jeziku C napravite program koji putem komandnog sučelja prima naziv datoteke kao parametar te ispisuje sljedeće informacije o datoteci:

a. tip datoteke
b. broj poveznica na datoteku
c. vrijeme zadnjeg pristupa datoteci
d. postavljena prava pristupa za čitanje, pisanje i izvršavanje

4. U nastavku je prikazan sadržaj datoteke *grepdata.txt*. Kreirajte datoteku tog naziva i u nju kopirajte prikazani sadržaj. Zatim izvršite niz *grep* naredbi koje izvode sljedeće funkcionalnosti:
    a. Ispišite sve linije koje sadrže telefonski broj s proširenjem (proširenje je označeno sa slovima x ili X te 4 znamenke).
    b. Ispišite sve linije koje započinju s 3 znamenke nakon kojih slijedi praznina.
    c. Ispišite sve linije koje u sadrže datum.
    d. Ispišite sve linije koje u sebi sadrže riječi u kojima su dva uzastopna samoglasnika jednaka (npr. *eve*, *adam*,…)
    e. Ispišite sve linije koje ne započinju s velikim slovom S.

Sep. 17, 2013
Esperanza High School
1830 N. Kellog Dr.
Anaheim, CA 92807-1281
Steve Marshal
714-555-7870 X7310
aztecwrestling@example.com
Brian Fortenbaugh
714-555-7870 x7309

Sep. 24, 2013
Sonora High School
401 S. Palm St.
La Habra, CA 90631
Carl Hohl (aka Krazy Rabbit)
562-555-9800

Oct. 1, 2013
Lakewood High School
440 Briercrest Ave.
Lakewood, CA 90713-2013
Andy Miramontes
562-555-1281

Oct. 8, 2013
North Torrance High School
2013 W. 182nd
Torrance, CA 90504
Don Henderson
310-555-4412

Oct. 15, 2013
El Dorado High School
1651 N. Valencia Ave.
Placentia, CA 90631
Steve Lawson
714-555-5350 x2134
Lawsonhawk@example.com

Nov. 12, 2013
Rosemead High School
9063 E. Sepulveda Dr.
Rosemead, CA 91770
Daren de Heras
daren103@example.com

4. LITERATURA

[1] Stallings, W., 2012. Operating systems: internals and design principles. Boston: Prentice Hall,.

[2] Tanenbaum, A.S. and Bos, H., 2015. Modern operating systems. Pearson.

[3] https://linuxjourney.com/

[4] http://wiki.tldp.org/

[5] https://tldp.org/LDP/intro-linux/html/index.html

[6] http://evc-cit.info/cit052/index.html