

# 1. Unix/Linux

UNIX je višekorisnički (eng. multiuser) i višezadačni (eng. multitasking) operacijski sustav, kojeg odlikuje visoka kvaliteta, koja se reflektira kroz visoku pouzdanost, stabilnost i sigurnost. Tokom godina, nastajali su i drugi operacijski sustavi bazirani na UNIX-u, od kojih su najpoznatiji Linux i OpenBSD. UNIX, a nakon njega i Linux su se dugo vremena koristili uglavnom na server računalima, ali se upotreba, prvenstveno Linux-a, sve više širi, te polako prodire i na *desktop* tržište. Trenutno u svijetu UNIX i Linux zajedno pokreću većinu internet servera i njihov zajednički udio je u porastu.

Sa korisničke strane, za razliku od Windows sustava gdje se maksimalno pokušava sakriti relativno jednostavna i neefikasna komandna linija, u UNIX-u je ona znatno moćnija te se i dalje puno koristi, jer izuzetno olakšava i pojednostavljuje posao iskusnijim korisnicima. Pored komandne linije (terminalski način rada), korisnik može birati između više grafičkih korisničkih sučelja (eng. GUI - Graphical User Interface), ali će u ovim uputama biti objašnjene samo naredbe koje se koriste na terminalu i mogu se na identičan način koristiti i na UNIX-u i na Linux-u.

Jedna od, od cijelog niza, velikih prednosti UNIX baziranih operacijskih sustava je i to što se svi podaci o konfiguraciji računala (servera, pisača, mreže, programa itd.) zapisuju u obične tekstualne datoteke, što omogućuje jednostavno administriranje sustava i pronalazak uzroka eventualnih pogrešaka u radu, sa lokalnog ili udaljenog računala.

Kada se navedenim karakteristikama sustava doda i činjenica da su hardverski zahtjevi npr. Linux operacijskog sustava (koji je i besplatan) znatno niži od konkurentskih Windows sustava (NT, 2000, XP), postaje jasno da nema niti jednog razloga (osim neznanja ili neupućenosti) zašto bi se netko odlučio da na svoje server računalo stavi skupe i vrlo nepouzdan Windows operacijske sustave. Upravo ta činjenica objašnjava iznimno brz rast udjela Linux operacijskog sustava na tržištu server računala. Detaljnija usporedba ili pregled mogućnosti ovih sustava izlazi iz okvira ovih uputa.

Za primjenu na *desktop* računalima, postoje još neki problemi koji zaustavljaju proboj na to tržište, od kojih su najveći: nedostatak kvalitetnih aplikacija koje postoje za Windows sustav (u nekim područjima primjene je problem prevaziđen, a na mnogim drugima se pojavljuju sve kvalitetnija rješenja) i nemogućnost interoperabilnosti između datoteka kreiranih u različitim programima (sve veći broj programa ipak podržava najčešće korištene formate).

Iako je i u prošlosti bilo različitih pokušaja proboja novih i konkurentskih ideja i tehnologija, čije je mnoge tvorce Microsoft, zloupotrebom svog monopolističkog položaja finansijski uništavao, u slučaju Linux-a, takav scenarij nije moguć, jer je Linux zasnovan na sasvim drugim principima te ga se finansijski ne može uništiti.

Iz ovih se kratkih razmatranja može doći do odgovora zašto udio UNIX baziranih operacijskih sustava sve više raste i zašto se i u budućnosti očekuje sve veći rast u svim područjima primjene

## 1.1 Organizacija podataka na disku

Podaci su i pod Unix operacijskim sustavom, kao i u većini drugih, organizirani u stablastu strukturu direktorija. Početni direktorij se naziva *root* direktorij (eng. root directory) i u operacijskom

sustavu, njegovo ime je / (kosa crta, eng. slash). Za razliku od DOS-a, gdje se koristi \ (obrnuta kosa crta, eng. backslash), u UNIX-u se uvijek pri rukovanju direktorijima koristi /. Bitno je naglasiti da u UNIX-u ne postoji velika razlika u pristupu direktorijima i datotekama - direktorij je samo posebna vrsta datoteke i većina stvari koja vrijedi za datoteke, vrijedi i za direktorije.

Sljedeća bitna razlika u odnosu na DOS je potpuno drukčiji pristup korištenju čvrstih diskova, odnosno particija na njima: unix omogućuje pristupanje svakoj pojedinoj particiji na nekom disku preko bilo kojeg direktorija (eng. mount point), a svaka particija i monogi direktoriji imaju neku svoju namjenu, npr.:

- / sadrži sve što je potrebno da se sistem može podići i da se na računalu može raditi
- /usr sadrži naredbe, biblioteke, stranice s uputstvima za upotrebu (eng. manual pages) i slične datoteke, koje se ne mijenjaju tokom rada na računalu
- /var sadrži datoteke koje se mijenjaju, privremene datoteke, *log* datoteke, *spool* direktorije za printere, e-mail i sl.
- /home sadrži korisničke poddirektorije-svaki korisnik ima svoj direktorij, unutar kojega se smještaju svi podaci vezani uz njegov rad - sve datoteke koje korisnik kreira (dokumenti, slike, projekti kreirani u različitim razvojnim okolinama i sl.) i sve postavke programa koje koristi (adresari web preglednika, izgled desktopa, izgled grafičkih programa, postavke različitih grafičkih i tekstualnih programa koje je korisnik koristio)

Takav način organizacije uvodi red u smještaj podataka na disku, tako da se točno zna gdje se što nalazi i čemu služi, nasuprot Windows pristupu, gdje uglavnom vlada prilično veliki kaos. Ovim su pristupom korisnici jedan od drugoga striktno odijeljeni i nemaju ni na koji način uvid u privatne podatke drugog korisnika.

Backup podataka se u ovom slučaju svodi na jednostavno komprimiranje i pohranu /home direktorija (za sve korisnike) ili vlastitog direktorija /home/moj\_dir za svakog pojedinog korisnika i time je sačuvano sve što je korisnik ikada radio.

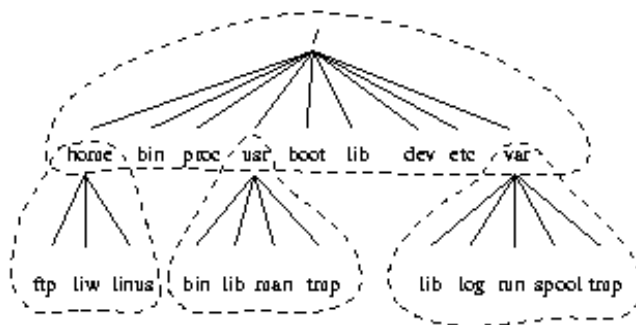
Ako dođe do potrebe da se stari podaci i postavke vrate na disk ili da korisnik pređe na drugo računalo, sve što je potrebno učiniti je kreirati korisnički direktorij (ako nije kreiran) te presnimiti u njega sačuvane podatke, i isto ga trenutka korisnik može nastaviti s normalnim radom, a da pri tome ne osjeti nikakvu promjenu!

U slučaju Windows operacijskog sustava, backup kreiranih datoteka može biti jednako jednostavan (ako se korisnik pridržava toga da sve snima unutar jednog direktorija), ali backup postavki programa (adresari, mail-ovi, postavke web preglednika i mail klijenata, postavke tekst procesora itd.) se pretvara u izrazito dugotrajan i mukotrpan posao pretraživanja interneta, dokumentacije i direktorija na disku kako bi se pronašle sve datoteke koje je potrebno sačuvati i vodila evidencija gdje ih treba snimiti. Sačuvati baš sve je gotovo nemoguće, ali niti povratak postavki ili prelazak na drugo računalo nije značajno lakši.

Na slici 1.1 je prikazan primjer strukture direktorija Unix operacijskog sustava. Crtkanim linijama, označene su granice između pojedinih particija na disku. Unutar osnovnog direktorija (/), koji predstavlja sadržaj *root* particije nalaze se i poddirektoriji *home*, *usr* i *var*. Prilikom podizanja sistema, prvo se sa *root* particije podiže operacijski sustav. Direktoriji *home*, *usr* i *var* su u tom

trenutku prazni direktoriji (ne sadrže niti jednu datoteku ili direktorij). Nakon početnog podizanja sustava, svako od tih direktorija se pridjeljuje odgovarajuća particija diska te struktura direktorija izgleda kao što je to prikazano slikom 1.1.

Iako se svi poddirektoriji direktorija *home* nalaze u strukturi direktorija *root* particije (/) oni su na disku fizički zapisani u nekoj drugoj particiji.



Slika 1.1 Primjer strukture direktorija i granica particija Unixa

## 1.2 Početak rada na računalu

Prije početka rada na računalu, korisnik se mora *logirati* (na nekim sustavima se može definirati i automatsko logiranje, pri čemu se nakon paljenja računala i podizanja sustava automatski prijavljuje *default* korisnik). To je proces prijave nekog korisnika operacijskom sustavu. Svaki korisnik ima svoje korisničko ime (eng. login, user name, user login) i zaporku (eng. password), s kojima se prijavljuje na računalo.

Ukoliko se korisnik želi prijaviti na serversko računalo, tada to može napraviti i sa nekog drugog računala (eng. remote computer) osim samog servera (eng. local computer, localhost), korištenjem *telnet* ili *ssh* (eng. Secure SHell) klijenta. Telnet i ssh klijenti su programi koji komuniciraju sa telnet, odnosno ssh, server programom na serverskom računalu.

Osnovna razlika između telnet i ssh načina komunikacije je što telnet sve podatke šalje u nekrptiranom (čistom tekstualnom) obliku, dok ssh koristi kriptiranje. Ssh je iz tog razloga, po pitanju sigurnosti, znatno pogodniji za korištenje, jer pri korištenju telnet-a, zloćudni korisnici, vrlo jednostavnim “slušanjem” prometa na mreži mogu doći do podataka o korisničkim zaporkama i svih drugih povjerljivih podataka.

Obzirom da se do serverskog računala vrlo često pristupa s računala koje ima Windows operacijski sustav (u koji je ugrađen telnet klijent, ali ne i ssh), to uzrokuje da se češće koristi telnet klijent, kojega bi radi sigurnosti trebalo maksimalno izbjegavati.

### 1.3 Izvođenje programa pod UNIX-om

Za razliku od DOS/Windows sustava gdje ekstenzija imena datoteke određuje da li se datoteka može pokrenuti, pod UNIX-om to određuje *x* pravo pristupa datoteci (vidi podpoglavlje 1.3) - operacijski će sustav pokušati pokrenuti svaku datoteku za koju korisnik ima *x* (execute) pravo pristupa.

Programi/naredbe se pod UNIX-om pokreću upisom imena programa i pritiskom na tipku <ENTER>. Svaki pokrenuti program se naziva proces (eng. process) i dobiva svoj jedinstveni identifikacijski broj (eng. PID-Process Identification Number) te i sam može stvarati i ubijati (eng. kill) druge procese.

Za ubijanje procesa (zaustavljanje izvođenja i uklanjanje iz memorije) koristi se naredba *kill*. Najjednostavniji način da se neki proces ubije je korištenjem:

```
kill -9 PID.
```

Značenje signala kojim se proces ubija (u ovom slučaju signal 9 - SIGKILL) i ostalih signala koji se mogu poslati procesu je izvan fokusa ove skripte.

Nakon pokretanja, izvođenje programa se može zaustaviti kombinacijom tipki Ctrl+Z ili se može prekinuti, kombinacijom tipki Ctrl+C. Osim standardnog pokretanja programa (eng. foreground) pri čemu on preuzima kontrolu nad komandnom linijom, program se može pokrenuti i u pozadini (eng. background). Pokretanjem u pozadini, program se normalno izvršava, ali komandna

linija ostaje slobodna da se korisnik može i dalje njome služiti i pokretati (prema potrebi) nove programe ili naredbe. Program se pokreće u pozadini tako da se iza njegova imena navede znak &.

Ispis pokrenutih procesa dobiva se naredbom *ps*, a ispis programa pokrenutih u pozadini, ili onih čije je izvođenje privremeno zaustavljeno, naredbom *jobs*.

Na slici 1.2 je prikazan primjer pokretanja dva programa u pozadini (test1 i test2) te ispisi naredbi *jobs* i *ps* nakon njihova pokretanja. Programi *test1* i *test2* su pokrenuli svaki po jedan proces *sleep*.

Program koji se nalazi u pozadini (eng. background) može se staviti u *foreground* naredbom *fg*. Za stavljanje programa u *background* koristi se naredba *bg*. Kao argument obje naredbe koristi se broj pokrenutog programa iz ispisa naredbe *jobs* (eng. job\_id), a ukoliko se broj izostavi, podrazumijeva se da se naredba odnosi na zadnji program koji je zaustavljen ili prebačen u pozadinu.

#### Primjer:

```
test1          pokreće se program test1
```

```
bash-2.03$ ps
  PID TTY          TIME CMD
 5529 pts/1        0:00 csh
 5815 pts/1        0:00 bash
bash-2.03$ test1 &
[1] 5818
bash-2.03$ test2 &
[2] 5820
bash-2.03$ jobs
[1]-  Running                  test1 &
[2]+  Running                  test2 &
bash-2.03$ ps
  PID TTY          TIME CMD
 5529 pts/1        0:00 csh
 5820 pts/1        0:00 bash
 5819 pts/1        0:00 sleep
 5818 pts/1        0:00 bash
 5815 pts/1        0:00 bash
 5821 pts/1        0:00 sleep
bash-2.03$ █
```

Slika 1.2 Pokretanje programa u pozadini

Ctrl+Z        zaustavlja se izvođenje programa *test1*  
 bg            izvođenje *test1* se nastavlja u pozadini  
 fg 3         program broj 3 se iz pozadine stavlja u *foreground*  
 kill -9 2451 ubija proces 2451

## 1.4 Dobivanje pomoći za korištenje naredbi

Obzirom da je dosta teško u svakom trenutku poznavati sve naredbe i načine njihovog korištenja, u UNIX-u za gotovo svaku naredbu i mnoge programe postoje stranice *priručnika* (eng. man pages). Na taj način korisnik pokretanjem naredbe *man* može dobiti detaljne informacije o načinu korištenja pojedine naredbe ili programa. Osim samih detalja oko korištenja, često se dobivaju i informacije o naredbama koje su na neki način povezane s tom naredbom (upotpunjuju je ili imaju sličnu ulogu), popis datoteka koje su s njom povezane, te niz primjera korištenja.

*Man* stranice su prvo mjesto gdje korisnik treba tražiti pomoć pri nailasku na neki problem.

Naredbe obrađene u ovim uputama imaju objašnjen samo dio mogućnosti korištenja. Za potpuniji pregled mogu se pogledati *man* stranice.

## 1.5 Dozvole pristupanja datotekama i direktorijima (eng. permissions)

Svaka datoteka i direktorij imaju tri grupe prava pristupa i svakoj su grupi pridijeljena tri prava:

- r (eng. read)        pravo čitanja sadržaja datoteka, odnosno ispisa sadržaja direktorija
- w (eng. write)     pravo mijenjanja sadržaja i brisanja datoteke, odnosno pravo stvaranja novih i brisanja postojećih datoteka unutar direktorija
- x (eng. execute)    pravo izvršavanja datoteka, odnosno ulaza u direktorij; po *x* dozvoli se određuje koja je datoteka izvršna, odnosno koje će datoteke sustav pokušati pokrenuti ako to korisnik zatraži (pod DOS-Windows to su datoteke s ekstenzijom .EXE, .COM i .BAT)

Grupe dozvola pristupanja odnose se na:

- vlasnika (eng. owner)    inicijalno je to korisnik koji je kreirao datoteku ili direktorij
- grupu (eng. group)      grupa korisnika koja ima posebno definiran pristup datoteci ili direktoriju
- ostali (eng. others)     svi ostali korisnici računala

Svi podaci o dozvolama i vlasnicima datoteka i direktorija mogu se dobiti pri ispisu sadržaja direktorija naredbom "ls -l". Primjer djela ispisa sadržaja direktorija je:

```
-rwxr-xr-x  2  mmarkovic  xxprojekt  19      Oct 29 16:11  test2
drwxr-xr-x  2  mmarkovic  yyprojekt  512     Oct 31 09:06  tmp
```

Prvi znak označava da li je riječ o direktoriju (d) ili o običnoj datoteci (-). Iza njega slijede 3 znaka koji označavaju dozvole za vlasnika, pa 3 znaka za grupu korisnika te na kraju za sve ostale.

Broj 2 označava da postoji jedan link na datoteku, a u sljedeća dva stupca se nalaze imena vlasnika datoteke i grupe koja joj može pristupiti. Nakon toga slijede veličina datoteke, datum zadnje promijene te njeno ime.

### **Primjer:**

Korisnik mmarkovic kreira datoteku *prva.verzija.doc* za koju želi da je mogu pročitati i dati o njoj svoje mišljenje i komentare svi njegovi kolege koji rade na projektu *xx*, ali da ujedno ne smiju i ne mogu mijenjati njen sadržaj, te da nitko drugi nema pristup toj datoteci. Da bi to postigao, potrebno je da sebi kao vlasniku dodijeli dozvole *rw*, grupi *xxprojekt r*, a svim ostalim korisnicima ukine sve dozvole pristupa. Ispis bi tada izgledao:

```
-rw-r----- 1 mmarkovic xxprojekt 45119 Oct 29 16:11 prva.verzija.doc
```

Nitko nema *x* dozvolu iz razloga što datoteka *prva.verzija.doc* nije izvršna datoteka i ne može se pokrenuti.

### **1.5.1 Promjena dozvola pristupa datotekama i direktorijima - naredba *chmod***

Naredbom *chmod* mogu se promijeniti dozvole pristupa za datoteke i direktorije, a njena je sintaksa:

```
chmod [-R] vrsta_dozvole datoteke_za_promjenu_dozvole
```

Opcija *-R* označava da je promjenu dozvola potrebno provesti rekurzivno (eng. recursive), odnosno i u svim poddirektorijima.

Vrsta dozvole se može definirati na dva načina - broječanom kombinacijom ili slovima. Ako se koriste brojevi, potrebno je upisati troznamenkasti broj, kod kojeg svaka znamenka predstavlja jednu grupu dozvola. Broj za svaku grupu se dobiva sumiranjem pojedinih dozvola, koje imaju sljedeće vrijednosti:

- r - read 4
- w - write 2
- x - execute 1

Dakle, ako se želi svima omogućiti čitanje i izvršavanje datoteke, a vlasniku i promjena, tada je vrsta dozvole *755*.

Ukoliko se vrsta dozvole definira slovima, potrebno je navesti na koju grupu se dozvola odnosi, na koji način se pridjeljuje i koje se dozvole pridjeljuju. Moguće grupe su:

- u - user vlasnik datoteke
- g - group grupa korisnika koja ima pristup datoteci
- o - others svi ostali korisnici računala
- a - all sve tri prethodne grupe

Prava se mogu pridjeljivati na tri načina:

= dozvole pristupa se postavljaju točno na navedene dozvole

- + navedene dozvole se dodaju postojećim dozvolama
- navedene dozvole se oduzimaju od postojećih

Konačno, moguće dozvole koje se mogu navesti su standardne *r*, *w* i *x* dozvole.

Datoteke za promjenu dozvole se mogu navesti ili pojedinačno (može ih biti i više, odvojene razmakom) ili korištenjem specijalnih znakova (\*, ?, []) za odabir grupe datoteka.

**Primjer:**

```
chmod 754 test1 test2 tmp
```

datotekama test1, test2 i tmp se pridijeljuju *rw*x dozvole za vlasnika, *rx* za grupu korisnika i *r* za sve ostale

```
chmod ug=rx test*
```

svim datotekama čiji naziv počinje sa test se pridijeljuju prava čitanja i pisanja za vlasnika i grupu (stare dozvole se brišu), dok se dozvole za sve ostale ne mijenjaju

```
chmod a-w test?
```

svima (vlasniku, grupi i ostalima) se oduzima pravo izmjene svih datoteka čije ime počinje sa test, a pored toga još imaju samo jedan znak u imenu (npr. test1, test5, testa, testb, ...)

```
chmod -R og+rx *xxprojekt*
```

svim datotekama u trenutnom direktoriju i svim poddirektorijima koje u svom imenu sadrže *xxprojekt*, se na postojeće dozvole za grupu i *ostale* dodaju i *read* i *execute*

```
chmod 700 xx*[abc]
```

svim datotekama čije ime počinje s *xx*, a završava sa jednim od slova *a*, *b* ili *c* se za vlasnika postavljaju *rw*x dozvole, a grupi i ostalima se ukidaju sva prava pristupa

## 1.6 Naredbe za rad s datotekama i direktorijima

Obzirom da u UNIX-u nema velike razlike između datoteka i direktorija (direktorij je posebna vrsta datoteke), mnoge naredbe se koriste na isti ili sličan način te će se zato i obraditi u istom poglavlju.

Pri radu s direktorijima se za razdvajanje pojedinih poddirektorija koristi znak / (eng. slash). Početni direktorij (eng. root) ima oznaku /, a korisnički *home* direktoriji se nalaze u osnovnom direktoriju, odnosno njihovo je puno ime /home. Unutar /home, svaki korisnik ima svoj direktorij, npr. /home/mmarkovic. Za pristup vlastitom *home* direktoriju može se koristiti znak ~ (eng. tilde), a za pristup *home* direktoriju nekog drugog korisnika ~ime\_korisnika, npr. ~iivic.

### 1.6.1 Ispis sadržaja direktorija

Za ispis sadržaja direktorija koristi se naredba *ls*, čija je sintaksa:

```
ls [opcije] datoteke
```

Od češće korištenih opcija bitno je spomenuti:

- *-l* za svaku datoteku ispisuje dozvole, broj linkova, vlasnika, grupu, veličinu i vrijeme zadnje promijene
- *-t* sortira ispit po vremenu zadnje promijene
- *-r* okreće redoslijed sortiranja
- *-a* ispisuje sve datoteke, uključujući i sakrivene (one čije ime počinje s *.*)
- *-R* ispisuje i sadržaj svih poddirektorija
- *-u* koristi se zajedno sa *-t* i sortira ispis po vremenu zadnjeg pristupa

Umjesto zasebnog navođenja svake opcije, UNIX omogućuje da se opcije spoje, npr. *ls -l -t -u* je isto što i *ls -ltu*.

#### Primjer:

```
ls          ispisuje popis imena svih datoteka i direktorija u trenutnom direktoriju
ls -lt     ispisuje sve podatke o datoteka i direktorijima trenutnog direktorija te sortira
           ispis po vremenu zadnje promijene
ls -altur  ispisuje datoteke i sve podatke o njima, uključujući i skrivene te ih sortira u
           padajućem nizu po vremenu zadnjeg pristupa
ls -R t*   ispisuje sve datoteke u trenutnom direktoriju čije ime počinje slovom t te
           sadržaj svih poddirektorija trenutnog direktorija, čija imena počinju sa t
           (uključujući i sve njihove poddirektorije)
ls ~       ispisuje sadržaj korisnikovog home direktorija
ls ~/tmp/t* ispisuje sve datoteke čije ime počinje slovom t, a nalaze se u poddirektoriju
           tmp, korisnikovog home direktorija
```

### 1.6.2 Ispis i promjena radnog direktorija

Korisnik naredbom *pwd* može u svakome trenutku provjeriti u kojem se direktoriju nalazi. Sintaksa naredbe je:

```
pwd
```

Za promjenu radnog direktorija koristi se naredba *cd*, čija je sintaksa:

```
cd [ime_direktorija]
```

Ukoliko se *cd* pozove bez argumenta *ime\_direktorija*, radni direktorij će se promijeniti u korisnički *home* direktorij.



**Primjer:**

`pwd`

ispisuje trenutni radni direktorij

`cd`

pozicionira se u korisnikov *home* direktorij (isto kao i `cd ~`)

`cd ~/tmp`

pozicionira se u direktorij *tmp* koji se nalazi u korisnikovom *home* direktoriju (isto kao i `cd /home/mmarkovic/tmp`)

`cd ../tmp`

pozicionira se u direktorij *tmp* koji se nalazi u istom direktoriju kao i trenutni direktorij (eng. parent directory)

`cd ../../tmp`

pozicionira se u direktorij *tmp* koji se nalazi u direktoriju koji je za 2 grane iznad trenutnog direktorija (npr. iz `/home/mmarkovic/dir1` u `/home/tmp`)

### 1.6.3 Naredba za kreiranje direktorija

Direktoriji se kreiraju naredbom *mkdir*, čija je sintaksa:

`mkdir [-m mode] [-p] ime_direktorija`

Opcijom `-m` mogu se definirati dozvole pristupa za novo-kreirani direktorij, dok opcija `-p` služi da se kreiraju svi potrebni poddirektoriji (ako ne postoje) do konačnog direktorija.

**Primjer:**

`mkdir moj_dir`

u trenutnom direktoriju kreira novi direktorij *moj\_dir*

`mkdir -m 700 ~/tmp`

u *home* direktoriju kreira direktorij *tmp* sa *rwX* dozvolama za vlasnika i bez ikakvih dozvola za sve ostale

`mkdir -p ~/tmp/tmp1/tmp2/tmp.konacni`

pod pretpostavkom da ne postoje, kreira direktorije *tmp1*, *tmp2* i *tmp.konacni*

### 1.6.4 Naredba za premještanje i preimenovanje direktorija i datoteka

Za premještanje i preimenovanje datoteka i direktorija koristi se naredba *mv*, čija je sintaksa:

```
mv [-fi] /stara_lokacija/staro_ime /nova_lokacija/novo_ime
```

Korištenjem opcije `-i` se korisnik osigurava da će `mv` postaviti zahtjev za potvrdom premještanja u slučaju da datoteka s takvim imenom već postoji na novoj lokaciji. Uz potvrdni odgovor, nova će se datoteka presnimati (eng. *overwrite*) preko stare, koja će se nepovratno izgubiti. Opcija `-f` (eng. *force*) ima suprotni učinak i ona služi da se izostavi zahtjev za potvrdom presnimavanja datoteka preko postojećih te da se nove datoteke bezuvjetno premjeste na novu lokaciju.

U posebnom slučaju, kada su stara i nova lokacija iste, a staro i novo ime različiti postize se efekt preimenovanja datoteke ili direktorija.

### Primjer:

```
mv -f file1 ~/file2
```

bezuovjetno premješta datoteku *file1* iz trenutnog direktorija u korisnikov *home* direktorij, pod novim imenom *file2*; ako *file2* već postoji, njen će se sadržaj izgubiti jer će bit presnimljen s drugom datotekom, a ako je *file2* direktorij, tada će se *file1* premjestiti u *file2*, i zadržat će staro ime: *file1*

```
mv -i file* ~/tmp
```

premješta sve datoteke i direktorije čiji naziv počinje s *file* trenutnog direktorija u direktorij *tmp* koji se nalazi u korisnikovom *home* direktoriju; ako neka od datoteka s takvim imenom već postoji, `mv` će zahtjevati potvrdu, prije nego što preko nje presnimi novu datoteku

```
mv tmp tmp1
```

preimenuje datoteku ili direktorij *tmp* u trenutnom direktoriju u *tmp1*

```
mv tmp ~/tmp1
```

premješta datoteku ili direktorij *tmp* trenutnog direktorija u direktorij *tmp1*, koji se nalazi u korisnikovom *home* direktoriju

```
mv ~/tmp1/moj_dir* ~
```

premješta sve datoteke i direktorije čije ime počinje s *moj\_dir* iz direktorija *tmp1* koji se nalazi u korisnikovom *home* direktoriju, u korisnikov *home* direktorij

## 1.6.5 Kopiranje datoteka i direktorija

Datoteke i direktoriji se kopiraju korištenjem naredbe `cp`, čija je sintaksa:

```
cp [opcije] source destination
```

Najčešće korištene opcije su:

- `-i` zahtjeva potvrdu kopiranja, ako datoteka s tim imenom već postoji na odredištu (bez opcije `-i` se datoteka automatski presnimava preko odredišne)

- -r kopira i sadržaj svih poddirektorija
- -p osim samih datoteka, kopira i dozvole pristupa, vlasnika, grupu te vrijeme zadnje promjene, i vrijeme zadnjeg pristupa

### Primjer:

cp dat1 /tmp/dat2

kopira datoteku *dat1* iz radnog direktorija u direktorij *tmp*, koji se nalazi unutar radnog direktorija, pod novim imenom *dat2*

cp -i \*dat\* ~/tmp

kopira sve datoteke koje u svom imenu sadrže *dat* u direktorij *tmp* unutar korisničkog *home* direktorija; ako već postoji datoteka s takvim imenom, zahtjeva potvrdu, prije nego što preko nje presnima drugu datoteku

cp -p dat1 tmp

kopira datoteku *dat1* iz radnog direktorija u direktorij *tmp* (ako direktorij ne postoji, tada se datoteka kopira u novu datoteku s imenom *tmp*) unutar radnog direktorija, uz sačuvane podatke o dozvolama pristupa, vlasniku, grupi i vremenu zadnje promjene, odnosno pristupa

cp -r ~/tmp ~/tmp\_copy

direktorij *tmp* iz *home* direktorija se kopira u isti taj direktorij, ali pod novim imenom - rezultat su dva direktorija istog sadržaja, ali različitog imena

cp -r ../moj\_dir ~/tmp

direktorij *moj\_dir*, koji se nalazi u *parent* direktoriju se kopira u *tmp* direktorij, korisnikovog *home* direktorija

### 1.6.6 Brisanje datoteka i direktorija

Za brisanje datoteka i direktorija koriste se naredbe *rm* i *rmdir*, čije su sintakse:

rm [-rif] ime\_direktorija

rmdir ime\_direktorija

Naredbom *rm* se mogu obrisati direktoriji i poddirektoriji bez obzira na to da li su prazni ili sadrže druge datoteke i direktorije, dok se naredbom *rmdir* prvenstveno brišu prazni direktoriji.

Opcija naredbe *rm* su:

- -i koristi se da bi se od korisnika tražila potvrda (eng. confirmation) prije brisanja datoteka i direktorija
- -f ima suprotni učinak od *i* - sve se datoteke i direktoriji brišu bez traženja potvrde
- -r datoteke i direktorij se brišu rekurzivno, odnosno, brišu se i svi poddirektoriji

zadanog direktorija

**Primjer:**

`rm -ri tmp`

bríše direktorij *tmp* trenutnog direktorija i sve što se u njemu nalazi (uključujući i poddirektorije), a prije svakog brisanja zahtjeva od korisnika potvrdu

`rm -rf ~/tmp1`

bríše direktorij *tmp1*, koji se nalazi u korisnikovom *home* direktoriju, i sve što se u njemu nalazi, bez da se traži potvrda

`rm -i *zip`

iz radnog direktorija bríše sve datoteke čije ime završava sa *zip*, a prije brisanja svake datoteke, zahtjeva potvrdu

`rmdir moj_dir`

pod uvjetom da je prazan, bríše direktorij *moj\_dir*, koji se nalazi u trenutnom direktoriju

### 1.6.7 Ispis sadržaja datoteka

Za ispis sadržaja tekstualnih datoteka koriste se naredbe *less* i *cat*. Sintaksa naredbi je:

`less ime_datoteke`

`cat [opcije] ime_datoteke [| more] [| less]`

Ispisom pomoću *less* dobiva se ispis stranicu po stranicu, s mogućnošću navigacije unutar datoteke. Korištenje naredbe *cat* omogućuje korištenje nekih dodatnih opcije, kao npr.:

- `-n` pri ispisu numerira redke datoteke
- `-b` pri ispisu numerira redke datoteke, ali izostavlja prazne retke

Uz naredbu *cat*, mogu se koristiti i *more* i *less*, koji omogućuju navigaciju po ispisu naredbe *cat*.

Osim navedenih naredbi za ispis cijele tekstualne datoteke, postoje i naredbe za ispis samo početka, odnosno kraja datoteke:

`head [-broj_redaka] ime_datoteka`

`tail [+/-broj_redaka][r] ime_datoteka`

*Head* naredba ispisuje *broj\_redaka* redaka od početka datoteke, a ako se *broj\_redaka* izostavi, tada je podrazumijevana vrijednost 10.

*Tail* ispisuje *broj\_redaka* redaka datoteka, računajući od početka (+) ili od kraja (-) datoteke. Ispis može biti i u suprotnom redosljedu (r).

**Primjer:**

```
less file1.txt
```

ispisuje sadržaj *file1.txt*, stranicu po stranicu

```
cat -n index.html
```

ispisuje sadržaj *index.html* i numerira sve redke, uključujući i prazne

```
cat -b index.html | more
```

ispisuje sadržaj *index.html* i numerira sve redke osim praznih, te ispisuje rezultat stranicu po stranicu

```
head *txt
```

ispisuje prvih 10 redaka svih datoteka koje završavaju s *txt*

```
head -20 index*
```

ispisuje prvih 20 redaka svih datoteka čije ime počinje s *index*

```
tail -20r *txt
```

ispisuje u suprotnom redosljedu zadnjih 20 redaka svih datoteka čije ime završava s *txt*

```
tail +20 *html
```

ispisuje sve redke od 20. do zadnjeg, datoteka čije ime završava s *html*

## 1.7 Redirekcija podataka i ulančavanje naredbi

Pod UNIX-om su definirane standardne datoteke: standardni ulaz, standardni izlaz i standardni izlaz za greške. Za sve tri datoteke se podrazumijeva da se koristi terminal. Programi koji se izvršavaju, u principu, sve što je potrebno ispisuju na standardni izlaz, a sve potrebne podatke učitavaju sa standardnog ulaza. Pri tome sami program ne zna što je u stvari standardni ulaz, odnosno izlaz. To omogućuje da se prije pokretanja programa definira npr., da je standardni ulaz datoteka *ulaz.dat*, a standardni izlaz datoteka *ulaz2.dat*. Na taj će način program učitati sve što mu je potrebno iz prve datoteke, dok će rezultat ispisa snimiti u drugu datoteku, umjesto ispisa na ekran.

To preusmjeravanje standardnog ulaza, izlaza i izlaza za greške se naziva redirekcija. Za redirekciju se koriste sljedeći znakovi:

- < preusmjerava standardni ulaz na neku datoteku ili vanjsku jedinicu
- > preusmjerava standardni izlaz u neku datoteku ili vanjsku jedinicu; ako datoteka već postoji, biti će izbrisana
- >> preusmjerava standardni izlaz tako da podatke dodaje na kraj postojeće datoteke (ako datoteka ne postoji, biti će kreirana)

Standardni izlaz za greške se ne može samostalno preusmjeravati, ali se može preusmjeriti zajedno sa standardnim izlazom, dodavanjem znaka & iza > ili >>.

Osim preusmjeravanja podataka, moguće je i “ulančavanje” toka podataka kroz “protočnu strukturu” (eng. pipeline). Korištenjem znaka | može se standardni izlaz iz jedne naredbe povezati sa standardnim ulazom u drugu, što omogućuje povezivanje više naredbi koje koriste i modificiraju jedan ulazni niz podataka.

**Primjer:**

ls -l > sadrzaj\_direktorija  
ispis sadržaja radnog direktorija snima u datoteku *sadrzaj\_direktorija* (ako datoteka već postoji, njen će sadržaj biti izgubljen)

ls -l >> sadrzaj\_direktorija  
ispis sadržaja radnog direktorija dodaje se na kraj datoteke *sadrzaj\_direktorija* (ako datoteka na postoji, biti će kreirana)

sort < file1  
preusmjerava podatke iz datoteke *file1* na standardni ulaz naredbe *sort*

rmdir tmp >& /dev/null  
preusmjerava standardni izlaz i standardni izlaz za greške u /dev/null (/dev/null se naziva *data sink*, jer se gube svi podaci koji se u njega upišu)

ls | sort  
sortira ispis naredbe *ls* (standardni izlaz od *ls* je spojen sa standardnim ulazom naredbe *sort*)

## 1.8 Ostale naredbe UNIX-a

Obzirom da bi bilo nemoguće u ovim kratkim uputama obraditi sve mogućnosti i naredbe UNIX-a, u ovom će poglavlju biti obrađene neke od češće korištenih.

### 1.8.1 Pretraživanje diska

Za traženje datoteka i direktorija na disku koriste se naredbe *find* i *locate*, čije su sintakse:

```
find pocetni_dir izraz
locate dio_imena_datoteke
```

Naredba *locate* će u svojoj bazi podataka pokušati pronaći sve datoteke koje sadrže u svom imenu *dio\_imena\_datoteke*. Baza se obnavlja periodički, što znači da se u njoj neće nalaziti nedavno kreirano datoteke.

Naredba *find* će, počevši od direktorija *pocetni\_dir*, pokušati pronaći na disku datoteku ili grupu datoteka koje su definirane izrazom *izraz*. Neki od mogućih izraza su:

- `-atime` `[+/-]n` sve datoteke i direktoriji kojima je pristupano prije više od  $+n$  dana, prije manje od  $-n$  dana ili prije točno  $n$  dana
- `-ctime` `[+/-]n` sve datoteke i direktoriji čiji je sadržan mijenjan prije više od  $+n$  dana, prije manje od  $-n$  dana ili prije točno  $n$  dana
- `-group` `gname` sve datoteke i direktoriji koji pripadaju grupi *gname*
- `-user` `username` sve datoteke i direktoriji čiji je vlasnik *username*
- `-size` `[+/-]nc` sve datoteke koje su veće od  $+n$  B, manje od  $-n$  B, odnosno velike točno  $n$  B
- `-name` `izraz` sve datoteke i direktoriji koji su definirani *izrazom*, koji može biti bilo koji od standardnih izraza (*ime*, *dio\_imena\**, *\*dio?*, i sl.)

**Primjer:**

`locate file`

ispisuje sve datoteke i direktorije koji u svom imenu sadrže *file*

`find ~ -atime +10 -group moja_grupa -size -2048c`

ispisuje sve datoteke i direktorije unutar korisničkog *home* direktorija i svih njegovih poddirektorija, kojima je pristupano prije više od 10 dana, pripadaju grupi *moja\_grupa* i manje su od 2KB

`find . -name *txt -user korisnik -ctime -2`

ispisuje sve datoteke čije ime završava na *txt*, čiji je vlasnik *korisnik* i koje su zadnji puta mijenjane prije manje od dva dana

`find ~/tmp -ctime 10`

ispisuje sve datoteke unutar direktorija *tmp*, koji se nalazi u korisničkom *home* direktoriju, koje su zadnji puta mijenjane prije točno 10 dana

### 1.8.2 Usporedba datoteka

Pomoću naredbe *diff*, mogu se usporediti sadržaji dviju tekstualnih datoteka. Na standardni izlaz se ispisuju sve razlike među datotekama. Sintaksa je:

`diff [opcije] datoteka1 datoteka2`

Od mnogobrojnih opcija može se izdvojiti *-i*, koja pri uspoređivanju uzrokuje zanemarenje razlike da li se u tekstu nalaze mala ili velika slova.

**Primjer:**

`diff -i dat1 dat2`

ispisuje razlike između datoteka *dat1* i *dat2*, ne vodeći računa o velikim i malim slovima

### 1.8.3 Prebrojavanje znakova unutar datoteke

Za prebrojavanje znakova unutar neke tekstualne datoteke koristi se naredba `wc`, čija je sintaksa:

```
wc [opcije] ime_datoteke
```

Od opcija je bitno spomenuti:

- `-C` broji znakove
- `-l` broji redke
- `-w` broji riječi

#### Primjer:

```
wc -C file1.txt  
ispisuje koliko u datoteci file1.txt ima znakova
```

### 1.8.4 Naredbe za dobivanje podataka o zauzeću diska

Podaci o slobodnom prostoru na disku, odnosno o zauzeću mjesta mogu se dobiti korištenjem naredbi `df` (eng. disk free) i `du` (eng. disk usage). Sintaksa naredbi je:

```
df [-k]  
du [-k] pocetni_dir
```

#### Primjer:

```
df -k ispisuje u KB i postocima koliko je na kojoj particiji slobodnog, odnosno  
koliko je zauzetog prostora  
du -k ~ ispisuje u KB koliko mjesta na disku zauzima korisnikov home direktorij i svi  
njego vi poddirektoriji
```

### 1.8.5 Naredba za pronalazak uzorka u znakovnom nizu

Ako je unutar nekog znakovnog niza potrebno pronaći neki uzorak teksta, tada se koristi naredba `grep`. Rezultat će biti ispis svih redaka u kojima se nalazi zadani uzorak. Naredba `grep` se najčešće koristi u kombinaciji s drugim naredbama.

#### Primjer:

```
cat popis | grep Marko  
ispisuje sve redke datoteke popis, koji u sebi sadrže ime Marko
```

### 1.8.6 Naredba za sortiranje podataka



Za sortiranje podataka koristi se naredba *sort*, čija je sintaksa:

```
sort [opcije] podaci
```

Češće korištene opcije su:

- `-r` sortiranje u suprotnom redosljedu
- `-k n` sortiranje po *n*-tom stupcu podataka

Nakon pokretanja naredbe, mogu se upisivati podaci koje je potrebno sortirati. Kraj upisa podataka označava se pritiskom kombinacije tipki Ctrl+D (skraćeni zapis ^D).

Naredba *sortse* na ovaj način koristi vrlo rijetko, ali se zato dosta česta primjena u kombinaciji s drugim naredbama.

**Primjer:**

```
ls -l | sort -r -k 9
sortira rezultat naredbe ls -l u padajućem nizu, po 9 stupcu (u ovom slučaju po imenu datoteke)
```

```
cat file1.txt | grep Adresa | sort > file1.txt
sortira sve redke datoteke file1.txt, koji u sebi sadrže riječ Adresa i zapisuje ih u datoteku file1.txt
```

### 1.8.7 Naredbe za dobivanje informacija o drugim korisnicima

Za dobivanje informacija o drugim korisnicima računala, najčešće se koriste naredbe *who* i *finger*.

**Primjer:**

```
who ispisuje tko je trenutno, i s kojeg računala, logiran na računalo
finger jsmith ispisuje podatke o korisniku računala čije je korisničko ime (login) jsmith
(korisnik ne mora biti logiran na računalu)
```

### 1.8.8 Promjena zaporke (eng. password)

Korisnik u svakom trenutku smije promijeniti vlastiti *password*, koristeći naredbu *passwd*. Nakon pokretanja, potrebno je upisati stari *password* te nakon toga dva puta za redom novi. Pri odabiru, potrebno je voditi računa o tome da ga se lako zapamti, ali da ga je teško pogoditi, te da zadovoljava minimalne uvjete koje operacijski sustav zahtjeva (npr. minimalna dužina ili korištenje barem jednog velikog i jednog malog slova i sl).

### 1.8.9 Ispis kalendara

Naredbom *cal*, dobiva se na standardni izlaz ispis kalendara za bilo koji mjesec i godinu. Sintaksa naredbe je:

```
cal [mjesec [godina]]
```

Ako se ne navedu mjesec ili godina, tada je podrazumijevana vrijednost tekući mjesec, odnosno tekuća godina.

```
cal          ispisuje kalendar za tekući mjesec u tekućoj godini
cal 10 2003  ispisuje kalendar za 10. mjesec, 2003. godine
```

## 1.9 Naredbe za komunikaciju s drugim korisnicima

Često se javlja potreba za komunikacijom između dva korisnika računala. U svakom se trenutku može poslati e-mail, ali je često jednostavnije poslati kratku poruku koju će drugi korisnik vidjeti na ispisu terminala. Također, ukoliko je potrebna kratka izmjena mišljenja oko neke teme, komunikacija elektronskom poštom bi trajala predugo te je bolje uspostaviti direktnu komunikaciju.

### 1.9.1 Naredbe za slanje poruka

Poruka se drugom korisniku, koji je trenutno logiran na računalu, može poslati korištenjem naredbe *write*, čija je sintaksa:

```
write user_login [poruka]
```

Ako se ne navede poruka, tada se ona upisuje nakon pokretanja naredbe. U tom slučaju, poruka može imati proizvoljan broj redaka, a upis se završava pritiskom kombinacije tipki Ctr+D.

#### Primjer:

```
write mmarkovic  Kasnimo na vjezbe!
```

### 1.9.2 Naredba za otvaranje razgovora između dva korisnika

“Razgovor” se između dva korisnika (koji su trenutno logirani na računalu) započinje naredbom *talk*:

```
talk user_login
```

#### Primjer:

```
talk mmarkovic
```

Nakon pokretanja, dobiva se poruka: “Waiting for your party to respond”, dok se korisniku s kojim se pokušava uspostaviti komunikacija ispisuje poruka tko želi s njim

“razgovarati” i što treba napisati da bi se uspostavila komunikacija: “talk: respond with: talk iivic@rijeka.riteh.hr”. Ako *mmarkovic* želi razgovarati s *iivicem*, treba upisati *talk iivic*, a u protivnom može normalno nastaviti s radom.

Razgovor se prekida pritiskom kombinacije tipki Ctrl+D ili Ctrl+C.

### 1.9.3 Kontrola primanja poruka i zahtjeva za razgovorom

Ukoliko dođe do situacije kada korisniku računala smetaju dolazeće poruke i zahtjevi za razgovor, on ih korištenjem naredbe *mesg* može zabraniti. Sintaksa naredbe je:

```
mesg [-y | -n]
```

Opcija *-y* omogućava primanje poruka i zahtjeva za razgovor, dok ih *-n* zabranjuje. Ukoliko se izostave obje opcije, tada se ispisuje trenutni status.

#### Primjer:

```
mesg          ispisuje status  
mesg -y      dozvoljava ispis poruka i zahtjeve za razgovor
```

## 1.10 Zadaci

1. U svom *home* direktoriju kreirati direktorij *proba*.
2. U direktoriju *proba* kreirati datoteku naziva *moj.home.dir* s ispisom sadržaja *home* direktorija u dugom formatu, sortirano po vremenu zadnje promjene (koristiti redirekciju).
3. Promijeniti dozvole pristupa datoteci *moj.home.dir* tako da korisnik ima *r* i *w* pravo pristupa, a grupa i svi ostali samo *r*.
4. Preimenovati datoteku *moj.home.dir* u *moj.dir*.
5. Preimenovati direktorij *proba* u *proba1*.
6. U svom *home* direktoriju kreirati direktorij *proba.nova*.
7. U direktoriju *proba.nova* kreirati datoteku *korisnici*, koja sadrži popis svih korisnika koji su trenutno logirani na serveru *rijeka.riteh.hr*, a za rad su se prijavili preko nekog od računala unutar fakulteta (adresa računala s kojeg su se logirali u sebi sadrži *riteh.hr*). Korisnici u datoteci trebaju biti sortirani po adresi računala s kojeg su se logirali. Koristiti *who*, *grep*, *sort*, ulančavanje naredbi i redirekciju.
8. Ispisati sadržaj datoteke *korisnici* uz numeriranje redaka.
9. Prebrojati koliko riječi sadrži datoteka *korisnici*.
10. Pronaći sve datoteke u svom *home* direktoriju i svim njegovim poddirektorijima koje su mijenjane prije manje od 200 dana, kojima je pristupano prije manje 20 dana i koje su veće od 1KB.
11. Iskopirati direktorij *proba.nova* u direktorij *proba1*.
12. Provjeriti koliko mjesta na disku zauzimaju datoteke *home* direktorija.
13. Promijeniti *password*.
14. U direktoriju *proba1* kreirati datoteku *veljaca* u kojoj je zapisan kalendar mjeseca veljače, 2002. godine.
15. Datoteku *veljaca* iskopirati u direktorij *proba.nova*
16. Izbrisati direktorije *proba1* i *proba.nova*

|       |   |    |
|-------|---|----|
| 1.    | Unix/Linux  | 1  |
| 1.1   | Organizacija podataka na disku  | 1  |
| 1.2   | Početak rada na računalu  | 3  |
| 1.3   | Izvođenje programa pod UNIX-om  | 4  |
| 1.4   | Dobivanje pomoći za korištenje naredbi                                      | 5  |
| 1.5   | Dozvole pristupanja datotekama i direktorijima (eng. permissions)           | 5  |
| 1.5.1 | Promjena dozvola pristupa datotekama i direktorijima - naredba <i>chmod</i> | 6  |
| 1.6   | Naredbe za rad s datotekama i direktorijima                                 | 7  |
| 1.6.1 | Ispis sadržaja direktorija  | 8  |
| 1.6.2 | Ispis i promjena radnog direktorija   | 8  |
| 1.6.3 | Naredba za kreiranje direktorija  | 9  |
| 1.6.4 | Naredba za premještanje i preimenovanje direktorija i datoteka              | 9  |
| 1.6.5 | Kopiranje datoteka i direktorija  | 10 |
| 1.6.6 | Brisanje datoteka i direktorija   | 11 |
| 1.6.7 | Ispis sadržaja datoteka   | 12 |
| 1.7   | Redirekcija podataka i ulančavanje naredbi                                  | 13 |
| 1.8   | Ostale naredbe UNIX-a   | 14 |
| 1.8.1 | Pretraživanje diska   | 14 |
| 1.8.2 | Usporedba datoteka  | 15 |
| 1.8.3 | Prebrojavanje znakova unutar datoteke                                       | 16 |
| 1.8.4 | Naredbe za dobivanje podataka o zauzeću diska                               | 16 |
| 1.8.5 | Naredba za pronalazak uzorka u znakovnom nizu                               | 16 |
| 1.8.6 | Naredba za sortiranje podataka  | 16 |
| 1.8.7 | Naredbe za dobivanje informacija o drugim korisnicima                       | 17 |
| 1.8.8 | Promjena zaporke (eng. password)  | 17 |
| 1.8.9 | Ispis kalendara   | 18 |
| 1.9   | Naredbe za komunikaciju s drugim korisnicima                                | 18 |
| 1.9.1 | Naredbe za slanje poruka  | 18 |
| 1.9.2 | Naredba za otvaranje razgovora između dva korisnika                         | 18 |
| 1.9.3 | Kontrola primanja poruka i zahtjeva za razgovorom                           | 19 |
| 1.10  | Zadaci  | 20 |