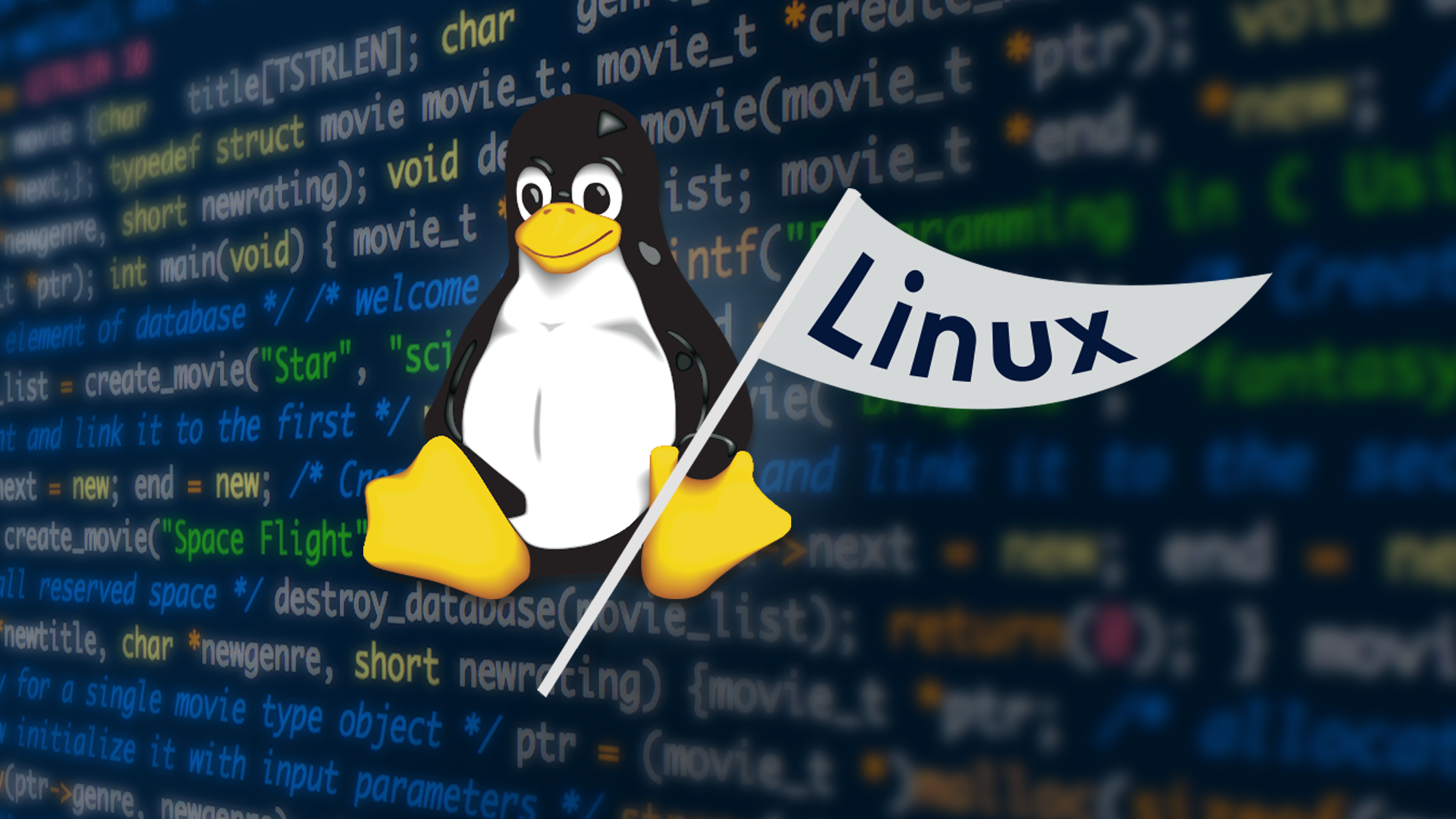
**Uvod u Linux**



**Sadržaj**

[1. Osnovni pojmovi 1](#_Toc483079003)

[1.1. Osnove rada u *Linuxu* 1](#_Toc483079004)

[1.2. Načini rada 1](#_Toc483079005)

[1.3. Udaljeni pristup 1](#_Toc483079006)

[1.4. Vježba: Prijavljivanje na sustav i osnove rada s programom PuTTY 2](#_Toc483079007)

[1.5. Pitanja za ponavljanje 4](#_Toc483079008)

[2. Naredbe u *Linuxu* 5](#_Toc483079009)

[2.1. Radni direktorij 5](#_Toc483079010)

[2.2. Prompt 5](#_Toc483079011)

[2.3. Izgled naredbi 6](#_Toc483079012)

[2.4. Promjena lozinke 6](#_Toc483079013)

[2.5. Odjavljivanje sa sustava 7](#_Toc483079014)

[2.6. Naredbe echo i clear 7](#_Toc483079015)

[2.7. Informacije o korisnicima na sustavu: who, w, i last 7](#_Toc483079016)

[2.9. Sustav pomoći: man i info 8](#_Toc483079017)

[2.10. Pitanja za ponavljanje 9](#_Toc483079018)

[3. Sustav datoteka 11](#_Toc483079019)

[3.1. Općenito o sustavu datoteka 11](#_Toc483079020)

[3.2. Imena datoteka 11](#_Toc483079021)

[3.3. Ispis radnog direktorija: pwd 12](#_Toc483079022)

[3.4. Apsolutne i relativne staze 13](#_Toc483079023)

[ljuska 3.5. Promjena radnog direktorija: cd 14](#_Toc483079024)

[3.6. Ispis sadržaja direktorija: ls 14](#_Toc483079025)

[3.7. Stvaranje i uklanjanje direktorija: mkdir i rmdir 15](#_Toc483079026)

[3.8. Stvaranje datoteka: touch 16](#_Toc483079027)

[3.9. Brisanje datoteka: rm 16](#_Toc483079028)

[3.10. Kopiranje i premještanje datoteka i direktorija: cp i mv 17](#_Toc483079029)

[3.11. Ispis sadržaja datoteka: cat, more i less aaaa 18](#_Toc483079030)

[3.12. Zamjenski znakovi 18](#_Toc483079031)

[3.13. Pretraživanje sustava datoteka: find 19](#_Toc483079032)

[3.14. (π)Pitanja za ponavljanje 19](#_Toc483079033)

[4. Prava pristupa datotekama i direktorijima 21](#_Toc483079034)

[4.1. Naredba id 21](#_Toc483079035)

[4.2. Razine prava pristupa: *user*, *group* i *other* 21](#_Toc483079036)

[4.3. Vrste prava pristupa: *read*, *write* i *execute* 22](#_Toc483079037)

[4.4. Postavljanje prava pristupa pomoću simboličkih oznaka 23](#_Toc483079038)

[4.5. Postavljanje prava pristupa pomoću numeričkih oznaka 24](#_Toc483079039)

[4.6. Pitanja za ponavljanje 25](#_Toc483079040)

[5. Uređivač teksta *GNU nano* 26](#_Toc483079041)

[5.1. Općenito o uređivačima teksta 26](#_Toc483079042)

[5.2. Osnovna svojstva uređivača teksta *GNU nano* 26](#_Toc483079043)

[5.3. Pokretanje programa 26](#_Toc483079044)

[5.4. Osnovne komande 28](#_Toc483079045)

[5.5. Pitanja za ponavljanje 29](#_Toc483079046)

[6. Ljuska *Bash* 30](#_Toc483079047)

[6.1. Što je ljuska? 30](#_Toc483079048)

[6.2. Upis naredbi 30](#_Toc483079049)

[6.3. Vrste naredbi 30](#_Toc483079050)

[6.4. Pitanja za ponavljanje 32](#_Toc483079051)

[7. Procesi 33](#_Toc483079052)

[7.1. Općenito o procesima 33](#_Toc483079053)

[7.2. Popis procesa: ps i top 33](#_Toc483079054)

[7.3. Izvršavanje naredbi u pozadini 34](#_Toc483079055)

[7.4. Upravljanje procesima u ljusci 34](#_Toc483079056)

[7.5. Signali i naredba kill 35](#_Toc483079057)

[7.6. Pitanja za ponavljanje 36](#_Toc483079058)

[8. Arhiviranje i sažimanje datoteka 37](#_Toc483079059)

[8.1. Arhiviranje datoteka: tar 37](#_Toc483079060)

# 1. Osnovni pojmovi

## 1.1. Osnove rada u *Linuxu*

*Linux* omogućava istovremeni rad više korisnika, stoga kažemo da je *Linux* **višekorisnički sustav**.

Svakom korisniku dodijeljen je **korisnički račun** – skup resursa koje može koristiti. Primjeri resursa koje korisnici mogu koristiti su diskovni prostor, procesorsko vrijeme, radna memorija, pisači i drugo.

Kako bi sustav mogao razlikovati korisničke račune (a time i same korisnike na sustavu), svaki korisnički račun ima svoje **ime**. Taj je podatak, zajedno s korisničkom **lozinkom** potrebno upisati prilikom prijavljivanja za rad na sustavu. Na taj način sustav zna koji se korisnik prijavljuje i provjerava njegov identitet. Korisničko je ime javni podatak, dok je lozinka tajni podatak.

**Napomena**

Važno je čuvati tajnost lozinki!

**Napomena**

Korisničko ime administratora sustava je **root**.

Sustav se brine o tome da su korisnička radna okruže~

nja međusobno odvojena te da korisnici mogu upravljati svojim radnim okruženjem – određivati prava pristupa do svojih podataka, podešavati svoje radno okruženje (putem konfiguracijskih datoteka), pokretati programe i upravljati njihovim izvođenjem i drugo.

## 1.2. Načini rada

**Napomene**

**Terminali** se sastoje od ekrana i tipkovnice.

**Konzola** je terminal putem kojeg se upravlja računalom tijekom pokretanja sustava te se na njega ispisuju kritične poruke za vrijeme rada sustava.

*Linux* omogućava više načina rada. Prvi način rada je neposredan rad na konzoli ili terminalu koji su direktno spojeni na računalo. Drugi je način udaljen – putem računalne mreže i drugog računala koje simulira terminal.

U oba je slučaja moguće raditi u tekstualnom ili grafičkom radnom okruženju.

U **tekstualnom radnom okruženju** sustavom se upravlja unosom tekstualnih naredbi.

U **grafičkom radnom okruženju** sustavom se upravlja putem grafičkog sučelja u kojem su pojedini elementi sustava i naredbe prikazani kao grafički objekti (ikone).

**Napomena**

Rad u grafičkom radnom okruženju sustava *Linux*, napredno služenje sustavom *Linux* te sistemska administracija sustava *Linux* nisu predmet ovog tečaja.

## 1.3. Udaljeni pristup

Komunikaciju između udaljenog i lokalnog sustava određuje odabrani protokol – jezik kojim se služe računala za povezivanje i razmjenu podataka.

Danas se za terminalski pristup udaljenim računalima najčešće koristi protokol **SSH** koji omogućava sigurno povezivanje računala kriptiranjem podataka koji se razmjenjuju među računalima.

Postoje i drugi protokoli za terminalski pristup udaljenim računalima. Neki od njih (na primjer, **telnet** ili **rlogin**) ne koriste kriptiranje prilikom razmjene podataka. Ti se protokoli smatraju nesigurnima i njihovo korištenje treba izbjegavati.

Da bi se putem lokalnog računala moglo raditi na udaljenom računalu, potrebno je na lokalnom računalu imati instaliran odgovarajući program (klijent). Taj program ima dvojnu funkciju: on ostvaruje vezu s udaljenim računalom (rabeći pri tome odgovarajući protokol) te simulira rad terminala.

**Napomena**

Nesigurne protokole koji ne koriste kriptiranje prilikom razmjene podataka treba izbjegavati!

**PuTTY** je jedan od takvih programa. Besplatan je i moguće ga je preuzeti na adresi: **http://www.chiark.greenend.org.uk/~sgtatham/putty/**.

Instalacija tog programa je vrlo jednostavna: nakon preuzimanja s Interneta dovoljno je program raspakirati i kopirati na željeno mjesto na disku.

Da bi se mogla uspostaviti veza s udaljenim računalom, potrebno je poznavati njegovo mrežno (simboličko) ime ili njegovu mrežnu (numeričku) adresu.

Mrežna imena strojeva su oblika *imestroja.domena* (na primjer, zeus.pitagora.hr), a numeričke adrese su nizovi od četiri okteta razdvojenih točkom (na primjer, 162.54.8.89).

## 1.4. Vježba: Prijavljivanje na sustav i osnove rada s programom PuTTY

Da bi se ova vježba mogla izvesti potrebno je:

* na lokalnom stroju imati instaliran program PuTTY
* poznavati mrežno ime stroja na kojem će se raditi
* imati korisnički račun na udaljenom stroju (poznavati svoje korisničko ime i lozinku).

Navedene podatke dobit ćete od predavača na početku vježbe. Zapamtite ili zabilježite te podatke jer ćete ih često koristiti tijekom ovog tečaja.

1. **Pokrenite program PuTTY.**

ikona2Program PuTTY pokreće se na neki od uobičajenih načina: dvostrukim klikom na ikonu koja na radnoj površini ili na tvrdom disku predstavlja navedeni program ili odgovarajućim odabirom u izborniku sustava.

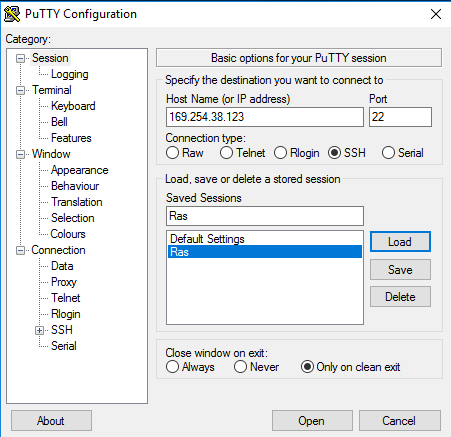
1. **Odaberite i upišite osnovne parametre veze.**

Nakon pokretanja programa, pojavit će se prozor u kojem je moguće odabrati osnovne postavke veze koja se želi uspostaviti.

Upišite ime udaljenog stroja i odaberite protokol SSH (kao na slici). Nakon toga kliknite na dugme **Open**.

Program će nakon toga otvoriti novi prozor i uspostaviti vezu s udaljenim računalom.

**Napomena**: Prilikom prvog uspostavljanja veze s nekim računalom, PuTTY od korisnika traži odobrenje da uspostavi odnos povjerenja s navedenim računalom. Da biste programu potvrdili da može nastaviti s radom, dovoljno je da kliknete na dugme **Ok**.



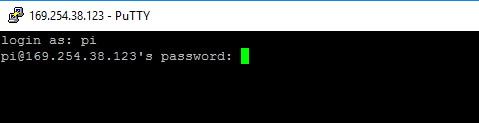
ime udaljenog stroja

dugme za uspostavljanje veze

odabir protokola

1. **Prijavite se na sustav.**

Nakon uspostavljanja veze s udaljenim računalom, potrebno je prijaviti se za rad na sustavu. Upišite svoje korisničko ime i lozinku.



Nakon uspješne prijave, sustav je ispisao nekoliko redaka teksta. Što možete zaključiti o sustavu na osnovi teksta koji je ispisao?

1. **Promijenite vrstu i veličinu slova koja se koriste za prikaz teksta u prozoru programa PuTTY.**

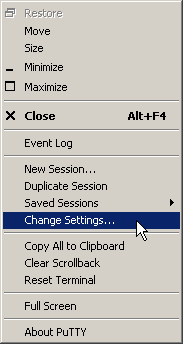
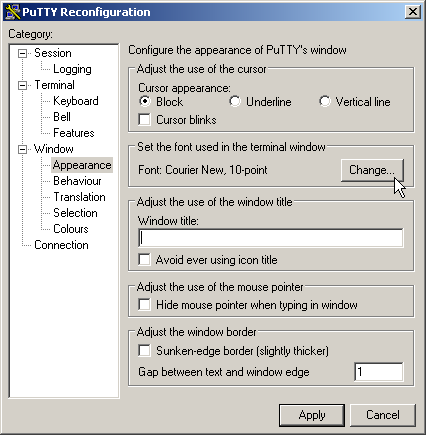
Kliknite desnom tipkom miša na gornji dio okvira radnog prozora programa PuTTY. Pojavit će se izbornik.

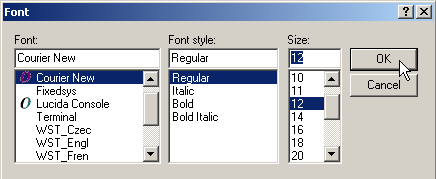
Odaberite **Change Settings**. Pojavit će se novi prozor **PuTTY Reconfiguration**.

U novom prozoru u okviru **Category** odaberite **Window/Appearance** i unutar okvira s naslovom **Set the font used in terminal window** kliknite na dugme **Change…**

U novom prozoru odaberite sljedeći font: **Courier New, Regular, 12 pt** i kliknite na **Ok**.

U prozoru **PuTTY Reconfiguration** kliknite na dugme **Apply** koje se nalazi na dnu prozora.



U novom prozoru odaberite sljedeći font: **Courier New, Regular, 12 pt** i kliknite na **Ok**.

U prozoru **PuTTY Reconfiguration** kliknite na dugme **Apply** koje se nalazi na dnu prozora.

## 1.5. Pitanja za ponavljanje

1. *Linux* je višekorisnički sustav. Što to znači?
2. Što je korisnički račun?
3. Što je potrebno upisati prilikom prijavljivanja na sustav?
4. Na koje se načine može raditi na *Linuxu*?
5. Po čemu se razlikuje tekstualno od grafičkog radnog okruženja?
6. Što su protokoli i čemu služe?
7. Koji se protokol danas najčešće koristi za rad na udaljenim sustavima i zašto?
8. Koje je podatke nužno poznavati da bi se moglo raditi na nekom udaljenom sustavu?
9. Ako je prijava na sustav neuspješna, koji su mogući razlozi za to?

# 2. Naredbe u *Linuxu*

## 2.1. Radni direktorij

Tijekom rada u *Linuxu* uvijek se nalazimo u nekom direktoriju (mapi). To znači da kada *Linuxu* zadajemo neku naredbu i ako nismo drugačije naveli, ona se odnosi upravo na taj direktorij i njegov sadržaj.

Taj se direktorij zove **radni direktorij**.

Na primjer, ako *Linuxu* zadamo da stvori neku datoteku, on će je stvoriti upravo u radnom direktoriju (osim ako nismo izravno naveli da to napravi u nekom drugom direktoriju).

Prvi radni direktorij u kojem se nađemo nakon prijave na sustav zove se **početni direktorij** (*home*).

Svaki korisnik ima svoj početni direktorij.

Više o direktorijima objašnjeno je u poglavlju *4. Sustav datoteka*.

## 2.2. Prompt

**Prompt** je niz znakova koje sustav ispisuje kao obavijest da je spreman za izvršenje sljedeće naredbe.

Uobičajeno se prompt sastoji od korisničkog imena, imena stroja i(li) imena radnog direktorija međusobno odijeljenih znakovima (separatorima) @, : i $.

Na primjer:

a01@192.168.1.108:/tmp$

* a01 je korisničko ime
* @ je znak kojim se razdvaja korisničko ime od ostatka prompta (prvi separator)
* 192.168.1.108 je ime stroja
* : je znak kojim se razdvaja ime stroja od ostatka prompta (drugi separator)
* /tmp je ime radnog direktorija
* $ je znak koji označava kraj prompta.

Izgled prompta moguće je mijenjati. Izmjene mogu raditi sistemski administratori (za sve korisnike na sustavu) i sami korisnici (za svoje potrebe).

Primjer vrlo jednostavnog prompta je: $.

Početni direktorij se često označava znakom **~** (tilda).

**Napomena**

Znak **~** (tilda) označava početni direktorij (*home*).

Na primjer, ako je naš početni direktorij /home/a01 i ako se nalazimo u njemu, tada će naš prompt umjesto

a01@192.168.1.108:/home/a01$

izgledati ovako:

a01@192.168.1.123:~$ .

## 2.3. Izgled naredbi

Naredbe u *Linuxu* upisujemo u naredbenu liniju iza prompta.

Kraj unosa naredbe označavamo pritiskom na tipku **[Return]**. Ta je tipka ujedno znak za kraj retka i prelazak u novi.

Osim u posebnim slučajevima (koji su objašnjeni u poglavlju *6. Ljuska Bash*), kraj retka ujedno predstavlja i kraj naredbe.

Primjer vrlo jednostavne naredbe je **date**. Ta naredba ispisuje datum i vrijeme:

$ date

Mon Sep 16 10:52:32 CEST 2013

$

Uočite da smo naredbu date upisali iza prompta (koji je u ovom slučaju vrlo jednostavan, samo znak $). Sustav je u novi red ispisao datum i vrijeme. Nakon toga je ponovno ispisao prompt kao znak da je završio s izvođenjem naše naredbe i da je spreman prihvatiti novu naredbu.

***Linux* razlikuje mala od velikih slova**. Tako su date i DATE (ili DaTe) različite naredbe.

**Napomena**

*Linux* razlikuje mala od velikih slova. U pravilu, sve se naredbe u *Linuxu* pišu malim slovima.

U pravilu, sve se naredbe u *Linuxu* pišu malim slovima.

Neke naredbe (većina njih) omogućavaju nam da upisivanjem dodatnih parametara mijenjamo način na koji se one izvode.

Parametri mogu biti **opcije** i **argumenti**.

**Napomena**

Ime naredbe određuje što treba napraviti, opcijama se određuje kako to treba napraviti, a argumentima se obično upućuje na objekt koji je predmet obrade.

Pojedine dijelove naredbi (ime, opcije i argumente) međusobno odvajamo pritiskom na razmaknicu.

Na primjer, upoznali smo naredbu date koja standardno ispisuje datum i vrijeme. Ako želimo izmijeniti datum i vrijeme na sustavu, također se možemo poslužiti naredbom date, ali tada moramo promijeniti način na koji se ona izvodi. To radimo tako da nakon imena naredbe dodamo dodatne parametre:

$ date -s "3 OCT 2013 18:00:00"  
Thu Oct 3 CEST 18:00:00  
$

Opcijom –s smo promijenili ponašanje naredbe tako da umjesto ispisa ona mijenja datum i vrijeme na sustavu, a argument "3 OCT 2013 18:00:00" predstavlja izmjenu koju želimo napraviti.

## 2.4. Promjena lozinke

Lozinka se mijenja naredbom **passwd**.

Prilikom izmjene prvo treba upisati staru lozinku, nakon toga treba upisati novu lozinku i potvrditi je ponovnim upisom.**Primjer:**

**Napomena**

Lozinke se tijekom upisa ne prikazuju na ekranu.

$ passwd  
Changing password for a01  
(current) UNIX password:  
New UNIX password:  
Retype new UNIX password:  
$

**Napomena**

Osnovne su smjernice za odabir lozinke:

* lozinke ne smiju sadržavati osobne podatke korisnika
* lozinke trebaju biti dugačke najmanje šest znakova
* lozinke trebaju sadržavati različite znakove – brojke i slova
* najbolje je da je odabir i redoslijed znakova u lozinci slučajan.

Administratori koji održavaju sustav mogu postaviti različita ograničenja na odabir lozinki. Administratori mogu:

* odrediti najmanju duljinu lozinke
* odrediti trajanje valjanosti lozinke
* odrediti složenost lozinke
* spriječiti ponavljanje lozinki ili njihovu međusobnu sličnost
* spriječiti da lozinke sadrže osobne podatke korisnika poput korisničkog imena ili osobnog imena.

U slučaju da se prilikom odabira nove lozinke ne poštuje neko od pravila koje je postavio administrator, sustav neće prihvatiti lozinku i tražit će upisivanje neke druge nove lozinke.

## 2.5. Odjavljivanje sa sustava

Naredba za odjavljivanje sa sustava je **logout**.

**Napomena**

Na kraju rada obavezno se treba odjaviti.

U određenim slučajevima moguće je odjaviti se i upisivanjem naredbe **exit** ili kombinacijom (istovremenim pritiskom) tipki **[Ctrl]** i **[D]** (ta se kombinacija obično piše kao **[Ctrl-D])**.

## 2.6. Naredbe echo i clear

Naredba **echo** na ekran ispisuje zadane argumente.

Naredba **clear** briše sadržaj ekrana.

**Sintaksa:**

echo [STRING …]

clear

**Primjer:**

$ echo Pozdrav svima

Pozdrav svima

$

## 2.7. Informacije o korisnicima na sustavu: who, w, i last

Naredba **who** ispisuje popis korisnika koji trenutačno rade na sustavu te podatke o tome od kada rade na sustavu i od kuda su se prijavili na sustav.

**who am i** ispisuje podatke samo o korisniku koji je pokrenuo naredbu.

Naredba **w** također ispisuje popis korisnika koji trenutačno rade na sustavu, ali s nekim dodatnim podacima (na primjer, kojim se programom korisnik trenutačno koristi i prije koliko je vremena upisao nešto na terminalu). Na vrhu ispisa bit će navedeni neki statistički podaci vezani uz sustav.

Naredbi w kao argument može se zadati ime nekog korisnika. Tada će se ispisati podaci samo za tog korisnika.

**Sintaksa:**

who [am i]

w [KORISNIK]

**Primjeri:**

$ who

Root pts/1 Sep 12 08:15 (192.168.1.108)

a01 pts/3 Sep 12 10:23 (192.168.1.108)

$ who am i

a01 pts/3 Sep 12 10:23 (192.168.1.108)

$

Naredba **last** ispisuje popis svih prijavljivanja na sustav.

**Napomena**

Zapisi o prijavama na sustav (*log*) povremeno se brišu.

Stoga se na kraju popisa koji ispisuje naredba last nalazi naveden podatak (datum i vrijeme) od kada su na sustavu zabilježeni podaci o prijavama.

Ukoliko se kao argument naredbe last navede neko korisničko ime, ispisat će se podaci samo za tog navedenog korisnika.

**Sintaksa:**

last [KORISNIK]

**Primjeri:**

$ last

$ last a01

## 2.9. Sustav pomoći: man i info

*Linux* ima dva sustava pomoći. Prvi sustav čine takozvane **man-stranice** (*man* je kratica od engleske riječi *manual*). Riječ je o dokumentaciji koja opisuje naredbe i druge dijelove sustava, a koja je podijeljena na poglavlja (sekcije) i stranice.

Navedena je dokumentacija na sustavu pohranjena u elektroničkom obliku, a dostupna je pomoću naredbe **man**.

Najjednostavniji način korištenja naredbe man je da se upiše njeno ime, a kao argument navede ime neke naredbe. U tom će nam slučaju sustav ispisati pripadajuću stranicu koja opisuje tu naredbu.

Neke naredbe ili općenito neke ključne riječi mogu biti opisane u više od jednog poglavlja (sekcije). Standardno će naredba man ispisati samo stranicu koju je prvo pronašla pretražujući dokumentaciju na sustavu.

Ako želimo da nam se ispišu sve stranice (iz različitih poglavlja), trebamo navesti opciju **–a**.

Ako želimo da nam se ispiše stanica iz nekog konkretnog poglavlja, tada kao argument trebamo navesti broj poglavlja.

Ako ne znamo ime naredbe za koju tražimo dodatne informacije, možemo pretraživati sustav pomoći pomoću ključne riječi. U tom slučaju kao opciju trebamo navesti **–k**, a kao argument željenu ključnu riječ. Rezultat će biti popis svih stranica koje u sebi sadrže navedenu ključnu riječ.

**Sintaksa:**

man [-a | POGLAVLJE] NAREDBA

man –k KLJUČNA\_RIJEČ

**Primjeri:**

$ man shutdown

$ man –a shutdown

$ man –k shutdown

Prilikom čitanja man-stranica možemo koristiti sljedeće tipke:

* kursorske tipke - za kretanje kroz ispisane stranice
* razmaknicu (**[Space]**) - za prelazak na sljedeću stranicu
* **[p]** - za povratak na prethodnu stranicu)
* **[q]** - za kraj ispisa.

Naredba **info** omogućava pristup do dokumenata koji čine drugi sustav pomoći, takozvane **info-stranice**. Ovaj se sustav pomoći sastoji od hipertekstualnih dokumenata. Dokumentaciju u ovom formatu imaju samo neki, većinom noviji programi posebno pisani upravo za *Linux*. Ukoliko ne postoji dokumentacija za traženi pojam među info-stranicama, naredba info će ispisati odgovarajuću man-stranicu (ako ona postoji).

**Sintaksa:**

info NAREDBA

**Primjer:**

$ info info

# 3. Sustav datoteka

## 3.1. Općenito o sustavu datoteka

**Sustav datoteka** je dio operacijskog sustava koji omogućava pohranjivanje i upravljanje datotekama i direktorijima.

**Datoteka** je logička cjelina koja služi za pohranjivanje podataka unutar sustava datoteka.

**Direktorij** (mapa) je posebna vrsta datoteke koja sadrži datoteke i druge direktorije (poddirektorije).

*Linux* ima **hijerarhijski sustav datoteka** (direktoriji mogu imati poddirektorije).

**Vršni direktorij** se još zove i *root*-direktorij i označava se znakom /.

Na *Linuxu* može postojati samo jedan sustav datoteka i samo jedan vršni direktorij. Ako na sustavu postoji više diskova ili diskovnih particija, oni su na *Linuxu* uključeni u jedan te isti sustav datoteka.

Na *Linuxu* su svi hardverski uređaji (pisači, mrežne kartice, uređaji za trajnu pohranu podataka, USB-priključci i drugo) također prikazani kao datoteke i može im se pristupati na isti način kao i običnim datotekama na disku. Operacijski sustav se brine o tome da se podaci prenesu navedenim uređajima na odgovarajući način.

Dakle, datoteke na *Linuxu* mogu biti:

* obične
* direktoriji
* specijalne (uređaji, cjevovodi za komunikaciju među procesima i drugo).

Sustav datoteka na *Linuxu* sastoji se od tisuća datoteka i direktorija. Uobičajena je organizacija sljedeća:

**Napomena**

Na različitim sustavima *Linux* (koji se popularno zovu distribucije) organizacija sustava datoteka može biti različita.

/bin - programi koje koriste i administratori i korisnici

/dev - datoteke koje predstavljaju hardverske uređaje

/etc - konfiguracijske datoteke

/home - osobni direktoriji korisnika

/sbin - sistemski programi

/tmp - privremene datoteke

/usr - korisnički programi, dokumentacija i biblioteke

/var - sistemski zapisi i druge promjenjive datoteke.

## 3.2. Imena datoteka

Datoteke na *Linuxu* mogu imati imena duljine do 255 znakova. Mogu se koristiti svi mogući znakovi, premda je preporučljivo da se koriste samo alfanumerički znakovi.

Ne preporuča se uporaba hrvatskih dijakritičkih znakova te metaznakova koji imaju posebno značenje za sustav:

**\* ? > < / ; ! [ ] | \ ' ″ ( ) { }**

**Primjeri dobro odabranih imena datoteka:**

moja\_datoteka

izvjestaj.txt

program3.c

primjer-123a

Poruka-za-Marka

.sifra

**Primjeri loše odabranih imena datoteka:**

moja datoteka

izvještaj.txt

+program.c

primjer\*123a

Račun za restoran ″Tri školjke″

izvještaj/poglavlje2

Svaki direktorij sadrži i dvije „posebne“ datoteke. To su **.** (točka) i .. (dvije točke).

. - oznaka za radni direktorij

- oznaka za nadređeni direktorij.

Imena datoteka koja počinju znakom . (na primjer: .profile) sustav smatra **skrivenim**. To znači da kad se, na primjer, ispisuje sadržaj direktorija pomoću naredbe ls i ne navedu se dodatne opcije (o čemu će biti riječi u nastavku ovog poglavlja) imena datoteka koja počinju znakom **.** se ne pojavljuju u popisu.

## 3.3. Ispis radnog direktorija: pwd

Naredba **pwd** ispisuje koji je naš radni direktorij.

**Sintaksa:**

pwd

**Primjer:**

$ pwd

/home/a01

$

## 3.4. Apsolutne i relativne staze

**Staza** (*pathname*) je niz imena odijeljenih znakom / (kosa crta, *slash*) koja opisuju put ili rutu koju je potrebno proći kroz sustav datoteka da bi se došlo do željene datoteke ili direktorija.

Posljednje ime u stazi može biti ime datoteke ili direktorija. Sva ostala imena u stazi moraju biti nazivi direktorija.

Svaka staza koja ne počinje znakom / je **relativna** i opisuje put od radnog direktorija do krajnjeg cilja.

Staze koje počinju znakom / su **apsolutne** i opisuju rutu od vršnog direktorija do krajnjeg cilja.

Ako se koristi na početku staze, znak **~** (tilda) označava početni direktorij.

**Potpuno ime** neke datoteke ili direktorija je apsolutna staza do te datoteke koja završava s (kratkim) imenom te datoteke ili direktorija. Na sustavu ne može postojati više datoteka (ili direktorija) s istim potpunim imenom.

**Primjeri apsolutnih i relativnih staza**:

Pretpostavimo da sljedeća slika opisuje naš sustav datoteka i da je naš radni direktorij /home/tecaj03 (označen sjenčanjem):

**tecaj03**

**/**

**tmp**

**home**

**tecaj01**

**tecaj02**

**f1**

**f1**

**f1**

**f1**

**zadaci**

**f1**

1

2

3

4

U sljedećoj se tablici nalaze relativne i apsolutne staze za četiri označena slučaja:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Cd** |  | **Apsolutna staza** |
| **1.** | f1 | **1.** | cd /home/a03/f1 |
| **2.** | zadaci/f1 | **2.** | /home/tecaj03/zadaci/f1 |
| **3.** | ../tecaj01/f1 | **3.** | /home/tecaj01/f1 |
| **4.** | ../../tmp/f1 | **4.** | /tmp/f1 |

## ljuska 3.5. Promjena radnog direktorija: cd

Naredba **cd** omogućava promjenu radnog direktorija.

Dovoljno je kao argument navesti ime direktorija za koji želimo da bude novi radni direktorij.

Vrijede sljedeći **posebni slučajevi**:

* ako se kao argument navede znak **-** (minus), tada će se za novi radni direktorij postaviti direktorij koji je bio radni direktorij neposredno prije trenutačnog
* ako se ne navede nikakav argument, tada će se za novi radni direktorij postaviti početni direktorij.

Ime novog radnog direktorija može se zadati bilo relativnom, bilo apsolutnom stazom.

**Sintaksa:**

cd [DESTINACIJA]

**Primjeri:**

$ cd /tmp

$ cd /usr

$ pwd

/usr

$cd –

$ pwd

/tmp

$ cd

$ pwd

cd /home/a01

$

## 3.6. Ispis sadržaja direktorija: ls

Naredba **ls** ispisuje sadržaj direktorija. Dodatni podaci koji se pri tome ispisuju određuju se opcijama.

**Sintaksa:**

ls [opcije] [DIREKTORIJ\_ILI\_DATOTEKA]

Ako se umjesto imena direktorija navede ime neke datoteke, ispisat će se podaci o toj datoteci.

Ako se ne navede nikakav argument, podrazumijeva se da se radi o radnom direktoriju. Moguće je zadati i više argumenata (popis direktorija čiji sadržaj želimo ispisati).

Najčešće korištene **opcije** naredbe ls su sljedeće:

**-l** - ispis dodatnih podataka o datotekama i poddirektorijima

**-a** - ispis svih datoteka i direktorija, uključivši i one skrivene (čija imena počinu točkom)

**-d** - umjesto sadržaja ispisuje svojstva direktorija (najčešće se koristi zajedno s opcijom –l)

**-F** - na kraj imena direktorija dodaje znak /, a na kraj imena datoteke koja se može izvoditi dodaje znak \*

**-R** - rekurzivno ispisuje sadržaj zadanog direktorija i svih poddirektorija.

Više opcija moguće je navesti zajedno (na primjer: ls –al).

**Primjeri:**

$ ls

doc test

$ ls -a

.profile doc test

$ ls abc

ls: cannot access abc: No such file or directory

$ ls /

bin etc lib tmp var

dev home sbin usr

$

**Primjer ispisa naredbe ls:**

$ ls –la

drwxr-x--- 5 tecaj01 tecaj 4096 Sep 29 16:22 .

drwxr-xr-x 26 root root 4096 Aug 10 09:27 ..

-rw-r--r-- 1 tecaj01 tecaj 191 Feb 7 2005 .bash\_profile

drwx------ 3 tecaj01 tecaj 4096 Sep 29 16:22 privatno

-rw-r--r-- 1 tecaj01 tecaj 2332 Oct 21 2005 zadatak.txt

$

B

C

A

**A** - veličina u bajtovima

**B** - datum/vrijeme zadnje modifikacije

**C** - ime

Ispis naredbe ls bit će detaljnije objašnjen u sljedećem poglavlju.

## 3.7. Stvaranje i uklanjanje direktorija: mkdir i rmdir

Naredba **mkdir** stvara, a **rmdir** uklanja direktorije sa sustava.

Da bi se direktorij mogao obrisati on mora biti prazan (ne smije sadržavati neki poddirektorij ili datoteku).

Radni direktorij (direktorij u kojem se trenutačno nalazimo i iz kojeg pozivamo naredbu rmdir) ne može se obrisati.

Obje naredbe mogu se zadati s opcijom **–p**. U tom slučaju stvorit će se ili ukloniti svi direktoriji navedeni kao dio puta koji je naveden kao argument.

Na primjer, naredba mkdir –p test/abc prvo će, ako on ne postoji, stvoriti direktorij test, a nakon toga u njemu poddirektorij abc.

**Sintaksa:**

mkdir [-p] DIREKTORIJ

rmdir [-p] DIREKTORIJ

**Primjeri:**

$ mkdir doc

$ mkdir test/abc

mkdir: cannot create directory 'test/abc': No such file or directory

$ mkdir –p test/abc

$ rmdir test

rmdir: test: Directory not empty

$ rmdir doc

$

## 3.8. Stvaranje datoteka: touch

Naredba **touch** stvara datoteku sa zadanim imenom. Tako nastale datoteke su prazne, veličine 0 bajtova.

Ako datoteka čije je ime zadano kao argument već postoji, promijenit će se samo datum njene zadnje modifikacije. Sadržaj datoteke neće biti izmijenjen.

**Sintaksa:**

touch DATOTEKA

**Primjer:**

$ ls testcd

ls: test.txt: No such file or directory

$ touch test.txt

$ ls test.txt

test.txt

$

## 3.9. Brisanje datoteka: rm

**Napomena**

**Obrisane datoteke (i direktorije) više nije moguće vratiti.**

Naredba **rm** uklanja datoteke sa sustava.

**Sintaksa:**

rm [-i] DATOTEKA

rm –r[i] DIREKTORIJ

Opcijom **–r** zadaje se rekurzivno uklanjanje datoteka i direktorija. Na ovaj način možemo ukloniti i direktorije koji nisu prazni.

Ukoliko se zada opcija **–i**, naredba rm će prije uklanjanja bilo koje datoteke zatražiti potvrdu. Na nekim je sustavima ova opcija postavljena kao podrazumijevana (nije je potrebno posebno navoditi). U tom se slučaju njezin učinak poništava navođenjem opcije –f.

**Primjeri:**

$ rm test

rm: cannot remove 'tmp': Is a directory

$ rmdir test

rmdir: test: Directory not empty

$ rm –ir test

$

## 3.10. Kopiranje i premještanje datoteka i direktorija: cp i mv

Naredba **cp** služi za kopiranje datoteka i direktorija.

Pri tome vrijedi sljedeće:

* treba navesti barem dva argumenta: izvor i destinacija
* ako se kopira samo jedna datoteka tada destinacija može biti ime nove datoteke ili ime direktorija u koji se kopira izvorna datoteka
* ako je kao cilj zadan direktorij, izvorna će se datoteka kopirati u zadani direktorij i sačuvati originalno ime
* kada se odjednom kopira više datoteka, kao cilj se mora zadati direktorij
* datoteka ne može biti kopirana u samu sebe
* direktoriji se kopiraju navođenjem opcije **–r**.

**Sintaksa:**

cp DATOTEKA DESTINACIJA

cp DATOTEKA [DATOTEKA …] DIREKTORIJ

cp –r DIREKTORIJ DESTINACIJA

Naredba **mv** služi za premještanje ili preimenovanje datoteka i direktorija.

Vrijedi sljedeće:

* treba navesti barem dva argumenta: izvor i destinaciju premještanja ili preimenovanja
* ako je prvi argument ime datoteke tada destinacija može biti novo ime datoteke ili ime direktorija u koji se premješta izvorna datoteka:
  + ako je kao destinacija zadan direktorij, izvorna će se datoteka premjestiti u zadani direktorij i sačuvati originalno ime
  + ako je kao destinacija zadano novo ime datoteke, tada će se originalna datoteka preimenovati ako je cilj u istom direktoriju ili premjestiti i dobiti novo ime ako je cilj u drugom direktoriju
* kada se odjednom premješta više datoteka, kao destinacija se mora zadati direktorij
* za premještanje i preimenovanje direktorija vrijede ista pravila kao i za datoteke.

**Sintaksa:**

mv DATOTEKA DESTINACIJA

mv DATOTEKA [DATOTEKA …] DIREKTORIJ

mv DIREKTORIJ DESTINACIJA

## 3.11. Ispis sadržaja datoteka: cat, more i less STALI

Naredbe **cat, more** i **less** ispisuju sadržaj datoteka.

**Sintaksa:**

cat DATOTEKA [DATOTEKA...]

more DATOTEKA [DATOTEKA...]

less DATOTEKA [DATOTEKA...]

Naredba cat je najjednostavnija. Ona redom ispisuje sadržaj datoteke ili datoteka koje su navedene kao argumenti.

Naredbe more i less ispisuju sadržaj datoteka koje su navedene kao argumenti ekran po ekran. Prilikom pregledavanja sadržaja datoteka moguće je koristiti sljedeće komande (tipke na tipkovnici):

**[q]** - završetak rada

**[Space]** - prikaz sljedeće stranice

**[Return]** - prikaz sljedećeg reda

**[b]** - prethodna stranica

**[/]** - pretraživanje.

## 3.12. Zamjenski znakovi

Najvažniji zamjenski znakovi su:

**\*** - zamjenjuje bilo koji broj znakova

**?** - zamjenjuje točno jedan znak

**[ ]**  - označava zamjenu za točno jedan znak koji odgovara znakovnom nizu zadanom unutar zagrada

Nizovi se zadaju nabrajanjem ili kao intervali.

Unutar zagrada moguće je koristiti i znak negacije **!** (uskličnik).

Zamjenski se znakovi mogu koristiti prilikom zadavanja argumenata u naredbama. Na primjer, naredba ls a\* će ispisati imena svih datoteka u radnom direktoriju čija imena započinju slovom a.

**Primjer:**

Pretpostavimo da se u našem radnom direktoriju nalazi sljedećih devet datoteka:

a, a1, a2, a3, ab, abc, b1, b2, c1 i d3.

Sljedeća tablica pokazuje što bi ispisala naredba ls u kombinaciji s nekoliko različitih argumenata zadanih pomoću zamjenskih znakova:

|  |  |  |
| --- | --- | --- |
| **naredba** | **ispisuje** | **ne ispisuje** |
| **ls a\*** | a a1 a2 a3 ab abc | b1 b2 c1 d3 |
| **ls a?** | a1 a2 a3 ab | a abc b1 b2 c1 d3 |
| **ls [ad]?** | a1 a2 a3 ab d3 | b1 b2 c1 a abc |
| **ls [a-c]?** | a1 a2 a3 ab b1 b2 c1 | d3 a abc |
| **ls [a]!** | b1 b2 c1 d3 | a a1 a2 a3 ab abc |

## 3.13. Pretraživanje sustava datoteka: find

Sustav datoteka se pretražuje naredbom **find**.

**Sintaksa:**

find [STAZA...] [UVJET]

STAZA je direktorij od kojeg počinje pretraživanje, a UVJET je izraz kojim se određuje koje se datoteke traže. Najčešće je to –name (pretraživanje po imenu datoteke).

**Primjeri:**

**Napraviti u svome home dir imena abc te unutar njega datoteku imena program.c**

$ find /home/uce27 -name cpu.txt

$ find . –name program.c

# 4. Prava pristupa datotekama i direktorijima

## 4.1. Naredba id

Svakom korisniku dodijeljen je jedinstven **korisnički broj** (*user ID*, **uid**).

Također, svaki je korisnik pridružen jednoj ili više **skupina korisnika.** Skupine također imaju svoje jedinstvene brojeve(*group ID*, **gid**).

Skupine služe za lakše dodjeljivanje istih korisničkih prava većem broju korisnika.

Svaki je korisnik pridružen barem jednoj skupini i ta je skupina za tog korisnika osnovna skupina kojoj pripada.

Korisnički broj je osnovni način na koji sustav razlikuje korisnike. Korisnički broj, osnovnu skupinu i pripadnost drugim skupinama određuje administrator sustava.

Naredba **id** (bez dodatnih argumenata) ispisuje pripadajući korisnički broj, broj osnovne skupine i popis svih skupina kojima pripada korisnik koji je pokrenuo naredbu.

Ako se kao argument zada ime nekog korisnika, tada se ispisuju podaci o tom korisniku.

**Sintaksa:**

id [KORISNIK]

**Primjeri:**

$ id

uid=500(tecaj01) gid=900(tecaj) groups=900(tecaj)

$ id tecaj02

uid=501(tecaj02) gid=900(tecaj) groups=900(tecaj)

## 4.2. Razine prava pristupa: *user*, *group* i *other*

Svaka datoteka (ili direktorij) ima svog vlasnika. Vlasnik je obično korisnik koji je stvorio tu datoteku.

Vlasnik je također korisnik koji određuje prava pristupa za navedeni objekt.

Svaka datoteka ili direktorij pripadaju i nekoj skupini korisnika. Najčešće je to ista skupina kojoj pripada i vlasnik, ali ne i nužno. Vlasnik određuje kojoj će skupini pripadati njegove datoteke i direktoriji.

*Linux* omogućava **tri razine dodjeljivanja prava pristupa** (**ugo**):

**u** - pravo pristupa koje se odnosi na vlasnika (***u****ser*)

**g** - pravo pristupa koje se odnosi na skupinu (***g****roup*)

**o** - pravo pristupa koje se odnosi na ostale (***o****ther*)

Za svaku od navedenih razina moguće je zasebno određivati prava pristupa.

## 4.3. Vrste prava pristupa: *read*, *write* i *execute*

Prava pristupa na datoteke i direktorije koja se mogu dodijeli korisnicima su:

**r** - pravo čitanja (***r****ead*)

**w** - pravo izmjene sadržaja (***w****rite*)

**x** - pravo izvršavanja (*e****x****ecute*).

Konkretan smisao navedenih prava pristupa ovisi o vrsti objekta na koji se odnosi (tj. je li riječ o datoteci ili direktoriju):

|  |  |  |  |
| --- | --- | --- | --- |
|  | **read** | **write** | **execute** |
| **datoteke** | Sadržaj se može čitati (more, cat). | Sadržaj se može mijenjati (u datoteku se može pisati). | Datoteka se može koristiti kao naredba. |
| **direktoriji** | Sadržaj se može čitati (ls). | Sadržaj se može mijenjati (rm, cp, mv). | Može postati radni direktorij (cd). |

Prava pristupa možemo pročitati iz ispisa koji daje naredba ls –l.

**Primjer:**

$ ls –la

drwxr-x--- 5 tecaj01 tecaj 4096 Sep 29 16:22 .

5

7

8

2

4

6

1

3

drwxr-xr-x 26 root root 4096 Aug 10 09:27 ..

-rw-r--r-- 1 tecaj01 tecaj 191 Feb 7 2005 .bash\_profile

drwx------ 3 tecaj01 tecaj 4096 Sep 29 16:22 privatno

-rw-r--r-- 1 tecaj01 tecaj 2332 Oct 21 2005 zadatak.txt

**1** - vrsta objekta (znak - označava obične datoteke, a   
slovo d označava direktorije)

**2** - prava pristupa za vlasnika, skupinu i ostale korisnike - tri skupine prava pristupa sa po tri oznake (**rwx**)

**3** - broj poveznica (sistemski podatak)

**4** - ime vlasnika

**5** - ime skupine

**6** - veličina u bajtovima

**7** - datum i vrijeme zadnje modifikacije

**8** - ime datoteke ili direktorija

Opis prava pristupa za vlasnika, skupinu i ostale korisnike je oblika:

**rwx rwx rwx**

Prva tri stupca odnose se na prava pristupa vlasnika, sljedeća tri na skupinu, a posljednja tri na ostale korisnike. Svaki od stupaca odgovara jednom od prava pristupa (***r****ead*, ***w****rite* ili *e****x****ecute*). Ako je određeno pravo pristupa dodijeljeno (omogućeno), tada se na odgovarajućem mjestu nalazi slovo koje odgovara toj vrsti pristupa (**r**, **w** ili **x**). U suprotnom se nalazi znak **–** (crta).

**Primjeri prava pristupa:**

rwxrwxrwx - svi imaju pravo čitanja, pisanja i izvršavanja

rwxr-xr-x - vlasnik ima sva prava, a skupina i svi ostali imaju samo prava čitanja i izvršavanja

rw-r----- - vlasnik ima pravo čitanja i pisanja, skupina ima samo pravo čitanja, a ostali korisnici nemaju nikakvo pravo pristupa.

## 4.4. Postavljanje prava pristupa pomoću simboličkih oznaka

Prava pristupa postavljamo naredbom **chmod**. Ova naredba omogućava postavljanje prava pristupa na dva načina. Prvi je način pomoću **simboličkih oznaka** za razine i vrste prava pristupa, a koje smo već imali prilike upoznati:

* **u**, **g** i **o** za razine pristupa (te **a** koji označava sve razine)
* **r**, **w** i **x** za vrste prava pristupa.

Vrsta izmjene se zadaje operatorima: **+** (dodaj), **-** (oduzmi) i **=** (izjednači).

**Sintaksa:**

chmod POSTAVKA [, POSTAVKA...] DATOTEKA

|  |  |  |
| --- | --- | --- |
| POSTAVKA | | |
| **tko** | **vrsta promjene** | **vrsta pristupa** |
| **u** (vlasnik)  **g** (grupa)  **o** (ostali)  **a** (svi) | **+** (dodaj)  **-** (oduzmi)  **=** (izjednači) | **r** (čitanje)  **w** (pisanje)  **x** (izvršavanje) |

Različite postavke je moguće ulančavati, pri čemu se kao separator koristi zarez. Lijeva i desna strana postavke može sadržavati i više od jednog simbola.

Najjednostavniji oblik zadavanja promjene prava pristupa je niz od tri znaka, gdje prvi znak označava na koga se promjena odnosi, drugi je znak operator koji označava o kakvoj se promjeni radi, a treći je znak vrsta pristupa koja se mijenja (na primjer: u+x – vlasniku se dodaje pravo čitanja datoteke).

**Primjeri:**

$ chmod a=r abc

$ ls –l abc

-r--r--r-- 1 tecaj01 tecaj 2332 Oct 21 2013 abc

$ chmod u+wx abc

$ ls –l abc

-rwxr--r-- 1 tecaj01 tecaj 2332 Oct 21 2013 abc

$ chmod g+x,o-r abc

$ ls –l abc

-rwxr-x--- 1 tecaj01 tecaj 2332 Oct 21 2013 abc

$

|  |  |  |
| --- | --- | --- |
| **prije** | **naredba** | **poslije** |
| rw- r-- --- | **chmod u-w,g-r**  **chmod a+x**  **chmod ug+x**  **chmod a=rx** | r-- --- ---  rwx r-x --x  rwx r-x ---  r-x r-x r-x |

## 4.5. Postavljanje prava pristupa pomoću numeričkih oznaka (stali 27.9.)

Drugi način postavljanja prava pristupa je preko numeričkih oznaka.

Svakoj vrsti pristupa dodijeljena je jedna numerička vrijednost:

**r**ead = 4

**w**rite = 2

e**x**ecute = 1.

Pravo pristupa za svaku od tri razina pristupa (*user*, *group*, *other*) određuje se zbrajajući pripadajuće vrijednosti načina pristupa: ako je odgovarajuće pravo pristupa dozvoljeno, tada se ukupnom zbroju dodaje pripadajuća numerička vrijednost. Na taj način se svaka kombinacija prava pristupa može jednoznačno predstaviti brojem iz intervala od 0 do 7.

Konačan rezultat je niz od tri jednoznamenkasta broja kojim određujemo prava pristupa za sva tri nivoa.

**Sintaksa:**

chmod POSTAVKA DATOTEKA

**Primjer:**

Pretpostavimo da želimo postaviti prava pristupa na neku datoteku tako da vlasnik ima sva prava nad datotekom (može je čitati, pisati u nju i izvršavati), a da skupina i svi ostali korisnici datoteku mogu samo čitati i izvršavati.

Numeričku oznaku izračunat ćemo na sljedeći način:

Postavke za vlasnika datoteke: rwx = 4+2+1 = 7.

Postavke za skupinu: r-x = 4 + 1 = 5.

Postavke za ostale: r-x = 4 + 1 = 5.

Konačan rezultat je 755, a naredba glasi: chmod 755 DATOTEKA.

|  |  |  |
| --- | --- | --- |
| **rwx r-x r-x** | | |
| **vlasnik** | **skupina** | **ostali** |
| rwx  ↓  4+2+1  ↓  7 | r-x  ↓  4+0+1  ↓  5 | r-x  ↓  4+0+1  ↓  5 |
| chmod **755** DATOTEKA | | |

# 5. Uređivač teksta *GNU nano*

## 5.1. Općenito o uređivačima teksta

**Napomene**

Uređivači teksta (editori) služe za pisanje i osnovno uređivanje teksta, poput pisanja programa, uređivanja konfiguracijskih datoteka i slično.

Uređivači teksta nisu isto što i programi za obradu teksta (poput programa *MS Word*).

**Uređivači teksta** (editori) su programi koji služe za pisanje i osnovno uređivanje teksta. Oni ne posjeduju napredne mogućnosti poput određivanja vrste fonta, veličine slova ili proreda između redaka, a koje imaju programi za obradu teksta.

Osnovne mogućnosti uređivača teksta su stvaranje, učitavanje i pohranjivanje tekstualnih datoteka, upisivanje i izmjena teksta te jednostavno pretraživanje teksta.

Na *Linuxu* možemo koristiti više uređivača teksta. Standardni uređivač teksta koji je sastavni dio svakog sustava je ***Vi***. Riječ je o kompleksnom programu teškom za uporabu. Jednostavniji za uporabu je uređivač ***GNU nano***. Ovaj je program u pravilu standardan dio svake distribucije *Linuxa*.

Osim ova dva uređivača teksta postoji i veliki broj drugih različitih uređivača teksta (jedan od popularnijih je program ***Joe***). Neki od njih rade u tekstualnom okruženju poput programa *Vi* i *GNU nano*, a neki rade u grafičkom okruženju (na primjer ***Kate*** u okruženju KDE, a ***Gedit*** u okruženju *Gnome*).

## 5.2. Osnovna svojstva uređivača teksta *GNU nano*

**Napomena**

Program ***GNU nano*** standardno je dostupan na velikoj većini sustava zasnovanih na *Linuxu*. Ukoliko to nije slučaj sa sustavom na kojem radite, tada trebate zamoliti sistemskog administratora da instalira navedeni program.

Prije toga možete provjeriti postoji li na sustavu editor *Pico* – rad s tim editorom je gotovo identičan radu s editorom *GNU nano*.

Uređivač teksta *GNU nano* je jednostavan program. Da bi se moglo raditi s njime dovoljno je naučiti mali broj osnovnih naredbi koje se pokreću odgovarajućim kombinacijama tipki. Stoga je i neiskusnim korisnicima dovoljno samo kratko vrijeme uvježbavanja prije početka korištenja.

*GNU nano* je „klon“ jednostavnog editora ***Pico*** koji je bio dio programskog paketa za rad s elektroničkom poštom *Pine* i tako je stekao velik broj korisnika. Neki inačice sustava *Linux* umjesto editora *GNU nano* i dalje sadrže editor *Pico*. Sve funkcije editora *GNU nano* koje se spominju u ovom tečaju identične su funkcijama editora *Pico*.

Razlog za nastanak editora *GNU nano* je to što, premda je editor *Pico* u svojoj osnovi besplatan i slobodno dostupan softver, licenca kojom je određeno pravo uporabe programa *Pico* nije sukladna licenci GPL (*General Public License*).

Osim kopiranja gotovo svih funkcija programa *Pico*, program *GNU nano* sadrži i brojna proširenja.

## 5.3. Pokretanje programa

Program *GNU nano* se pokreće upisivanjem naredbe **nano**. Ukoliko se naredba upiše bez dodatnih argumenata, na ekranu će se pokazati prazna stranica u koju odmah možemo početi pisati tekst. Ukoliko se kao argument navede ime neke datoteke, *GNU nano* će učitati sadržaj te datoteke.

**Sintaksa**:

nano [DATOTEKA]

**Primjeri:**

$ nano

$ nano .profile

Unutar teksta obično se možemo kretati pomoću kursorskih tipki koje se nalaze na tipkovnici. Ipak, može se dogoditi da sustav ne prepoznaje kursorske tipke. U tom slučaju možemo koristiti sljedeće kombinacije tipki:

**[Ctrl-B]** - lijevo

**[Ctrl-F]** - desno

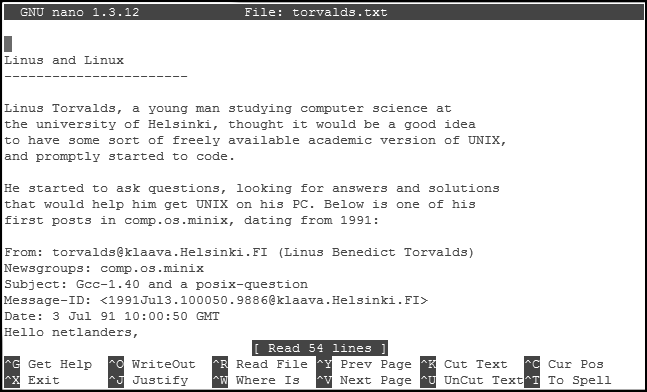
**[Ctrl-P]** - gore

**[Ctrl-N]** - dolje

Ako želimo obrisati dio teksta, to možemo učiniti pomoću za to uobičajenih tipki na tipkovnici: **[Delete]** i **[Backspace]**. Ukoliko sustav na kojem radimo ne prepoznaje navedene tipke, umjesto tipke [Delete] možemo koristiti kombinaciju tipki **[Ctrl-D]**.

Nakon pokretanja programa *GNU nano* na dnu će se ekrana prikazati popis najvažnijih naredbi. Kako bi povećao prostor za unos teksta, taj se popis može ukloniti upisivanjem kombinacije tipki **[Alt-X]**. Istom se kombinacijom tipki popis može ponovno vratiti na ekran.

Interne upute za rad s programom mogu se dobiti upisivanjem kombinacije tipki **[Ctrl-G]**. Upute sadrže osnovne informacije o programu i popis svih naredbi koje se mogu koristiti. Iz uputa se u normalno radno okruženje program *GNU nano* izlazi kombinacijom tipki **[Ctrl-X]**.



*Slika: Prikaz radnog ekrana programa GNU nano. Na dnu ekrana se vidi popis osnovnih naredbi.*

## 5.4. Osnovne komande

Naredbe se u programu *GNU nano* zadaju odgovarajućim kombinacijama tipki.

|  |
| --- |
| **[Crtl-O]** - pohranjivanje teksta u datoteku  **[Crtl-X]** - završetak rada  **[Crtl-R]** - učitavanje teksta iz datoteke  **[Crtl-K]** - izrezivanje cijelog retka (*cut*)  **[Crtl-U]** - umetanje izrezanog retka (*paste*)  **[Crtl-W]** - pretraživanje teksta  **[Crtl-G]** - prikazivanje uputa za rad s programom |

**Pohranjivanje teksta u datoteku**

Tekst se nakon upisa ili izmjene može pohraniti u neku datoteku. Komanda za to je **[Crtl-O]**. Nakon toga treba potvrditi ponuđeno ime datoteke ili upisati novo.

**Završetak rada**

Rad u programu GNU nano završava se komandom **Ctrl-X**. Ukoliko tekst na kojem smo radili nije pohranjen u datoteku, GNU nano će prije završetka rada upitati želimo li ga pohraniti u datoteku.

**Učitavanje teksta iz datoteke**

Tekst se pohranjen u nekoj datoteci može učitati direktno u uređivač teksta naredbom **[Ctrl-R]**. Nakon toga treba upisati ime datoteke ili odabrati je iz interaktivnog kataloga koji se pokreće komandom **[Ctrl-T]**.

Učitani tekst će biti umetnut u postojeći na mjestu na kojem se prilikom pokretanja naredbe za učitavanje iz datoteke nalazio kursor.

**Izrezivanje i umetanje teksta**

Komanda **[Ctrl-K]** će izrezati cijeli redak u kojem se nalazi kursor i pohraniti ga u međumemoriju. Više redaka možete pohraniti u međumemoriju višestrukim uzastopnim pritiskom na **[Ctrl-K]**.

Komanda **[Ctrl-U]** će umetnuti tekst iz međuspremnika na mjesto gdje se u tekstu nalazi kursor. Kopiranje se može ponoviti proizvoljan broj puta.

Ako se želi izrezati samo dio retka, potrebno je postaviti kursor na početak dijela koji se želi izrezati i pritisnuti **[Ctrl-^]** (ili **[Alt-A]**). Nakon toga treba pomaknuti kursor na kraj teksta koji se želi izrezati i pritisnuti **[Ctrl-K]** (slovo ispod kursora neće biti izrezano).

Kopiranje teksta može se napraviti tako da se tekst koji se želi kopirati prvo izreže, a onda se pritiskom na **[Ctrl-U]** ponovno umetne na mjesto s kojeg je izrezan te nakon toga pozicioniranjem kursora na željenom mjestu i ponovnim pritiskom na **[Ctrl-U]** umetne tamo gdje se želi kopirati.

**Pretraživanje teksta**

Tekst se pretražuje tako da se pritisne **[Ctrl-W]**. Nakon toga se u statusnu liniju upiše uzorak koji se želi pronaći u tekstu. Ukoliko se traženi uzorak nalazi u tekstu, kursor će se postaviti na mjesto njegovog prvog pojavljivanja iza pozicije s koje se pokrenulo pretraživanje.

# 6. Ljuska *Bash*

## 6.1. Što je ljuska?

Ljuska je interpreter naredbi. Svaki put kad upišemo neku naredbu, ljuska je preuzima, obrađuje i izvršava.

Korisnicima je na sustavima zasnovanim na *Linuxu* obično na raspolaganju više različitih ljuski. Standardna je ljuska ***Bash*** (*Bourne Again Shell*).

Ljuske se međusobno razlikuju po sintaksi ugrađenog interpreterskog jezika, opcijama koje podržavaju, konfiguracijskim datotekama i drugim detaljima.

## 6.2. Upis naredbi

Ljuska *Bash* standardno sve što se upiše u jedan redak smatra jednom naredbom.

Ako želimo više naredbi upisati u jedan redak, tada ih trebamo odvojiti znakom **;** (točka-zarez).

Ako jednu naredbu želimo napisati u više redaka tada na kraj retka trebamo upisati znak **\** (*backslash*).

Prilikom pisanja naredbi u više redova, nakon prvog reda pojavit će se **sekundarni prompt**. To je obično znak **>** (veće) .

**Primjeri:**

$ echo Jedan ; echo Dva

Jedan

Dva

$ echo Jedan \

> Dva

Jedan Dva

$

## 6.3. Vrste naredbi

Naredbe mogu biti:

* ugrađene u ljusku
* izvršne datoteke
* aliasi.

Prilikom pokretanja neke naredbe prvo što ljuska radi je otkrivanje tipa naredbe. Ukoliko se radi o **izvršnoj datoteci**, ljuska je treba pronaći na disku kako bi je mogla pročitati i izvršiti. Da bi ljuska mogla pronaći naredbu koja je izvršna datoteka, naredba mora biti zadana stazom koja vodi do nje ili se mora nalaziti u direktoriju koji je naveden u varijabli **PATH**.

Vrijednost varijable PATH čini niz imena direktorija koji su međusobno odijeljeni znakom **:** (dvotočka).

**Primjer:**

$ echo $PATH

/home/tecaj00/bin:/usr/local/bin:/usr/bin:/bin:.

↑

*uočite da je u ovom primjeru dio popisa direktorija u kojima ljuska provjerava postoji li zadana naredba direktorij . (radni direktorij)*

**Aliasi** su naredbe koje su definirane kao sinonimi ili zamjenska imena za neke druge naredbe. Naredba **alias** ispisuje popis aliasa i omogućava njihovo postavljanje.

**Sintaksa:**

alias [ALIAS=NAREDBA]

**Primjer:**

$ alias ← *ispisujemo popis aliasa*

alias md='mkdir'

alias rd='rmdir' ← *u ovom su nam trenutku dostupna dva*

*aliasa:* md *i* rm

$ moj\_ls ← *pokušavamo pokrenuti naredbu* moj\_ls

-bash: moj\_ls: command not found

↑

*sustav nam javlja da naredba* moj\_ls *ne postoji*

$ alias moj\_ls='ls -a'

↑

*definiramo alias* moj\_ls *kao* ls -a

$ moj\_ls ← *ponovno pokušavamo pokrenuti* moj\_ls

.bash\_history a b c

↑

*ovaj smo put dobili smo rezultat koji smo željeli*

$

Naredba **type** ispisuje kojeg je tipa naredba.

**Sintaksa:**

type NAREDBA

**Primjer:**

$ type pwd

pwd is a shell builtin ← pwd *je naredba ugrađena u*

*ljusku*

$ type cat

cat is /bin/cat ← cat *je izvršna datoteka na*

*disku (u direktoriju* /bin*)*

$ type rd

rd is aliased to 'rmdir' ← rd *je alias (kratica za* rmdir*)*

Naredba **which** ispisuje stazu do naredbi koje su izvršne datoteke.

Da bi naredba which mogla pronaći zadanu naredbu na disku, potrebno je da se ona nalazi u direktoriju koji je naveden u varijabli PATH.

**Sintaksa:**

which NAREDBA

**Primjer:**

$ which info

/usr/bin/info

$

# 7. Procesi

## 7.1. Općenito o procesima

**Procesi** su programi koji se izvršavaju.

**Programi** su izvršene datoteke.

Prilikom pokretanja sustav svakom procesu dodjeljuje njegov broj (**PID**). Taj je broj jedinstven. Sustav također uz svaki proces bilježi i broj procesa koji je pokrenuo (**PPID**), kao i njegov radni direktorij.

## 7.2. Popis procesa: ps i top

Naredba **ps** ispisuje popis procesa na sustavu.

**Sintaksa:**

ps [-f] [-e | -u KORISNIK]

top

Osnovne opcije su sljedeće:

**-e** - ispis svih procesa na sustavu (u suprotnom će naredba ps ispisati samo procese koji su vezani uz ljusku u kojoj se radi)

**-f** - ispis popisa s dodatnim podacima o procesima.

**Primjeri:**

$ ps

PID TTY TIME CMD

13092 pts/1 00:00:00 bash

13120 pts/1 00:00:00 ps

$ ps -f

UID PID PPID C STIME TTY TIME CMD

tecaj01 13092 13091 0 09:20 pts/1 00:00:00 -bash

tecaj01 13121 13092 0 09:20 pts/1 00:00:00 ps -f

Naredba **top** na ekranu kontinuirano ispisuje popis procesa poredanih prema količini procesorskog vremena koji troše.

**Sintaksa:**

top

Na mnogim je sustavima onemogućeno da korisnici bez administratorskih korisničkih prava imaju uvid u popis svih procesa na sustavu. U tom će slučaju naredbe ps (tj. ps -e) i top ispisivati samo popise procesa koji pripadaju korisniku koji je naredbu pokrenuo.

## 7.3. Izvršavanje naredbi u pozadini

Do sada smo naredbe zadavali tako da smo nakon njenog pokretanja morali čekati da sustav završi s njenim izvođenjem da bismo mogli zadati sljedeću. Takav način rada se zove **izvršavanje naredbi u prednjem planu**.

Pri tome je pojavljivanje prompta služilo, između ostaloga, kao znak je li sustav završio s izvršavanjem prethodne naredbe.

Ljuska omogućava da izbjegnemo ograničenje istovremenog pokretanja samo jedne naredbe. Postupak koji možemo u tu svrhu koristiti je **izvršavanje naredbi u pozadini**.

Izvršavanje naredbi u pozadini zadaje se znakom **&** na kraju linije s naredbom.

**Sintaksa:**

NAREDBA &

Nakon pokretanja naredbe s izvršavanjem u pozadini, ljuska će ispisati njezin PID kao znak da će se izvođenje nastaviti u pozadini, a odmah nakon toga pojavit će se prompt i moći ćemo nastaviti s radom.

Po završetku izvođenja naredbe u pozadini, ispisat će se odgovarajuća poruka.

**Primjer:**

$ ls -lR > /tmp/popis & ← *pokrećemo naredbu u pozadini*

[1] 16423 ← *PID je 16423, a redni broj koji*

*je ljuska dodijelila procesu je 1*

$ ← *pojavio se prompt i možemo*

*nastaviti s radom*

...

[1]+ Done ls -lR >/tmp/popis

↑

*na kraju dobivamo poruku da je izvršavanje naredbe*

*u pozadini završilo*

## 7.4. Upravljanje procesima u ljusci

|  |
| --- |
| **fg** - prebacivanje procesa u prednji plan  **bg** - nastavak izvršavanja zaustavljenog procesa u pozadini  **[Ctrl-Z]** - zaustavljanje procesa koji se izvršava u prednjem planu  **jobs** - popis procesa koje se izvršavaju u pozadini |

Izvršavanje naredbi koje smo pokrenuli u prednjem planu, možemo privremeno zaustaviti utipkavanjem kontrolnog znaka **[Ctrl-Z]**. Ljuska će ispisati odgovarajuću poruku i ispisati prompt. Nakon toga možemo zadavati naredbe ljusci. Zaustavljeni program ostaje u memoriji računala i čeka da zadamo neku od naredbi za nastavak njegovog izvođenja.

**Napomena**

**U većini slučajeva izvođenje naredbe u prednjem planu možemo prekinuti utipkavanjem kontrolnog znaka [Ctrl-C].**

Izvođenje naredbi koje su na takav način zaustavljene možemo:

* nastaviti izvoditi u pozadini (naredba **bg**)

**Napomena**

Važno je razlikovati prekidanje od zaustavljanja izvođenja.

Prekidanje izvođenja znači da je izvođenje naredbe prestalo, da je naredba obrisana iz memorije računala i da se njeno izvođenje više ne može nastaviti.

Prekid izvođenja je konačan, dok je zaustavljanje izvođenja privremeno.

* nastaviti izvoditi u prednjem planu (naredba **fg**).

Naredba fg nam omogućava prenošenje u prednji plan ne samo procesa koji su zaustavljeni sa **[Ctrl-Z]**, nego i svih koji se izvršavaju u pozadini.

Naredba **jobs** ispisuje popis svih procesa koji se izvršavaju u pozadini.

**Primjer:**

$ ls -lR > popis ← naredbu *pokrećemo u prednjem planu*

...

[Ctrl-Z] ← *zaustavljamo njeno izvođenje sa* [Ctrl-Z]

[1]+ Stopped ls -lR > popis

↑

*ispisuje se poruka, prvi podatak u poruci je redni*

*broj koji je procesu dodijelila ljuska*

$ ← *pojavio se prompt i možemo*

*nastaviti s radom*

...

$ bg ← *zadajemo nastavak izvođenja naredbe*

*u pozadini*

[1]+ ls -lR > popis &

↑

*ljuska nas obavještava da je izvršavanje naredbe*

*nastavljeno u pozadini*

## 7.5. Signali i naredba kill

*Linux* omogućava slanje signala procesima naredbom **kill**.

Procesima se mogu slati različiti signali, različitih značenja. Neki od signala su:

* **SIGHUP** (1) - zahtjev za re-inicijalizacijom procesa
* **SIGKILL** (9) - zahtjev za grubim prekidom izvršavanja
* **SIGTERM** (15) - zahtjev za prekidom izvršavanja (softverski završetak rada)

**Sintaksa:**

kill [-sSIGNAL] PROCES

Naredbi kill se kao argument treba zadati PID procesa čije izvođenje želimo prekinuti ili redni broj koji je procesu dodijelila ljuska. Ako se navodi redni broj koji je procesu dodijelila ljuska tada ispred njega treba upisati znak **%** (postotak).

**Napomena**

Programi mogu biti napisani tako da ignoriraju primljene signale (odnosno tako da sami određuju svoje ponašanje u slučaju da prime signal).

Izuzetak je signal SIGKILL (9). Njega programi ne mogu ignorirati.

Ako se ne navede signal koji šaljemo, podrazumijeva se da se radi o signalu SIGTERM. Signale možemo zadati njihovim rednim brojem ili simboličkim imenom.

**Primjeri:**

$ jobs

[1]- Running program1 &

[2]+ Running vjezba8 &

↑

*u ovom se primjeru u pozadini izvršavaju dva procesa*

$ ps

PID TTY TIME CMD

15028 pts/0 00:00:00 bash

29847 pts/0 00:00:00 program1 ← *njihove PID-ove*

29281 pts/0 00:00:00 vjezba8 *možemo doznati*

24783 pts/0 00:00:00 ps *naredbom* ps

$ kill -9 29847 ← program1 *prekidamo navodeći*

*njegov PID*

[1]+ Killed program1

$ kill -9 %2 ← *program* vjezba2 *prekidamo*

*navodeći redni broj dodijeljen*

*od ljuske*

[2]+ Killed vjezba8

$

# 8. Arhiviranje i sažimanje datoteka

## 8.1. Arhiviranje datoteka: tar

Naredba **tar** omogućava izradu, pregledavanje i raspakiravanje arhiva. Arhive su datoteke koje u sebi sadrže zapakirane druge datoteke.

**Sintaksa:**

tar c[v]f ARHIVA POPIS

tar t[v]f ARHIVA

tar x[v]f ARHIVA [POPIS]

Ponašanje naredbe tar ovisi o ključu – prvoj zadanoj opciji, koja može biti:

**c** - izrada arhive

**t** - pregledavanje sadržaja arhive

**x** - raspakiravanje cijele ili dijela arhive

Opcijom **f** se označava da se arhiviranje vrši u datoteku i da je sljedeći argument ime datoteke u koju se arhivira. (Ako se ta opcija ne navede, arhiviranje se radi na standardni izlaz.)

Ako se navede opcija **v**, tada naredba tar tijekom rada ispisuje dodatne podatke (npr. popis datoteka koje se arhiviraju ili raspakiravaju).

Obično se arhivama načinjenim naredbom tar daju imena s nastavkom .tar (npr. moji\_podaci.tar).

**Primjeri:(tt direktorij se nalazi unutar /home/user/ direktorija)**

$ ls -RF ← *naredbom* ls -RF *ispisujemo sadržaj*.: *radnog direktorija:*

tttt

d1

f1

f2

d1/ f1

./d1:

f2

$ tar cvf arhiva.tar tttt/ ← *pokrećemo izradu arhive*

tt *u datoteku* arhiva.tar

tt/d1/

tt/d1/f2 ← *zadana je opcija* c*, pa se ispisuje popis*

tt/f1 *arhiviranih datoteka i direktorija*

$ tar tf arhiva.tar ← *provjeravamo što se*

tt *nalazi u arhivi*

tt/d1/

tt/d1/f2

tt/f1

$ mkdir /home/user/vjezba ← *stvaramo direktorij u*

*koji ćemo raspakirati dio arhive i*

*postavljamo*

$ cd /home/user/vjezba ← *taj direktorij za radni*

*direktorij*

$ ls -l

total 0 ← *novi direktorij je prazan*

$ tar xvf /home/user/arhiva.tar –C /home/user/vjezba

./d1/ ↑

./d1/f2 *iz arhive raspakiravamo u direktorij /home/user/vjezba sadržaj tar datoteke*

$ ls -FR ← *provjeravamo što je raspakirano*

tt *iz arhive:*

tt

d1

f1

f2

tt/d1/

tt/d1/f2

tt/f1