



---

# opensource / Linux

epizoda treća

---

Igor Jagec  
Dalibor Karlović

**Linux komandna linija**  
**Pregled najčešćih naredbi**  
**Usporedba naredbi MS DOSa i Linuxa**

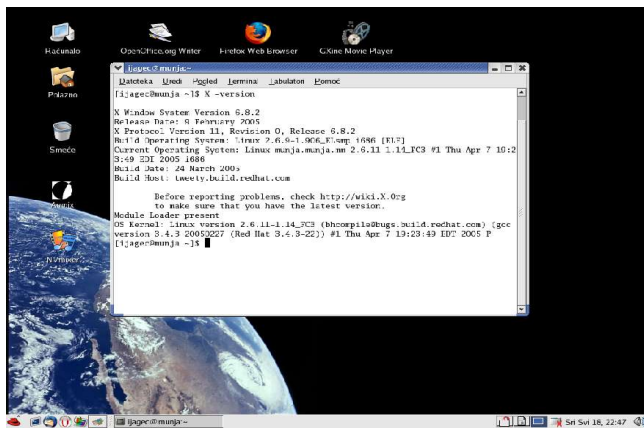


# Linux komandna linija

## Uvod

Komandna linija, tekstualna konzola, ili samo konzola - sve su to samo sinonimi za naredbeni redak odnosno tekstualno radno okruženje, kako su ti pojmovi prevedeni na hrvatski jezik. Zašto u današnje vrijeme koristimo naredbeni redak pokraj veoma naprednih grafičkih alata? Što je u stvari naredbeni redak?

Najjednostavnije rečeno, to je tekstualno sučelje tj. jedan od dvaju koncepata upravljanja operativnim sustavom i/ili pripadajućim mu aplikacijama. Najčešća zabluda među korisnicima je u tome što mnogi smatraju da je noviji koncept - grafičko radno okruženje<sup>1</sup> u stvari zamjena za tekstualno radno okruženje. Naime, to je u potpunosti netočno. Svaki od ovih dvaju koncepata ima svoju primjenu, ovisno o prirodi posla. Za razliku od operacijskog sustava MS Windows (koji je gotovo u potpunosti orijentiran ka grafičkom okruženju), Linux kao jedan od Unix klonova to ipak nije. U prilog tome svakako govori činjenica da je kod Unix klonova (samim time i Linuxa) grafičko sučelje izdvojeno iz jezgre<sup>2</sup> OSa što kod MS Windows nije slučaj.



**Grafika 1:** Emulator konzole

neki od komandno-linijskih alata te je u mnogim slučajevima mnogo lakše i brže

Svaki ozbiljniji Linux korisnik mora svladati osnove naredbenog retka. Iako postoje brojni sofisticirani i zreli grafički alati, neki važniji alati za administraciju sustava ne postoje u grafičkoj inačici. Također je važno naglasiti da su ti grafički alati uglavnom nadogradnja na

<sup>1</sup> GUI (eng. Graphical User Interface), grafičko korisničko sučelje

<sup>2</sup> Kernel, vidite **Epizodu prvu**.

napraviti posao kroz naredbeni redak. U mnogim slučajevima grafički alati nam ne pružaju toliku fleksibilnost i preglednost prilikom korištenja sustava, što može biti problem ukoliko se pojavi greška u programu. U tom slučaju problem se rješava isključivo upotrebom naredbenog retka.

Linuxovo grafičko okruženje je zadnjih godina veoma napredovalo. Možete raditi i u grafičkom okruženju, a da naredbeni redak otvarate samo povremeno. To je moguće na dva načina: pokretanjem emulatora terminala unutar grafičkog sučelja (slično DOS promptu pod Windowsima) te otvaranjem tekstualnog okruženja izvan grafičkog okruženja kombinacijom tipke ALT i neke od funkcijskih tipaka (F1 do F6). Ovako dolazimo do “prave” konzole.

## Ljuske

Kada otvorimo novu konzolu, ono što nam se prikaže unutra je ljuska<sup>3</sup> sustava. Osnovna funkcionalnost ljuske je da nam omogući da pokrećemo naredbe i vidimo što će one uraditi. Za usporedbu, ljuska je nešto vrlo slično MS DOS sustavu, samo mnogo sofisticiranije.

Prva ljuska se zvala jednostavno **sh** i bila je razmjerno jednostavna za korištenje, iako ne prebogata opcijama. Na osnovama otvorenog kôda, iz sh ljuske nastale su mnoge druge: bash, csh, tcsh, zsh i mnoge druge. Ono što će korisnika na prvi pogled zbuniti je činjenica da ljuske na prvi pogled isto izgledaju: imate polje u koje možete ukucavati naredbe, one će nešto raditi, ispisivati i sl., a vi ćete time obavljati posao. Razlika je u tome da svaka od ljusaka nudi različite opcije koje vam olakšavaju posao, možete ih podesiti na način na koji vama više odgovara od neke druge, skriptni jezik koji se koristi unutar ljuske vam više odgovara ili sl. Najpoznatija ljuska koja se koristi na Linux sustavima je **bash**<sup>4</sup> te će se primjeri odavde bazirati upravo na ovoj ljusci.

U tekstu će se prikazivati primjeri korištenja nekih naredbi te će biti posebno označeni:

---

<sup>3</sup> Shell (eng.), ljuska. Detaljniji opis rada ljuske u **Epizodi četvrtoj**.

<sup>4</sup> BASH (eng. Bourne Again Shell), ponovno rođena ljuska

```
[dado@amelie dado]$ whoami  
dado
```

U ovom primjeru u prvoj liniji vidimo naredbeni redak<sup>5</sup> (završava s znakom `$`), naredbu (označena je **podebljano**) dok u drugoj liniji vidimo izlaz koji nam naredba daje. Ovom konvencijom će biti označeni svi primjeri u tekstu koji slijedi.

Veliki broj mogućnosti koje bash (kao i druge ljuske) ima teško je opisati u ovom nevelikom dokumentu pa ćemo opisati samo najvažnije stvari za korištenje.

## SamoDovršenje

Nespretni izraz iz naslova<sup>6</sup> označava mogućnost ljuske da pokuša predvidjeti što želite napisati. Koristi se tako da se pomoću tipke **TAB** dobiju sve mogućnosti koje postoje, opcije se sužavaju dodavanjem dodatnih znakova. U primjeru se umjesto `<TAB>` treba pritisnuti tipka **TAB**:

```
[dado@amelie dado]$ ls /u<TAB>sh<TAB>do<TAB>g<TAB>i<TAB>  
[dado@amelie dado]$ ls /usr/share/doc/g  
gettext-0.14.1    gimpprint-4.2.7    gnuplot-4.0.0  
ghostscript-7.07  glibc-devel-2.3.3  gprolog-1.2.16  
[dado@amelie dado]$ ls /usr/share/doc/gimpprint-4.2.7/
```

Nakon upisivanja slova **g** u primjeru i pritiska na TAB, sustav nam je ponudio šest direktorija koji počinju sa tim slovom, tek smo dodavanjem slova **i** dovoljno suzili izbor da bi nam do kraja nadopunio ime direktorija. U prvoj liniji imamo prikazane tipke koje su bile dovoljne da bi “napisali” ono iz zadnje linije. Ušteda: 16 naspram 34 znaka, više nego dvostruko. Uz naviku na ovu funkciju možete “letjeti” konzolom. Ista stvar vrijedi i za samo ime naredbe:

---

<sup>5</sup> Command prompt (eng.) naredbeni redak, moguće ga je podesiti po želji

<sup>6</sup> Autocompletion (eng.), sposobnost automatskog nadopunjavanja

```
[dado@amelie dado]$ ho<TAB>
hostid      hostname
[dado@amelie dado]$ hostn<TAB>
[dado@amelie dado]$ hostname
amelie.vz
```

## Rad s više programa

Konzola nam, kako smo već rekli, omogućava da pokrećemo i koristimo programe. No, problem nastaje kada želimo koristiti više programa istovremeno. U MS DOSu to nije bilo moguće i upravo to (a ne sama činjenica da mu je sučelje grafičko) je bio temelj za popularnost MS Windows OSa. U isto vrijeme na nekoj drugoj valnoj dužini, Unix je sasvim jednostavno i elegantno u jednoj ljsuci omogućavao rad s više programa koristeći princip **FG-BG**<sup>7</sup>.

Ova metoda se zasniva na činjenici da u jednom trenutku možete koristiti samo jedan program, njega stavljamo u prednji plan (FG), dok su svi drugi programi postavljeni u pozadinu (BG). Kada poželimo koristiti neki od drugih programa, jednostavno onaj koji je u FG prebacimo u BG, a ovaj drugi pozovemo. Ekvivalent bi bila predodžba prozora na grafičkim OSima: prozor koji nam je u prvom planu je u FG, a svi iza njega su u BG.

Prebacivanje nekog programa u BG možemo izvesti na nekoliko načina. Prvi je da na kraj naredbe kojem program pozivamo (njegovo ime) dodamo **&**, a drugi je da za vrijeme dok nam je program u FG pritisnemo kombinaciju tipaka **Ctrl + Z**. Naredbom **jobs** vidimo popis programa u BG i njihov redni broj unutar BG.

Program u FG možemo pozvati samo kad tu nemamo niti jedan (očito, zato da možemo upisati potrebnu naredbu). To se radi s naredbom **fg** iza koje slijedi broj programa unutar BG ili s **%X** gdje je X taj isti broj.

---

<sup>7</sup> FG-BG (eng. Foreground, Background), prednji plan, pozadina.

```
[dado@amelie dado]$ top &
[1] 4698
[dado@amelie dado]$ jobs
[1]+  Stopped                  top
[dado@amelie dado]$ %1
```

Prvom naredbom pokrećemo naredbu **top** i odmah je stavljamo u BG, drugom gledamo koji su nam programi u BG, a trećom postavljamo top u FG te se on dalje normalno nastavlja izvršavati. U drugoj liniji nam prvi broj (u ovom slučaju 1) govori koji broj trebamo napisati iza % ili fg.<sup>8</sup>

## Preusmjeravanje

U **Epizodi prvoj** opisali smo da je osnova funkcioniranja Unix sustava tzv. “kutija s alatom” koncept. Ovaj koncept se zasniva upravo na preusmjeravanju. Konkretno, ukoliko preusmjerimo prvi program u drugi, tada će ono što prvi program ispiše postati ulazni podaci za drugi program. Ovo se može izvesti za proizvoljan broj programa (ograničeni smo, ali toliko visoko i komotno da nećemo niti spominjati kako). Ovako to radi:

```
[dado@amelie dado]$ grep i test
ti
mi
vi
oni
[dado@amelie dado]$ grep i test | wc -l
4
```

Datoteka **test** (koja sadrži sve osobne zamjenice, po jednu u svakom retku) nam je pokusni kunić. Ovdje naredbom **grep** iz te datoteke ispisujemo sve linije koje sadrže slovo **i**. Nakon toga u drugoj naredbi preusmjeravamo izlaz iz naredbe **grep** u naredbu **wc** i ispisujemo koliko redaka je prva naredba ispisala. Preusmjeravanje se vrši znakom **|** (poznatiji kao “pipe”, cijev) iz programa u

---

<sup>8</sup> Drugi broj će biti objašnjen u **Epizodi četvrtoj**.

program. Također možemo preusmjeravati iz ili u datoteku, npr.:

```
[dado@amelie dado]$ wc -l < test
8
[dado@amelie dado]$ wc -l < test > broj-linija-u-test
```

Prvom naredbom kažemo naredbi `wc` da nam kaže koliko linija sadrži datoteka `test`, a u drugoj taj podatak pospremamo u datoteku ***broj-linija-u-test***.

## *Dome, slatki dome*

Linux je višekorisnički OS što znači da istovremeno na njemu može raditi jedan ili više korisnika. Da bi svaki od ovih korisnika imao svoje mjesto pod suncem, dodjeljuje mu se tzv. kućni ili ***matični direktorij***<sup>9</sup>. Ovdje korisnik ima mogućnost pospremanja svojih datoteka, može ih zaštititi od čitanja drugih korisnika itd.

Kućni direktorij se najčešće kreira u `/home` direktoriju na koji se dodaje korisničko ime korisnika kojem dajemo kućni direktorij. Npr. za korisničko ime ***dado*** matični direktorij će najčešće biti ***/home/dado***<sup>10</sup>.

## Pregled najčešćih naredbi

Naredbi koje se koriste u konzoli ima uistinu mnogo, ali one koje ćemo najčešće koristiti ima razmjerno malo pa ćemo ih ovdje navesti s primjerima korištenja.

## *Konvencije*

Naredbe koje ćemo spominjati primaju parametre koji se dodaju iza poziva naredbe. Razlikujemo podatke koje šaljemo naredbi kao i “preklopnike”<sup>11</sup>. Preklopnici ispred sebe imaju znak `-` (ukoliko preklopnik ima jedan znak) ili `--` (ukoliko preklopnik ima više znakova). Npr. naredba

<sup>9</sup> Home directory (eng.) kućni direktorij.

<sup>10</sup> Detalji o datotečnom sustavu, korisnicima i sličnim divotama u ***Epizodi četvrtoj***.

<sup>11</sup> Switch (eng.), preklopnik, u ovom kontekstu je to mogućnost davanja do znanja programu što su podaci koje mu dajemo iza ili slično.



```
[dado@amelie dado]$ ls -a
```

će nam ispisati sve u trenutnom direktoriju (zadano preklopnikom **a**). Preklopnici se mogu kombinirati pa tako

```
[dado@amelie dado]$ ls -a -l /home
```

možemo napisati kao

```
[dado@amelie dado]$ ls -al /home
```

što će biti shvaćeno kao dva preklopnika zbog samo jednog -, a ne dva. Ime direktorija /home iza je podatak o tome koji direktorij želimo izlistati.

## Samopomoć

Većina naredbi podržava preklopnik **-help** koji će dati osnovne informacije kako koristiti naredbu, izlaz možete preusmjeriti u naredbu **less** kako mi mogli jednostavno čitati, krećete se strelicama na tipkovnici.

```
[dado@amelie dado]$ ls -help | less
```

man, info

Mnogo opsežnija pomoć o upotrebi najčešće dolazi upotrebom priručnika, naredbom **man**<sup>12</sup> iza koje slijedi ime naredbe čiji priručnik želimo. Za istu primjenu, uz svježiju dokumentaciju i naprednije čitanje služi naredba **info**.

```
[dado@amelie dado]$ man man
```

## Rad s datotekama

pwd

Kada počnete pretraživati direktorije, lako zaboravite svoj tekući direktorij. Na mnogim distribucijama okruženje bash standardno prikazuje samo njegovo ime, ali ne i putanju do njega. Gdje se trenutno nalazite, saznajete naredbom **pwd**<sup>13</sup>.

---

<sup>12</sup> MAN (eng. MANual), priručnik

<sup>13</sup> PWD (eng. Print Working Directory), ispiši trenutni direktorij.

```
[ijagec@munja ~]$ pwd  
/home/ijagec
```

Iz rezultata se vidi da se nalazite u matičnom direktoriju korisnika *ijagec*.

cd

Kada želite promijeniti direktorij, upotrijebite naredbu **cd**<sup>14</sup>. Ako je napišete bez argumenata, uvijek ćete se naći u matičnom direktoriju; da biste prešli u neki drugi direktorij, neophodno je da upišete i odgovarajuću putanju. Putanja može biti relativna ili apsolutna.

Apsolutne putanje počinju od vrha hijerarhije, znakom / (koji označava korijenski direktorij), a zatim idu niz stablo sustava datoteka sve do traženog direktorija. Uz korištenje apsolutne putanje ćete uvijek završiti u istom direktoriju bez obzira koji vam je trenutni direktorij.

Relativne putanje niz stablo prolaze počevši od tekućeg direktorija tako da se promjena "izračuna" počevši od direktorija gdje se trenutno nalazite.

Uzmimo kao primjer ovo stablo direktorija:

```
/home  
  /ijagec  
    /dokumenti  
      /openoffice  
        /slike
```

Pretpostavimo da se nalazimo u matičnom direktoriju /home/ijagec. Da bi prešli u direktorij **openoffice**, možemo napisati

```
[ijagec@munja ~]$ cd dokumenti/openoffice
```

Da bi prešli u direktorij **slike**, možete napisati relativno (.. kaže da se vratimo jedan korak iznad u hijerarhiji, kao u MS DOSu):

---

<sup>14</sup> CD (eng. Change Directory), promijeni direktorij.

```
[ijagec@munja ~]$ cd ../slike
```

ili apsolutno

```
[ijagec@munja ~]$ cd /home/ijagec/slike
```

To je bio primjer apsolutne putanje; naredba kaže Linuxu da od vrha hijerarhije (/) potraži home, tada u njemu ijagec itd. Putanja je apsolutna ako počinje sa znakom /. U svim drugim slučajevima ona je relativna. Na primjer, ukoliko želite ući u direktorij X11 koji se nalazi u direktoriju /etc koji se nalazi u korijenskom datotečnom sustavu, tada ćete u naredbeni redak upisati sljedeće:

```
[ijagec@munja ~]$ cd /etc/X11
```

i tada ćete ući u direktorij X11 u kojemu se nalaze konfiguracijske datoteke sustava X Windows<sup>15</sup>.

### Naredba

### Što radi

cd	Vraća vas u matični direktorij
cd ~	Također vas vraća u matični direktorij
cd /	Prebacuje vas u glavni direktorij sustava
cd /root	Prebacuje vas u matični direktorij administratora sustava. Samo administrator može pristupiti ovom direktoriju.
cd /home	Prebacuje vas u direktorij /home u kojem se obično nalaze korisnički matični direktoriji.
cd ..	Prebacuje vas jednu stepenicu naviše u roditeljski direktorij.
cd /etc/X11	Bez obzira na to gdje se nalazite, ova apsolutna putanja će vas odvesti u X11, poddirektorij direktorija etc.

**Tabela 1:** primjeri CD naredbe

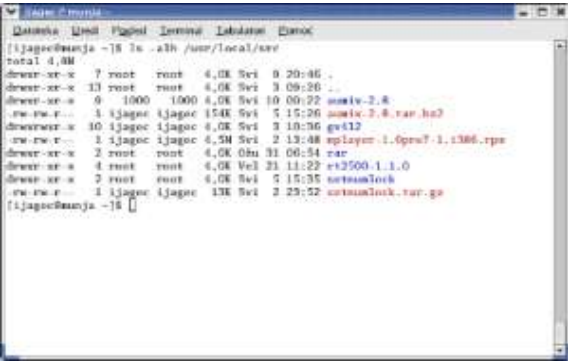
ls

Budući da smo naučili kako prelaziti iz direktorija u drugi direktorij, sada je potrebno naučiti i pregledavati sadržaj tih direktorija. Sadržaj tekućeg direktorija

---

<sup>15</sup> Objašnjeno u *Epizodi drugoj*.

se pregledava naredbom **ls**<sup>16</sup>.



**Grafika 2:** ispis LS naredbe

nam omogućuje da vidimo karakteristike datoteka. Bez ikakvih parametara će samo ispisati sadržaj trenutnog direktorija, ali se prava situacija vidi tek korištenjem nekog od njih, parametri se mogu kombinirati.

Prikazivanje datoteka uz komandu **ls -a** daje poprilično puno detalja, ali će ih uz opciju **-l** biti još više, dobiti ćete podatke o veličini datoteke ili direktorija, datumu njihovog nastanka, vlasniku, ovlastima itd. Kada koristite ovu naredbu, ne morate nužno biti u direktoriju kojeg pregledavate. Na primjer, da biste vidjeli sadržaj direktorija **/etc**, dok se u matičnom direktoriju, upišite:

```
[ijagec@munja ~]$ ls -al /etc
```

**Opcija**

**Što radi**

<b>Ls -a</b>	Izlistava datoteke u direktoriju, uključujući skrivene datoteke.
<b>Ls -l</b>	Izlistava detalje o sadržaju, uključujući ovlaštenja, vlasnika, grupu, veličinu, datum nastanka, da li je datoteka veza ka nečem drugom u sustavu i ako jest, što je odredište veze.
<b>Ls -h</b>	Prilagođava redove veličina u ispisu nama čitljivijem obliku.
<b>Ls -r</b>	Izlistava sadržaj direktorija od kraja k početku.
<b>Ls -R</b>	Ova opcija rekurzivno izlistava sve poddirektorije tekućeg direktorija.

<sup>16</sup> LS (eng. LiSt), “izlistaj”.  
<sup>17</sup> Dot-files (eng.) datoteke s točkom.

**Tabela 2:** LS parametri

Svako zadavanje putanje (ne samo kod ove naredbe) dopušta upotrebu takozvanih zamjenskih znakova. Unesite dio imena kojeg se sjećate, a ostatak zamijenite zamjenskim znakom<sup>18</sup>. To su simboli kojima možete zamijeniti slova, brojeve i druge znakove i tako lakše nađete datoteku, umjesto da čitate dugačke liste direktorija da biste prepoznali što tražite. Naprimjer, znamo da datoteka počinje sa riječju mplay, stoga upisujemo sljedeće:

```
[dado@amelie dado]$ ls mplay*  
mplayer-1.0pre7-1.i386.rpm
```

Znak \* (zvjezdica) zamjenjuje bilo koliko bilo kojih znakova, dok ? (upitnik) zamjenjuje točno jedan bilo koji znak.

**mkdir, rmdir**

Kreiranje direktorija obavlja se naredbom **mkdir**<sup>19</sup>. Pretpostavlja se da u direktoriju u kojem želite kreirati direktorij imate prava pisanja. Za uklanjanje direktorija naredbom **rmdir** je dovoljno da imate prava da ga uklonite kao i da bude prazan. Obje naredbe primaju niz znakova te prva kreira, a druga uklanja direktorij u tekućem direktoriju ako takav postoji. Također se može koristiti i apsolutna sintaksa:

<sup>18</sup> Wildcard (eng.), “divlja karta”, u ovom kontekstu znak koji može zamijeniti bilo koji drugi.

<sup>19</sup> MKDIR (eng. MaKe DIRectory), kreiraj direktorij.

```
[ijagec@munja ~]$ mkdir proba
[ijagec@munja ~]$ ls
proba
[ijagec@munja ~]$ rmdir proba
```

## rm

Naredbom **rm**<sup>20</sup> brišemo direktorij ili datoteku ako imamo prava na to. Najčešća joj je primjena brisanje nepraznih direktorija koje bi bila prava muka isprazniti jedan po jedan uz parametre **-rf**.

Tako će

```
[ijagec@munja ~]$ rm -rf dokumenti
```

nepovratno obrisati direktorij **dokumenti** i sve što se u njemu nalazi.

## mv, cp

Ukoliko nešto želimo pomaknuti, koristit ćemo naredbu **mv**<sup>21</sup>, dok ćemo za kopiranje koristiti naredbu **cp**<sup>22</sup>. Obje naredbe primaju minimalno dva parametra pri čemu je prvi izvorišna datoteka ili direktorij, a drugi odredišna datoteka ili direktorij. Ukoliko se zadaje više od dva parametra, tada zadnji nužno mora biti direktorij u koji će se prethodni pomaknuti tj. kopirati.

```
[ijagec@munja ~]$ mv -R proba test slike
[ijagec@munja ~]$ cp slike/test .
```

Prva naredba će rekurzivno pomaknuti direktorij **proba** i datoteku **test** u direktorij **slike**, dok će druga kopirati datoteku test nazad u trenutni direktorij.

Zanimljivo je spomenuti da se naredba mv koristi i za preimenovanje datoteka ili direktorija. Ukoliko je ishodišni i odredišni direktorij datoteke koju “mičemo” isti, tada se datoteci mijenja ime:

---

<sup>20</sup> RM (eng. ReMove), obriši.

<sup>21</sup> MV (eng. MoVe), pomakni.

<sup>22</sup> CP (eng. CoPy), kopiraj.

```
[ijagec@munja ~]$ mv test novi-test
```

## find, slocate

Doći ćete u situaciju kada znate da neka datoteka ili direktorij postoje, ali ne znate gdje se nalazi. Traženje datoteka i direktorija lakše je uz naredbu **slocate**<sup>23</sup> ili naredbu **find**. S pomoću naredbe slocate možete prikazati svaku datoteku ili direktorij čije ime sadrži zadanu riječ ili dio riječi. Na primjer, ako želite pronaći sve datoteke koje u nazivu sadrže riječ **numlock**, tada će vam odziv biti otprilike ovakav:

```
[ijagec@munja ~]$ slocate numlock
/home/ijagec/dokumenti/html/linux/ubuntu/ubuntu5.04/sample/Default_numlockx
/home/backup/Ubuntu/ijagec/ubuntu/ubuntu5.04/sample/Default_numlockx
/usr/local/src/setnumlock.tar.gz
/usr/local/src/setnumlock
/usr/local/src/setnumlock/Makefile
/usr/local/src/setnumlock/COPYING.LIB
/usr/local/src/setnumlock/Numlock.c
/usr/local/src/setnumlock
/usr/bin/setnumlock
```

Razlika između find i slocate je da prva izravno pretražuje disk, dok druga pretražuje indeksnu datoteku u kojoj se nalazi popis datoteka na disku, prva metoda je sporija, ali aktualnija. Ukoliko tražite datoteku koja je razmjerno nedavno došla na disk, koristite find, u suprotnom će dostajati slocate.

Naredba find je izrazito moćna u mogućnosti selekcije traženog pa tako možemo zadati ovakvu potragu:

---

<sup>23</sup> SLocate (eng. Secure Locate), sigurno pretraživanje.

```
[dado@amelie dado]$ find . -name *bz2 -type f -size +25000k -user  
dado  
./kernel-2.6.10+config.tar.bz2
```

Ovaj redak će u trenutnom direktoriju rekurzivno tražiti **datoteku** čiji je vlasnik **dado**, ime joj završava na **bz2** te je **veća od 25MB**. Pronađena je jedna.

## less, more

U Linuxu cijevi povezuju standardni izlaz jedne naredbe sa ulazom druge naredbe. Naredba `ls` ispisuje puno podataka, ali problem se javlja kada se sadržaj direktorija počne tako brzo pomicati da jednostavno ne stignete pratiti što se u njemu nalazi. U tom slučaju jedino rješenje da izlazne podatke naredbe `ls` preusmjerite uslužnom programu **less**<sup>24</sup> ili **more**<sup>25</sup>. To su naredbe za zaustavljanje ispisa tako da ga možete regulirati. Kad se ekran napuni ispisom naredbe, ove naredbe će ga zaustaviti kako biste mogli na miru pročitati što vam prva naredba poručuje. Kad pročitate do kraja, pristiskom na razmaknicu dobit ćete novi ekran pun korisnih i nutricionistički ispravnih informacija.

## Sadržaj datoteke

Ukoliko ste vrijedno pročitali sve do amo i još uvijek se pitate gdje je skrivena ta velika moć kojom se svi Linuxaši hvale, probat ćemo približiti nekim primjerima. Iako su primjeri prejednostavni za dostojno predstavljanje mogućnosti alata, poanta je u masovnosti posla. Tako ćete vi sve navedeno moći napraviti s nekim Windows alatom, ali npr. imati 1000 datoteka i npr. svakoj zamijeniti riječ “august” sa “kolovoz” je nemali posao. Magične riječi: automatizam, skriptabilnost.

## head, tail

Naredba **head**<sup>26</sup> služi za pregledavanje početka datoteke, kao prvi parametar prima putanju do tekstualne datoteke koju želimo pregledavati. Može biti korisna ukoliko znate da se informacija koju tražite nalazi u prvih nekoliko redova datoteke

---

<sup>24</sup> Less (eng.) manje.

<sup>25</sup> More (eng.) više.

<sup>26</sup> Head (eng.) glava.



pa da ne pretražujete cijelu. Podrazumijevano ispisuje prvih 10 redova. Broj redova koje će naredba prikazati možete i zadati, kao u sljedećem primjeru:

```
[dado@amelie dado]$ head -20 xorg.conf
```

Dakle, u ovom slučaju naredba `head` će ispisati samo prvih 20 redova datoteke *xorg.conf*.

Naredba *tail*<sup>27</sup> je suprotna naredbi `head`. S pomoću nje možete pročitati posljednjih nekoliko redova datoteke, princip rada je isti kao i za `head`. Često se koristi kod pregledavanja neke datoteke kojoj se sadržaj stalno dodaje (npr. sistemski izvještaji) uz pomoć parametra *-f*.

```
[dado@amelie dado]$ tail -f /var/log/messages
```

## grep

Alat `grep` služi za pretraživanje niza znakova unutar datoteke ili skupine datoteka. Npr. pronađimo sve linije u datoteci koje sadrže niz znakova *By*.

```
[dado@amelie dado]$ grep By /etc/X11/xorg.conf
# By default, Mandrake 6.0 and later now use a font server
```

Za drugi primjer nađimo sve datoteke unutar `tmp` direktorija i iz njih ispišimo sve linije koje imaju niz znakova *I get* nakon čega može uslijediti bilo što, bitno je da iza toga imaju *any*.

---

<sup>27</sup> Tail (eng.) rep.

```
[dado@amelie dado]$ grep --color -RE "I get.*any" tmp/*
tmp/Linux/ipw2200/t20346.html:  <!--QuoteBegin-durvish+Dec 16
2004, 11:10 PM--><div class='quotetop'>QUOTE(durvish &#064; Dec 16
2004, 11:10 PM)</div><div class='quotemain'><!--QuoteEBegin-->2
things<br /><br />you don't have theESSID name in there<br /
>and try auto for your chanel<br />and boy do you gotalot of stuff
in that file<br /><br />here is the file i put on my girlfriends
laptop for when she is here --- very short and very sweet<br /><br
/>remember when first trying something &quot;KISS&quot; -- Keep It
Simple Stupid -- works for me until I get something working
good<br /><br /><br />DEVICE=wlan0<br />BOOTPROTO=dhcp<br /
>ONBOOT=yes<br />ESSID=stayoffmynet<br />MODE=Managed<br /><br /
>it works great and she has no prob whatsoever connecting<br /
><div align='right'><a
href="index.php?act=findpost&pid=161158"><img
src='http://mandrakeusers.org/style_images/1/post_snapback.gif'
alt='*' border='0' /></a></div><br /><!--QuoteEnd--></div><!--
QuoteEEnd--><br /><br /><br />ifup wlan0 on that config just
results in it trying to find the dhcp server on my network and
stalling.<br /><br />i cant seem to change the IEEE thing in
iwconfig - it says &quot;802.11g&quot; - while my base station is
802.11b - any way to force this to 802.11b?<br />
```

## sed

Alat <sup>28</sup>sed služi za automatsko i brzo editiranje ulaznih podataka. Npr. imamo datoteku u kojoj želimo zamijeniti sva pojavljivanja niza (at) s @.

```
[dado@amelie dado]$ sed "s/(at)/@" < datoteka.txt
```

## awk

Alat **awk** je programski jezik u malom, služi uglavnom za pisanje skripti i najčešće se kombinira sa sedom. Npr, imamo ovakvu datoteku:

```
Igor Jagec igor.jagec@udruga-point.hr
Dalibor Karlovic dado@krizevci.info
Uduga HULK hulk@linux.hr
```

Ispišimo samo "imena" (tj. sve ispred prvog razmaka):

<sup>28</sup> SED (eng. Stream EDitor), uređivač toka.

```
[dado@amelie dado]$ awk -F' ' '{ print $1 }' < primjer.txt
Igor
Dalibor
Uduga
```

Ispišimo sad tu istu datoteku, ali ovaj put samo redni broj linije unutar datoteke zajedno s e-mail adresom sa svim slovima velikim:

```
[dado@amelie dado]$ awk -F' ' '{ print NR".)", toupper($3) }' <
primjer
1.) IGOR.JAGEC@UDRUGA-POINT.HR
2.) DADO@KRIZEVCI.INFO
3.) HULK@LINUX.HR
```

## file

Alat <sup>29</sup>file je jedan od vrlo korisnih alata kakav teško da ćete naći na MS Windows. Namjena mu je ta da uzme neku vama nepoznatu datoteku i kaže vam što to jest kako bi je vi npr. mogli otvoriti ispravnim programom. Primjena je jednostavna, samo se kao prvi parametar navede putanja do datoteke koju želite analizirati:

---

<sup>29</sup> File (eng.), datoteka.

```
[dado@amelie Dokumenti]$ ls
Audio/      Faks/                  POINT/
Simon.jpg*  Tekstovi/
djole.m3u   Lion Female - 1024x768.jpg* radionica_3.sxw* Slike/
[dado@amelie Dokumenti]$ file radionica_3.sxw
radionica_3.sxw: Zip archive data, at least v2.0 to extract
[dado@amelie Dokumenti]$ file Lion\ Female\ -\ 1024x768.jpg
Lion Female - 1024x768.jpg: JPEG image data, JFIF standard 1.01
[dado@amelie Dokumenti]$ file Faks
Faks: directory
```

## Vlasništvo nad datotekama

Kada pokušate prijeći u matični direktorij administratora, dobiti ćete ovakvu poruku:

```
[ijagec@munja ~]$ cd /root
bash: cd: /root: Pristup odbijen
```

To je bio prikaz rada Linuxovog mehanizma sigurnosti. Linux je, kao što rekosmo, višekorisnički sustav i ovlasti za pristupanje datotekama predstavljaju jedan vid zaštite od namjernog ili nenamjernog “brljanja” po sustavu. Jedan način da i u takvim situacijama pristupate direktoriju je prelaskom u administratoski način rada naredbom **su**<sup>30</sup>.

```
[ijagec@munja ~]$ su
Password:
[root@munja ijagec]# cd /root
```

Primjetite da je ijagec u prednjem dijelu naredbenog retka zamijenjeno s **root**, postali smo super-korisnik na sustavu, možemo uraditi sve što želimo.

Preuzimanje prava administratora nije uvijek zgodno niti preporučljivo, zato što su tako veoma lako možete slučajno oštetiti ili izmijeniti važne konfiguracijske datoteke. Sve datoteke imaju svog vlasnika - korisnika koji ih je napravio ili koje

<sup>30</sup> SU (eng. Switch User), promijeni korisnika.

su mu pridjeljene. Ako ste npr. napravili datoteku seminar.txt u matičnom direktoriju, tada ta datoteka pripada vama. Možete odrediti tko je smije čitati, upisivati podatke u nju, ili (ukoliko se radi o aplikaciji) izvršavati je. Čitanje, upisivanje i izvršavanje su tri glavne stavke prilikom određivanja dopuštenja. Budući da se prilikom kreiranja korisničkog računa korisnici svrstavaju u grupe, možete zadati grupe koje su ovlaštene da čitaju datoteku, upisuju podatke u nju ili je izvršavaju te tako još fleksibilnije podesiti dopuštenja.

```
[ijagec@munja ~]$ ls -l seminar.txt
-rw-r--r-- 1 ijagec users 39 2005-05-18 11:19 seminar.txt
```

Dobili smo puno podataka. Možemo vidjeti tko čita (r) datoteku, a tko smije u nju upisivati podatke (w), kao i kojem korisniku (ijagec) i grupi (users) datoteka pripada.

Tri spomenuta skupa ovlaštenja odnose se na: vlasnika datoteke, vlasničku grupu i na ostale korisnike.

-	(rw-)	(rw-)	(r--)	1	ijagec	users
tip	vlasnik	grupa	ostali			

Prvo polje, koje predstavlja tip datoteke, može sadržavati sljedeće:

<b>d</b> – direktorij
- – obična datoteka
<b>l</b> – veza (nešto kao prečica u MS Windows)

Naredna tri skupa od po tri polja, mogu sadržavati sljedeće:

<b>r</b> – datoteka se može čitati (eng. Read)
<b>w</b> – u datoteku se može upisivati (eng. Write)
<b>x</b> – datoteka se može izvršavati (eng. eXecute)
- – dozvola (bilo koja) je uskraćena

Vlasnik datoteke (u ovom slučaju, to je ijagec) ovlašten je da datoteku čita i da u nju upisuje podatke. Korisnici koji su stavljeni u grupu users (a kojoj ijagec

može, ali ne mora pripadati) imaju dozvolu čitanja. Ostali korisnici (ne ijagec i ne članovi grupe users) imaju ovlaštenja za čitanje datoteke. Ovdje se ne radi o programu pa niti vlasnik a niti grupa nisu ovlašteni da datoteku izvršavaju, jer se ona i ne može izvršavati.

## chmod

S pomoću naredbe `chmod`<sup>31</sup> mijenjaju se postojeća dopuštenja. U sljedećem primjeru prikazujemo promjenu ovlaštenja za rad sa datotekom `seminar.txt`. Ovlaštenja za rad sa spomenutom datotekom izgledaju ovako:

```
-rw-r--r-- 1 ijagec users 39 2005-05-18 11:19 seminar.txt
```

Sada ćemo upisati sljedeću naredbu:

```
[ijagec@munja ~]$ chmod a+w seminar.txt
```

i sada će ovlaštenja izgledati ovako:

```
-rw-rw-rw- 1 ijagec users 39 2005-05-18 11:19 seminar.txt
```

Naredba **a+w** dopušta svim korisnicima da mogu upisivati u datoteku `seminar.txt`, i sada svi korisnici mogu upisivati u datoteku `seminar.txt`. Kada želite ukinuti sve dozvole za rad sa datotekom `seminar.txt`, pomoću naredbe `chmod` poništite ovlaštenja za čitanje i upisivanje podataka.

```
[ijagec@munja ~]$ chmod go-rw seminar.txt
```

Kada upišete naredbu `chmod go-rw`, time naređujete sustavu da vlasničkoj grupi i ostalim korisnicima zabrani čitanje i upisivanje u datoteku `seminar.txt`. Ovlaštenja će sada izgledati ovako:

```
-rw----- 1 ijagec users 39 2005-05-18 11:19 seminar.txt
```

Zamišljajte ovaj postupak kao stenografski način mijenjanja ovlaštenja pomoću naredbe `chmod`, budući da je potrebno samo zapamtiti nekoliko simbola i slova da biste je koristili.

Kome dodjeljujemo ili oduzimamo dozvole:

---

<sup>31</sup> CHMOD (eng. CHange MODe), promijeni prava.

**u** - Korisnik (eng. user) kome pripada datoteka (tj. njen vlasnik)  
**g** - Grupa (eng. group) kojoj datoteka pripada  
**o** - Ostali (eng. others) korisnici (niti vlasnik niti grupa)  
**a** - Svi (eng. all), tj. Sva tri prethodno navedena

#### Ovlaštenja:

**r** - Čitanje  
**w** - Pisanje  
**x** - Izvršavanje

#### Akcije:

**+** - Davanje ovlasti  
**-** - Ukidanje ovlasti  
**=** - Isključivo ovlaštenje

Sve što trebate jest kombinirati odgovarajuću kombinaciju znakova. Ukoliko želite izmijeniti ovlasti cijelom direktorijskom stablu, tada je potrebno upotrijebiti i komandu **-R**. Npr. ukoliko želite dozvoliti svim korisnicima čitanje i upisivanje u direktorij **ijagec** (na bilo koje mjesto unutar njega), tada je potrebno upisati sljedeću naredbu:

```
[ijagec@munja ~]$ chmod -R a+rw ijagec
```

Još jedan način za mijenjanje ovlaštenja koji će vam se u početku možda činiti kompliciran. Prikažimo ponovno prvobitna ovlaštenja za rad nad datotekom **seminar.txt**.

```
-rw-r--r-- 1 ijagec users 39 2005-05-18 11:19 seminar.txt
```

Svaka vrsta ovlaštenja se može predstaviti oktalnim brojem:

**r** = 4  
**w** = 2  
**x** = 1  
**-** = 0

Kada ove vrijednosti zbrojite, njihov zbroj će označavati ovlaštenja koja ste zadali. Primjer:

-	(rw-)	(rw-)	(r--)	1	ijagec	users
	4+2+0	4+2+0	4+0+0			

Zbroj za vlasnika je 6, za grupu također 6, a za ostale korisnike je 4. Sva ovlaštenja će biti prikazana nizom oktalnih brojeva **664**. Ova naredba je jedna od najvećih tajni za Windows web-mastere na Unix/Linux serveru pa imaju i naziv **chmodanje** za taj mistični proces.

Ovlaštenja	Što znači
-rw----- (600)	Samo vlasnik ima ovlasti za čitanje i upisivanje podataka
-rw-r--r-- (644)	Samo vlasnik ima ovlaštenje za čitanje i upisivanje podataka; grupa i ostali korisnici imaju samo ovlaštenja za čitanje.
-rwx----- (700)	Samo vlasnik ima ovlaštenje za čitanje i upisivanje podataka, kao i za izvršenje datoteke
-rwxr-xr-x (755)	Vlasnik ima ovlaštenja za čitanje, upisivanje podataka i izvršavanje datoteke; grupa i ostali korisnici imaju samo ovlaštenja za čitanje i izvršavanje.
-rwx--x--x (711)	Vlasnik ima ovlaštenja za čitanje, upisivanje podataka i izvršavanje datoteke; grupa može samo izvršavati datoteku, kao i ostali korisnici.
-rw-rw-rw- (666)	Svatko može čitati datoteku i u nju upisivati podatke.
-rwxrwxrwx (777)	Svatko može čitati datoteku, u nju upisivati podatke i izvršavati ih.
drwx----- (700)	Samo vlasnik može čitati ovaj direktorij i upisivati u njega.
drwxr-xr-x (755)	Svatko može čitati podatke iz direktorija, ali njegov sadržaj može mijenjati samo vlasnik.

**Tabela 3:** tipične CHOWN vrijednosti



## chown

Naredbom `chown`<sup>32</sup> možete mijenjati vlasnika datoteka ili direktorija.

```
-rw-r--r-- 1 ijagec users 39 2005-05-18 11:19 seminar.txt
```

Vlasnik datoteke `seminar.txt` je bio korisnik `ijagec`, a također pripada i grupi `users`. Ukoliko želimo promijeniti vlasnika, tj. dati vlasništvo nad datotekom `seminar.txt` nekom drugom korisniku, to ćemo učiniti s pomoću naredbe `chown`.

Ukoliko nismo super-korisnik, mi moramo biti vlasnik datoteke da bi joj mogli dalje mijenjati vlasništvo, ali ne možemo promijeniti vlasnika datoteke već samo dati nekoj drugoj grupi vlasništvo. Također, uvjet je taj da pripadamo i toj drugoj grupi. Ova ograničenja postoje zbog smanjenja mogućnosti proboja sigurnosnih mjera sustava.

U ovom primjeru ćemo kao super-korisnik mijenjati vlasnika datoteke (nema ograničenja navedenih gore):

```
[root@munja ijagec]# chown ivanka:users seminar.txt
[root@munja ijagec]# ls -l seminar.txt
-rw-r--r-- 1 ivanka users 39 Svi 18 11:19 seminar.txt
```

Dakle, sada vlasništvo nad datotekom `seminar.txt` ima korisnik `ivanka` te grupa `users`. Ukoliko želimo promijeniti vlasništva nad cijelim direktorijom i svim mu pripadajućim datotekama, to ćemo učiniti tako da ćemo naredbi `chown` dodati opciju `-R`.

```
[root@munja ijagec]# chown -R ivanka:users radionica/
```

Dakle, sada je korisnik `ivanka` iz grupe `users` vlasnik direktorija `radionica`.

---

<sup>32</sup> CHOWN (eng. CHange OWNer), promijeni vlasnika.

# Usporedba naredbi MS DOSa i Linuxa

Namjena naredbe	MS-DOS	Linux	Primjer u Linuxu
Kopira datoteke	copy	cp	cp seminar.txt /home/ivanka
Premješta datoteke	move	mv	mv seminar.txt /home/ivanka/dokumenti
Lista datoteke u direktoriju	dir	ls	ls
Briše ekran	cls	clear	clear
Zatvara naredbeni redak	exit	exit	exit
Prikazuje datum	date	date	date
Briše datoteke	del	rm	rm seminar.txt
Prikazuje parametar na ekranu	echo	echo	echo poruka
Otvora jednostavan program za obradu teksta	edit	pico	pico seminar.txt
Usporedi sadržaj dvije datoteke	fc	diff	diff datoteka1 datoteka2
Pronalazi zadani tekst u datoteci	find	grep	grep riječ_ili_fraza seminar.txt
Prikazuje objašnjenje naredbe	naredba /?	man	man chown
Prikazuje datoteku	more	more ili less	less seminar.txt
Mijenja ime datoteke	ren	mv	mv seminar.txt seminar- preimenovano.txt
Prikazuje tekuće mjesto u stablu datoteka	chdir	pwd	pwd
Mijenja radni direktorij prema zadatoj (apsolutnoj) putanji	cd putanja	cd putanja	cd /home/ivanka

**Tabela 4:** Usporedba naredbi MS DOSa i Linuxa