



# SHELL - alati za shell programiranje

Linux ▼

## WC

Brojanje reči karaktera i linija

```
$ wc [-cwl] filename
$ wc -l /etc/protocols
```

## Redirekcija ulaza i izlaza

UNIX komande primaju podatke sa **standardnog ulaza (stdin)**, rezultate izvršenja šalju na **standardni izlaz (stdout)**, a poruke o greškama na **standardni uređaj za greške (stderr)**. Većina UNIX komandi koristi tastaturu kao standardni ulaz, a monitor kao standardni izlaz i uređaj za greške.

**Ulaz komande preusmerava se pomoću znaka <** (manje od) na sledeći način:

```
$ command < inputdevice
```

Primer:

```
$ wc -l < /tmp/jsmith.dat
```

**Za redirekciju izlaza se koristi znak >**. Ukoliko se redirekcija vrši u postojeću datoteku, datoteka se briše, a zatim se kreira nova u koju se smešta rezultat izvršenja komande. Za dodavanje izlaza na postojeću datoteku koristi se **znak »**.

```
$ sort kyuss.txt > /dev/lp0
$ ls -l /home/jsmith > myfile
$ ls -l /tmp/jsmith >> myfile
$ >emptyfile
```

## Povezivanje komandi u pipeline

Pipeline funkcioniše na sledeći način: standardni izlaz komande sa leve strane znaka pipe (|) postaje standardni ulaz komande sa desne strane znaka. Znak pipe zahteva komande i sa leve i sa desne strane, a razmaci između znaka i komande su proizvoljni.

Primer:

```
$ ls -l /etc/ > /tmp/files_in_etc
$ wc -l < /tmp/files_in_etc
145
```

```
$ ls -l /etc | wc -l
145
```

## tee komanda

Komanda tee omogućava da se izlaz komande šalje na više lokacija odjednom. Na primer, ako se želi redirekcija u neki fajl, i istovremeni ispis na ekran, može se uraditi sledeće:

```
ps -ef | tee /tmp/troubleshooting_file
```

Da bi se, umesto da se fajl obriše, tekući izlaz na njega dodao, treba dodati opciju -a:

```
ps -ef | tee -a /tmp/troubleshooting_file
```

Mogu se specificirati i više fajlova kao argumenti **tee** komande, *output* procesa će ići u svaki od tih fajlova.

## head

Prikaz početka fajla

```
$ head -n 3 /var/log/syslog
```

## tail

Prikaz kraja fajla

```
$ tail -n 5 /var/log/syslog
```

Za nadgledanje promena u fajlu koristi se opcija -f.

## join i split

Komanda join dozvoljava spajanje više fajlova po zajedničkom polju.

**Primer:**

```
$ cat file1.txt
1 Pera
2 Mika
3 Laza
$ cat file2.txt
1 Peric
2 Lazic
3 Mikic
$ join file1.txt file2.txt
1 Pera Peric
2 Mika Lazic
3 Laza Mikic
```

U ovom slučaju fajlovi su spojeni preko prvog polja i ona moraju da budu identična. Ako nisu onda mogu da se sortiraju tako da se spajanje, u ovom slučaju, vrši preko vrednosti 1, 2 i 3.

Ako su dati sledeći fajlovi:

```
$ cat file1.txt
Pera 1
Mika 2
Laza 3
$ cat file2.txt
1 Peric
2 Lazic
3 Mikic
```

Da bi izvršili njihovo spajanje potrebno je pozvati sledeću komandu:

```
$ join -1 2 -2 1 file1.txt file2.txt
1 Pera Peric
2 Mika Lazic
3 Laza Mikic
```

-1 se odnosi na file1.txt, a -2 na file2.txt.

Komandom split fajl se deli u odgovarajući broj različitih fajlova.

```
$ split somefile
```

Ovako se u novonastalim fajlovima nalazi po 1000 linija iz originalnog fajla i oni se zovu x\*\*.

## sort komanda

Korisna komanda za sortiranje izlaza neke komande ili fajla u specifikiranom redosledu. Njene opcije su sledeće:

- -d Sorts via dictionary order, ignoring non-alphanumerics or blanks.
- -f Ignores case when sorting.
- -g Sorts by numerical value.
- -M Sorts by month (i.e., January before December).
- -r Provides the results in reverse order.
- -m Merges sorted files.
- -u Sorts, considering unique values only.

#### Primer:

- Kreirati fajl /tmp/outoforder sa sledećim sadržajem:

```
Zebra
Quebec
hosts
Alpha
Romeo
juliet
unix
XRay
xray
Sierra
Charlie
horse
horse
horse
Bravo
1
11
2
23
```

- Sortirati fajl u rečničkom redosledu:

```
sort -d /tmp/outoforder
```

Rezultat je:

```
1
11
2
23
Alpha
Bravo
Charlie
Quebec
Romeo
Sierra
XRay
Zebra
horse
horse
horse
hosts
juliet
unix
xtra
```

- Treba primetiti da se string horse pojavljuje više puta. Da bi se uklonile te dodatne pojave, koristiti opciju -u:

```
sort -du /tmp/outoforder
```

## Korišćenje kontrolinih karaktera

Kontrolni karakteri se zadaju: + karakter ( se na ekranu prikazuje kao simbol ^ (carat)). Kontrolni karakteri Bourne-again shella koji se najčešće koriste su:

- **<Ctrl-c>** prekida izvršenje procesa koji radi u prvom planu;
- **<Ctrl-d>** označava kraj fajla; napuštanje programa koji podatke čitaju sa standardnog ulaza
- **<Ctrl-u>** briše celu komandnu liniju;
- **<Ctrl-w>** briše zadnju reč u komandnoj liniji;
- **<Ctrl-s>** privremeno zaustavlja izvršenje procesa u prvom planu. Može se koristiti prilikom pregledanja sadržine nekog velikog direktorijuma komandom ls ili ukoliko se neka datoteka prikazuje na ekranu programom cat;
- **<Ctrl-g>** nastavlja se izvršenje procesa u prvom planu.

**Primer:** bc (basic calculator)

```
$ bc      # pokreće bc
100/5    # inicira operaciju deljenja
20       # program bc prikazuje rezultat prethodne operacije
<Ctrl-d> # napuštanje programa i povratak u shell
$
```

## Promenljive

Na Linux sistemima postoje dva tipa promenljivih:

- **sistemske promenljive**, koje kreira i održava sam operativni sistem. Ne preporučuje se promena njihovog sadržaja (zašto?). Ovaj tip promenljivih definiše se strogo velikim slovima,
- **korisnički definisane promenljive** (User defined variables - UDV), koje kreiraju i održavaju korisnici. Ovaj tip promenljivih se obično definiše malim slovima.

**U shell programiranju promenljive se ne deklarišu za specifični tip podataka** - dovoljno je dodeliti vrednost promenljivoj i ona će biti alocirana prema toj vrednosti. U Bourne Again Shell-u, promenljive mogu sadržavati brojeve, karaktere ili nizove karaktera.

## Važnije sistemske promenljive

Sistemske promenljive mogu se videti pozivom komande **env**:

```
$ env
LC_PAPER=sr_RS
XDG_VTNR=7
MATE_DESKTOP_SESSION_ID=this-is-deprecated
LC_ADDRESS=sr_RS
SSH_AGENT_PID=2681
XDG_SESSION_ID=c2
LC_MONETARY=sr_RS
...
```

Neke od njih su:

- SHELL lokacija komandnog interpretera
- HOME home direktorijum korisnika
- PATH putanja u kojoj se traže izvršne datoteke
- PWD tekući direktorijum

Pojedinačno, sadržaj promenljive može se videti pozivom:

```
$ echo $PATH
/usr/local/bin:/usr/bin:/bin:/usr/bin/X11:/usr/games
```

## Definisanje korisničkih promenljivih

Svaka promenljiva je univerzalna i nema nikakvu deklaraciju tipa (integer, float, string) i definiše se na sledeći način:

```
variablename=value
```

Primer.

```
$ br=10
```

Prilikom definisanja, odnosno dodele vrednosti, potrebno je primeniti sledeće konvencije o imenima promenljivih:

- **Ime promenljive mora početi alfanumeričkim karakterom ili donjom crtom** ‘\_’ (underscore character) praćenim jednim ili više alfanumeričkih karaktera.

Primer. Korektne promenljive su: HOME, SYSTEM\_VERSION, br, \_ime;

- **Prazne karaktere ne treba stavljati ni sa jedne strane znaka jednakosti** prilikom dodele vrednosti promenljivim.

Primer.

```
$ br=10 # neispravno - prazni karakteri
$ br= 10 # neispravno - prazni karakteri
$ br = 10 # neispravno - prazni karakteri
$ br=10 # ispravno
```

case sensitive

```
$ bR=20
$ Br=30
$ echo $bR
20
```

Može se definisati **promenljiva nulte dužine (NULL variable)**, odnosno promenljiva koja nema vrednost u trenutku definisanja. Vrednosti ovih promenljivih se ne mogu prikazati komandom echo, sve dok im se ne dodeli vrednost;

```
$ br=
$ ime=""
```

- **Imena promenljivih ne smeju sadržati specijalne znake** (poput ? i \*).

Primer 1.

```
$ echo $ime # prikazuje vrednost promenljive ime
johnny
$ echo ime # prikazuje string ime
ime
```

Primer 2.

```
$ x=10
$ xn=abc
$ echo $x $abc
10 abc
```

## Navodnici

Bash shell prepoznaje tri tipa navodnika, i to:

- **Dvostruki navodnici** - “Double Quotes”. Sve što se nalazi u ovim navodnicima gubi originalno značenje (osim \ i \$).
- **Jednostruki navodnici** - ‘Single quotes’. Sve što je zatvoreno jednostrukim navodnicima ostaje nepromenjeno.
- **Obrnuti navodnici** - `Back quote`. Izraz zatvoren obrnutim navodnicima tretira se kao komanda koju treba izvršavati.

Primer.

```
$ echo "Danasnji datum : date" # tretira date kao string
Danasnji datum : date
$ echo "Danasnji datum : `date`" # tretira date kao komandu
Danasnji datum : Fri Apr 2 16:30:35 CEST 2004
```

## Regularni izrazi i metakarakteri

Regularni izrazi su sintaksički skup fraza koje reprezentuju šablone unutar teksta ili stringova. Regularni izrazi omogućavaju reprezentaciju različitih nizova karaktera vrlo malim skupom predefinisanih karaktera. Često sadrže i **metakaraktere** - karaktere koji reprezentuju drugu grupu karaktera i komandi.

Fajl koji će se koristiti u svrhu testiranja je recimo **/tmp/testfile** (obratiti pažnju na interpunkciju i mala i velika slova):

```
Juliet Capulet
The model identifier is DEn5c89zt.
Sarcastic was what he was.
No, he was just sarcastic.
Simplicity
The quick brown fox jumps over the lazy dog
It's a Cello? Not a Violin?
This character is (*) is the splat in Unix.
activity
apricot
capulet
cat
celebration
corporation
cot
cut
cutting
dc9tg4
eclectic
housecat
persnickety
The punctuation and capitalization is important in this example.
simplicity
undiscriminating
Two made up words below:
c?t
C?*t
cccot
cccccot
```

## Metakarakteri

Metakarakteri su korisni u redukciji količine teksta koji se koristi sa komandama i za reprezentaciju tekstualnih grupa minimalnim skupom karaktera. Sledeći metakarakteri su u široj upotrebi:

- **.** - **Tačka. Reprezentuje jedan karakter.** Primer: Pronaći bilo koju pojavu slova c i slova t sa tačno jednim karakterom između.

```
$ grep c.t /temp/testfile
```

```
Simplicity  
apricot  
cat  
cot  
cut  
cutting  
dc9tg4  
housecat  
simplicity  
c?t  
cccot  
cccccot
```

- **[] - uglaste zagrade. Rezultat odgovara bilo kojem karakteru unutar zagrada.**

Primer: Pronaći bilo koju instancu slova c i slova t sa tačno jednim samoglasnikom između.

```
$ grep c[aeiou]t /temp/testfile
```

```
Simplicity  
apricot  
cat  
cot  
cut  
cutting  
housecat  
simplicity  
cccot  
cccccot
```

- **\* - zvezda. Reprezentuje nula ili više pojava bilo kojih karaktera.**

Primer: Pronaći sve instance slova c i slova t sa nula ili više karaktera između njih.

```
$ grep c*t /temp/testfile
```

```
Juliet Capulet  
The model identifier is DEn5c89zt.  
Sarcastic was what he was.  
No, he was just sarcastic.  
Simplicity  
The quick brown fox jumps over the lazy dog  
It's a Cello? Not a Violin?  
This character is (*) is the splat in Unix.  
activity  
apricot  
capulet  
cat  
celebration  
corporation  
cot  
cut  
cutting  
dc9tg4  
eclectic  
housecat  
persnickety  
The punctuation and capitalization is important in this example.  
simplicity  
undiscriminating  
c?t  
C?*t  
cccot  
cccccot
```

- **[^karakter]** - uglaste zagrade sa kapom između. Nijedan od navedenih karaktera se NE pojavljuje.

Primer: Pronaći sve pojave karaktera c i karaktera 5, a da između njih ne stoji nikakav samoglasnik.

```
$ grep c[^aeiou]t /temp/testfile
dc9tg4
c?t
```

- **^karakter** - Odgovara sekvenci samo ako je na početku linije.

Primer: Pronaći sve pojave stringa ca na početku linije.

```
$ grep ^ca /temp/testfile
capulet
cat
```

Bez karaktera ^, sekvenca ca bi mogla da se nađe bilo gde u stringu:

```
Sarcastic was what he was.
No, he was just sarcastic.
capulet
cat
housecat
The punctuation and capitalization is important in this example.
```

- **^[karakter(i)]** - Kapa ispred sekvence u uglastim zagradama. Odgovara bilo kom karakteru u uglastim zagradama, ali na početku linije.

Primer: Pronaći sve pojave slova c praćenog samoglasnikom i slovom t na početku linije.

```
$ grep ^c[aoiue]t /temp/testfile
cat
cot
cut
cutting
```

Da je izostavljen karakter ^, sekvenca bi mogla biti bilo gde u liniji:

```
Simplicity
apricot
cat
cot
cut
cutting
housecat
simplicity
cccot
ccccot
```

- **\$** - Znak dolara. Odgovara pojavi sekvence karaktera na kraju linije.

Primer: Pronaći linije koje se završavaju slovima c i t između kojih može biti bilo šta.

```
$ grep c*t$ /temp/testfile
Juliet Capulet
apricot
capulet
cat
cot
cut
housecat
c?t
C?*t
cccot
ccccot
```

- **\** - Backslash. Anulira specijalno značenje karaktera koji ga neposredno sledi.



Primer: Pronaći sve pojave sekvence karaktera c?t.

- **? - Upitnik. Reprezentuje nula ili jednu pojavu karaktera (ne treba ga mešati sa \*, koji odgovara nula, jednom, ili više karaktera). Ne podržavaju ga svi UNIX programi.**

Primer: Pronaći sve pojave karaktera c i karaktera t sa jednim ili nijednim karakterom između njih.

- **[a-z] - Potpuna engleska abeceda malim slovima. Odgovara bilo kojem slovu abecede.**

Primer: Pronaći sve pojave karaktera c i t sa bilo kojim slovom između njih.

- **[0-9] - Odgovara bilo kojoj cifri.**

Primer: Pronaći sve pojave karaktera c i t sa bilo kojom cifrom između njih.

- **[d-m7-9] - Odgovara jednom pojavljivanju bilo kog karaktera iz opsega d-m ili 7-9. Primer ilustruje grupisanje komandi.**

Primer: Pronaći sve pojave slova c i slova t, sa jednim karakterom između koji može biti bilo koje slovo u opsegu od c do t ili cifra između 0 i 4.

## Prošireni regularni izrazi (extended regular expressions)

Kod proširenih regularnih izraza koji se aktiviraju npr. navođenjem opcije -E grep komandi, a dostupni su i u programima, awk, emacs, vi, itd. Karakteristični su sledeći metakarakter koji služe za označavanje ponavljanja karaktera ili podizraza (blokova):

- **(karakter)** - označeni podizraz (blok).
- **+** - Izraz od jednog karaktera nakon kojeg sledi "+" sparuje jednu ili više kopija izraza. Na primer, "ab+c" sparuje "abc", "abbbc" itd. "[xyz]+" sparuje "x", "y", "zx", "zyx", i tako dalje
- **{x,y}** - Sparuje poslednji blok barem "x" i ne više od "y" puta. Na primer, "a{3,5}" sparuje "aaa", "aaaa" ili "aaaaa"
- **{x}** - Prethodni karakter se pojavljuje tačno x puta
- **{x,}** - Prethodni karakter se pojavljuje najmanje x puta
- **{,y}** - Prethodni karakter se pojavljuje najviše y puta
- **?** - prethodni karakter je opcioni i pojavljuje se najviše jednom.

### Složen primer:

Fajl proba.txt:

```
www8.dobarsajt1.edu
www678.amu.ac.zu5l
www6.ailmit.net
www.euler.ni.ac.yu
mitcl.edu
www.core.amu.edu
www.znanje.edu
```

Izlaz komande:

```
$ grep -E "^(w{3}[0-9]*.)?[a-z]{1}[a-z0-9]{1,7}(\.ac\.[a-z]{2})(\.edu))$" proba.txt
mitcl.edu
www.znanje.edu
```

**Objašnjenje:** Navedeni regularni izraz znači da string može, a ne mora (zbog "?") početi sekvencom "www"+bilo koje cifre nakon čega sledi tačka. Iza tačke mora biti slovo, a zatim sekvenca od najmanje 1, a najviše 7 alfanumerika (slova ili brojeva), dok na kraju mora biti domen ".ac.(dva bilo koja slova)" ili ".edu".

## Domaći zadaci

### Prvi zadatak

Napisati regularni izraz koji pronalazi decimalne brojeve.

**Primer**

decimalni broj	3.14529
decimalni broj	-255.34
decimalni broj	128
decimalni broj	1.9e10
decimalni broj	123,340.00
nije decimalni broj	720p

## Drugi zadatak

Napisati regularni izraz koji pronalazi brojeve telefona.

**Primer ispravnih brojeva telefona**

415-555-1234  
650-555-2345  
(416)555-3456  
202 555 4567  
4035555678  
1 416 555 9292