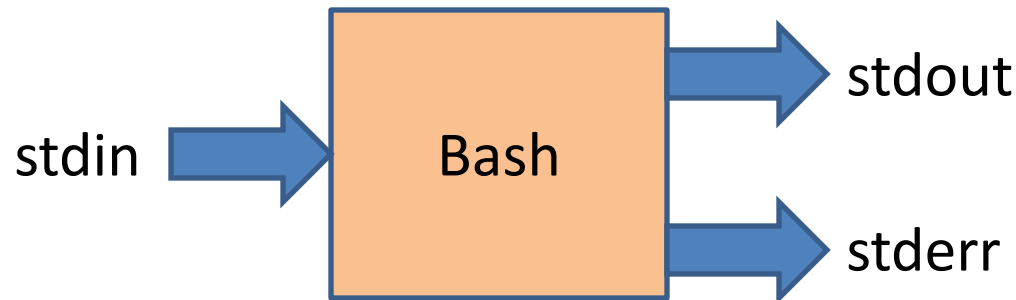


Strimovi

- Bash ima tri osnovna strima, jedan ulazni i dva izlazna
- Ulazni strim stdin (strim 0, fajl deskriptor 0) predstavlja standardni ulaz (input) u bash - tipično je u pitanju tastatura
- Izlazni strim stdout (strim 1, fajl deskriptor 1) predstavlja standardni izlaz (output) - tipično je u pitanju displej
- Izlazni strim stderr (strim 2, fajl deskriptor 2) predstavlja standardni izlaz (output) za poruke o grešakama - tipično je u pitanju displej
- Bash uzima ulaz iz stdin, vrši procesiranje zadatog i potom ispisuje izlazne poruke na stdout, a eventualne greške na stderr



Redirekcija strimova

- Nekad je zgodno vršiti takozvanu redirekciju strimova
- Na primer, umesto na displej, nekad je pogodnije ispisati poruke u fajl
- Ovo je naročito pogodno u slučaju grešaka za kasniju analizu
- Takođe, kao što ćemo videti redirekcija omogućava brisanje sadržaja fajla ili upis željenog sadržaja u fajl

Redirekcija izlaza

- Operator `>` se koristi za redirekciju standardnog izlaza (stdout)
- U stvari, ovaj operator je skraćenica od `1>` što predstavlja redirekciju stdout (`2>` predstavlja redirekciju stderr)
- Poruka stdout strima koja bi se ispisala na displeju, ovim operatorom redirekcije se može usmeriti na drugo mesto, tipično fajl
- Otuda se iza operatora navodi odredišno mesto (tipično fajl)
- Prilikom skeniranja linije, kad naiđe na operator `>`, bash će izvršiti brisanje sadržaja fajla koji predstavlja odredište redirekcije (OPREZ sa redirekcijom izlaza da ne obrišete neki sadržaj koji ne želite da obrišete)

Redirekcija izlaza - primeri

```
ubuntu@ubuntu-VirtualBox:~/Documents$ ls
ubuntu@ubuntu-VirtualBox:~/Documents$ echo tekst>fajl.txt
ubuntu@ubuntu-VirtualBox:~/Documents$ ls
fajl.txt
ubuntu@ubuntu-VirtualBox:~/Documents$ cat fajl.txt
tekst
ubuntu@ubuntu-VirtualBox:~/Documents$ echo novo > fajl.txt
ubuntu@ubuntu-VirtualBox:~/Documents$ cat fajl.txt
novo
ubuntu@ubuntu-VirtualBox:~/Documents$ ls > ../fajl2.txt
ubuntu@ubuntu-VirtualBox:~/Documents$ cat ../fajl2.txt
fajl.txt
ubuntu@ubuntu-VirtualBox:~/Documents$
```

Ako navedeni odredišni fajl ne postoji, biće kreiran. Otuda je prva redirekcija kreirala *fajl.txt*. Druga redirekcija je obrisala prethodni sadržaj fajla i u njega upisala novi sadržaj. Treći primer pokazuje da odredišni fajl ne mora biti u trenutnom direktorijumu, već se može navesti putanja (apsolutna ili relativna, u primeru je iskorišćena relativna putanja). U sva tri primera, izlaz nije ispisan na displej (izlazi **echo** i **ls** komandi), već u navedene odredišne fajlove.

Redirekcija izlaza - primeri

```
ubuntu@ubuntu-VirtualBox:~/Documents$ set -x
ubuntu@ubuntu-VirtualBox:~/Documents$ echo novo > fajl.txt
+ echo novo
ubuntu@ubuntu-VirtualBox:~/Documents$
```

Primetiti da se **echo** komanda izvršava regularno, samo se usled redirekcije rezultat **echo** komande ispisuje u fajl, a ne na displej.

```
ubuntu@ubuntu-VirtualBox:~/Documents$ ls
ubuntu@ubuntu-VirtualBox:~/Documents$ >fajl.txt
ubuntu@ubuntu-VirtualBox:~/Documents$ ls
fajl.txt
ubuntu@ubuntu-VirtualBox:~/Documents$ cat fajl.txt
ubuntu@ubuntu-VirtualBox:~/Documents$ echo tekst > fajl.txt
ubuntu@ubuntu-VirtualBox:~/Documents$ cat fajl.txt
tekst
ubuntu@ubuntu-VirtualBox:~/Documents$ > fajl.txt
ubuntu@ubuntu-VirtualBox:~/Documents$ cat fajl.txt
ubuntu@ubuntu-VirtualBox:~/Documents$ █
```

Ako se koristi samo operator redirekcije izlaza, tada se kreira prazan fajl (alternativa **touch** komandi) ili ako fajl postoji vrši se brisanje celokupnog sadržaja fajla.

Redirekcija izlaza

- Kad naiđe na operator redirekcije izlaza, bash će obrisati sadržaj navedenog odredišnog fajla, nezavisno od toga da li će komanda biti uspešno urađena ili ne

```
ubuntu@ubuntu-VirtualBox:~/Documents$ cat fajl.txt
tekst
ubuntu@ubuntu-VirtualBox:~/Documents$ echho novo > fajl.txt
No command 'echho' found, did you mean:
  Command 'echo' from package 'coreutils' (main)
echho: command not found
ubuntu@ubuntu-VirtualBox:~/Documents$ cat fajl.txt
ubuntu@ubuntu-VirtualBox:~/Documents$
```

- Kao što je rečeno operator `>` predstavlja skraćenicu od `1>`, stoga navođenje `1>` umesto `>` ima istu funkcionalnost

```
ubuntu@ubuntu-VirtualBox:~/Documents$ echo Novi tekst 1> fajl.txt
ubuntu@ubuntu-VirtualBox:~/Documents$ cat fajl.txt
Novi tekst
ubuntu@ubuntu-VirtualBox:~/Documents$
```

Sprečavanje brisanja sadržaja fajla

- Opcija *noclobber* sprečava da se prebriše sadržaj postojećeg fajla prilikom redirekcije izlaza (ako fajl ne postoji, onda je sve u redu jer nema prebrisavanja sadržaja)
- **set -o noclobber** aktivira ovu opciju, a **set +o noclobber** deaktivira ovu opciju
- Ako je aktivirana *noclobber* opcija, navođenjem operatora **>|** se premošćuje *noclobber* opcija za tu redirekciju
- Komanda **>| naziv fajla** kada je aktivirana *noclobber* opcija vrši brisanje sadržaja navedenog fajla
- Redirektovan izlaz se može i lepiti na postojeći sadržaj fajla (*append*) upotrebom operatora **>>** (novi tj. dodati sadržaj se lepi na kraj fajla)

Primeri

```
ubuntu@ubuntu-VirtualBox:~/Documents$ set -o noclobber
ubuntu@ubuntu-VirtualBox:~/Documents$ ls
ubuntu@ubuntu-VirtualBox:~/Documents$ echo tekst > fajl.txt
ubuntu@ubuntu-VirtualBox:~/Documents$ ls
fajl.txt
ubuntu@ubuntu-VirtualBox:~/Documents$ cat fajl.txt
tekst
ubuntu@ubuntu-VirtualBox:~/Documents$ echo Novi tekst > fajl.txt
bash: fajl.txt: cannot overwrite existing file
ubuntu@ubuntu-VirtualBox:~/Documents$ cat fajl.txt
tekst
ubuntu@ubuntu-VirtualBox:~/Documents$ echo Novi tekst >| fajl.txt
ubuntu@ubuntu-VirtualBox:~/Documents$ cat fajl.txt
Novi tekst
ubuntu@ubuntu-VirtualBox:~/Documents$ echo Dodati tekst >> fajl.txt
ubuntu@ubuntu-VirtualBox:~/Documents$ cat fajl.txt
Novi tekst
Dodati tekst
ubuntu@ubuntu-VirtualBox:~/Documents$ █
```


Redirekcija ispisa grešaka (stderr)

- Operator **2>** se koristi za redirekciju standardnog izlaza grešaka (stderr)
- Sve napomene oko prebrisavanja sadržaja fajlova, kreiranja fajla ako ne postoji, navedene za redirekciju stdout, važe i ovde (operator i princip rada su isti)
- Ako se kao odredište stavi */dev/null* onda će ispis završiti u “crnoj rupi” tj. nigde (ovo važi i za redirekciju stdout)
- Pomoću **2>&1** se vrši redirekcija stderr na isto mesto gde i stdout - pogodno ako se želi isti fajl koristiti za ispis i stdout i stderr strimova
- Ako ima više redirekcija navedenih u komandnoj liniji onda se one tumače sleva nadesno

Primeri

```
ubuntu@ubuntu-VirtualBox:~/Documents$ ls
ubuntu@ubuntu-VirtualBox:~/Documents$ eccho tekst 2>fajlerr.txt
ubuntu@ubuntu-VirtualBox:~/Documents$ cat fajlerr.txt
No command 'eccho' found, did you mean:
  Command 'echo' from package 'coreutils' (main)
eccho: command not found
ubuntu@ubuntu-VirtualBox:~/Documents$ pecho tekst 2>fajlerr.txt
ubuntu@ubuntu-VirtualBox:~/Documents$ cat fajlerr.txt
No command 'pecho' found, did you mean:
  Command 'bpecho' from package 'ion' (universe)
  Command 'aecho' from package 'netatalk' (universe)
  Command 'echo' from package 'coreutils' (main)
pecho: command not found
ubuntu@ubuntu-VirtualBox:~/Documents$ zecho tekst 2>>fajlerr.txt
ubuntu@ubuntu-VirtualBox:~/Documents$ cat fajlerr.txt
No command 'pecho' found, did you mean:
  Command 'bpecho' from package 'ion' (universe)
  Command 'aecho' from package 'netatalk' (universe)
  Command 'echo' from package 'coreutils' (main)
pecho: command not found
No command 'zecho' found, did you mean:
  Command 'aecho' from package 'netatalk' (universe)
  Command 'echo' from package 'coreutils' (main)
zecho: command not found
```

Primeri

```
ubuntu@ubuntu-VirtualBox:~/Documents$ cat fajlerr.txt
No command 'pecho' found, did you mean:
  Command 'bpecho' from package 'ion' (universe)
  Command 'aecho' from package 'netatalk' (universe)
  Command 'echo' from package 'coreutils' (main)
pecho: command not found
No command 'zecho' found, did you mean:
  Command 'aecho' from package 'netatalk' (universe)
  Command 'echo' from package 'coreutils' (main)
zecho: command not found
ubuntu@ubuntu-VirtualBox:~/Documents$ set -o noclobber
ubuntu@ubuntu-VirtualBox:~/Documents$ techo tekst 2>fajlerr.txt
bash: fajlerr.txt: cannot overwrite existing file
ubuntu@ubuntu-VirtualBox:~/Documents$ cat fajlerr.txt
No command 'pecho' found, did you mean:
  Command 'bpecho' from package 'ion' (universe)
  Command 'aecho' from package 'netatalk' (universe)
  Command 'echo' from package 'coreutils' (main)
pecho: command not found
No command 'zecho' found, did you mean:
  Command 'aecho' from package 'netatalk' (universe)
  Command 'echo' from package 'coreutils' (main)
zecho: command not found
```

```
ubuntu@ubuntu-VirtualBox:~/Documents$ ls
ubuntu@ubuntu-VirtualBox:~/Documents$ rm fajl 2>/dev/null
ubuntu@ubuntu-VirtualBox:~/Documents$ rm fajl
rm: cannot remove 'fajl': No such file or directory
```

Svrha treće komande u primeru je da se vidi da navedena komanda rezultuje greškom.

Ispis greške nije prikazan, niti je negde zapisan (druga komanda). Zgodna opcija za odredište je `/dev/null` kad se ne žele prikazivati ni zapisivati poruke grešaka.

Primeri

```
ubuntu@ubuntu-VirtualBox:~/Documents$ ls
fajl.txt
ubuntu@ubuntu-VirtualBox:~/Documents$ cat fajl.txt
ubuntu@ubuntu-VirtualBox:~/Documents$ eccho tekst > fajl.txt 2>&1
ubuntu@ubuntu-VirtualBox:~/Documents$ cat fajl.txt
No command 'eccho' found, did you mean:
  Command 'echo' from package 'coreutils' (main)
eccho: command not found
ubuntu@ubuntu-VirtualBox:~/Documents$ eccho tekst 2>&1 > fajl.txt
No command 'eccho' found, did you mean:
  Command 'echo' from package 'coreutils' (main)
eccho: command not found
ubuntu@ubuntu-VirtualBox:~/Documents$ cat fajl.txt
ubuntu@ubuntu-VirtualBox:~/Documents$ echo tekst 2>&1 > fajl.txt
ubuntu@ubuntu-VirtualBox:~/Documents$ cat fajl.txt
tekst
ubuntu@ubuntu-VirtualBox:~/Documents$ █
```

Redosled redirekcija je bitan. U prvom primeru redirekcije, `2>&1` je naveden na kraju pa se ta redirekcija tumači kao poslednja. Pre toga je izvršena redirekcija stdout na fajl.txt. Redirekcija `2>&1` usmerava stderr na isti fajl na koji je usmeren stdout, a to je fajl.txt. U sledeća dva primera, `2>&1` je prva redirekcija. Sada je stderr opet usmeren na isto odredište kao i stdout, ali pošto je to prva redirekcija, onda je stderr usmeren na displej jer je to originalno odredište za stdout, a druga redirekcija usmerava stdout na fajl.txt, ali to ništa ne znači za stderr koji je pre toga usmeren na displej. Kao što se vidi redosled je bitan.

Združena redirekcija izlaza

- Operator **&>** se koristi za združenu redirekciju stdout i stderr izlaza
- Operator **&>>** bi radio *append* varijantu

```
ubuntu@ubuntu-VirtualBox:~/Documents$ cat fajl.txt
ubuntu@ubuntu-VirtualBox:~/Documents$ echo tekst &> fajl.txt
ubuntu@ubuntu-VirtualBox:~/Documents$ cat fajl.txt
tekst
ubuntu@ubuntu-VirtualBox:~/Documents$ eccho tekst &> fajl.txt
ubuntu@ubuntu-VirtualBox:~/Documents$ cat fajl.txt
No command 'eccho' found, did you mean:
  Command 'echo' from package 'coreutils' (main)
eccho: command not found
ubuntu@ubuntu-VirtualBox:~/Documents$ echo Novi tekst &>> fajl.txt
ubuntu@ubuntu-VirtualBox:~/Documents$ cat fajl.txt
No command 'eccho' found, did you mean:
  Command 'echo' from package 'coreutils' (main)
eccho: command not found
Novi tekst
ubuntu@ubuntu-VirtualBox:~/Documents$ █
```

Redirekcija ulaza

- Operator `<` se koristi za redirekciju standardnog ulaza `stdin` (u suštini operator `<` je skraćenica od `0<`)
- Suština je da se umesto tastature ulazni sadržaj uzme sa drugog mesta, tj. iz fajla

```
ubuntu@ubuntu-VirtualBox:~/Documents$ cat fajl.txt
Tekst
ubuntu@ubuntu-VirtualBox:~/Documents$ cat < fajl.txt
Tekst
ubuntu@ubuntu-VirtualBox:~/Documents$ set -x
ubuntu@ubuntu-VirtualBox:~/Documents$ cat fajl.txt
+ cat fajl.txt
Tekst
ubuntu@ubuntu-VirtualBox:~/Documents$ cat < fajl.txt
+ cat
Tekst
ubuntu@ubuntu-VirtualBox:~/Documents$ □
```

Aktivacijom `set -x` možemo videti razliku kad se koristi i ne koristi `<` operator u slučaju `cat` komande. Kada nema `<` operatora onda se vidi da je argument `cat` komande `fajl.txt`. Ali, kada se koristi `<` operator onda nema argumenta `fajl.txt` jer se izvršila redirekcija pa `cat` ne zna odakle originalno potiče dobijen tekst.

Redirekcija ulaza - append

- Operator `<<` (tzv. *here document*) omogućava tumačenje unetog teksta (unetih linija teksta) kao stdin sve dok se ne unese zadata ključna reč za prekidanje **append** procesa
- Ako ključna reč nije pod navodnicima, unete linije su podložne shell ekspanziji (*parameter expansion, command substitution i arithmetic expansion*), a za karaktere ```, `$` i `\` se mora koristiti escape karakter `\` (posledica primene shell ekspanzije)
- Ako je bar nešto u ključnoj reči pod navodnicima onda se na unetim linijama ne radi shell expansion
- Na samoj ključnoj reči se ne radi shell expansion (takođe ako postoje navodnici oni ne ulaze u ključnu reč)
- Za prekidanje **append** procesa ključna reč se tumači bez navodnika

Primeri

```
ubuntu@ubuntu-VirtualBox:~/Documents$ cat << kraj
> prvi
> drugi
> kraj
prvi
drugi
ubuntu@ubuntu-VirtualBox:~/Documents$ cat << kraj
> $HOME
> $((7+7))
> kraj
/home/ubuntu
14
ubuntu@ubuntu-VirtualBox:~/Documents$ cat << "kraj"
> $HOME
> $((7+7))
> kraj
$HOME
$((7+7))
ubuntu@ubuntu-VirtualBox:~/Documents$ cat << 'kraj'
> $HOME
> $((7+7))
> kraj
$HOME
$((7+7))
ubuntu@ubuntu-VirtualBox:~/Documents$ █
```


Primeri

```
ubuntu@ubuntu-VirtualBox:~/Documents$ cat << 'kr'aj
> $HOME
> kraj
$HOME
ubuntu@ubuntu-VirtualBox:~/Documents$ cat << $HOME
> jedan
> $HOME
jedan
ubuntu@ubuntu-VirtualBox:~/Documents$ cat << $((7+7))
> $((5+5))
> dva
> $((7+7))
10
dva
ubuntu@ubuntu-VirtualBox:~/Documents$ cat << $HOME
> $USER
> $((7+7))
> \ $((7+7))
> kraj
> $HOME
ubuntu
14
$((7+7))
kraj
ubuntu@ubuntu-VirtualBox:~/Documents$ █
```

Napomena: ako se kao ključna reč stavi EOF, onda se za kraj može staviti ili EOF ili pritisnuti CTRL+D

Redirekcija ulaza - stringovi

- Operator <<< (tzv. *here strings*) omogućava da se uneti string prosledi komandi kao argument (kao da je unet sa stdin)
- Sam string je podlozan shell expansion procesu (sem *filename expansion* i *word splitting*) - rezultat shell expansion procesa se prosleđuje kao jedinstven string komandi kao da stiže sa stdin
- Napomena: *word splitting* podrazumeva razbijanje na stringa na delove razdvojene space karakterom ili nekim drugim delimiterom (u prethodnoj prezentaciji smo videli da upotrebom navodnika to možemo sprečiti)

Redirekcija ulaza - stringovi

- Alternativa operatoru <<< je upotreba **echo string | komanda** tj. pajp u kome echo prosleđuje string komandi
- Ova alternativa nije u potpunosti ekvivalentna kao što ćemo videti na jednom od sledećih slajdova

Primeri

```
ubuntu@ubuntu-VirtualBox:~/Documents$ cat <<<tekst
tekst
ubuntu@ubuntu-VirtualBox:~/Documents$ cat <<< $HOME
/home/ubuntu
ubuntu@ubuntu-VirtualBox:~/Documents$ cat <<< $((7*7))
49
ubuntu@ubuntu-VirtualBox:~/Documents$ cat <<< \\direktorijum
\direktorijum
ubuntu@ubuntu-VirtualBox:~/Documents$ cat <<<$HOME neki tekst
cat: neki: No such file or directory
cat: tekst: No such file or directory
ubuntu@ubuntu-VirtualBox:~/Documents$ cat <<<'$HOME neki tekst'
$HOME neki tekst
ubuntu@ubuntu-VirtualBox:~/Documents$ cat <<<$HOME <<<neki <<<tekst
tekst
```

U pretposlednjem primeru, da su korišćeni dvostruki navodnici, \$HOME bi bio zamenjen svojom vrednošću - ovde bi to bilo /home/ubuntu.

Primeri

```
ubuntu@ubuntu-VirtualBox:~$ tr a-z A-Z
rec
REC
^C
ubuntu@ubuntu-VirtualBox:~$ tr a-z A-Z <<< rec
REC
```

U prvom primeru se zasebno ukucava string, u drugom primeru string je direktno prosleđen komandi. Ako bi se izostavio <<< u drugom primeru prijavila bi se greška jer **tr** komanda ne očekuje string odmah, već samo definicije skupova karaktera (skup šta se menja i skup čime se menja). Alternativa upotrebi <<< je upotreba **echo** komande čiji se ispis prosleđuje preko pajpa drugoj komandi - u ovom slučaju bi imali izvršenje dve komande jer se sada izvršava i **echo** komanda.

Primeri

```
ubuntu@ubuntu-VirtualBox:~$ echo "jedan dva"|read var1 var2
ubuntu@ubuntu-VirtualBox:~$ echo $var1 $var2

ubuntu@ubuntu-VirtualBox:~$ read var1 var2 <<< "jedan dva"
ubuntu@ubuntu-VirtualBox:~$ echo $var1 $var2
jedan dva
```

Postoji suptilna razlika u upotrebi echo varijante i <<< varijante. U slučaju echo varijante u prvom primeru, drugi deo u pajpu (read deo) se izvršava u shell detetu (podshell) jer pajp podrazumeva da se komande izvršavaju u podshell-ovima i kao što je navedeno u prethodnoj prezentaciji varijable se ne prenose iz shell deteta u shell roditelj. Zato u prvom primeru nema ispisa vrednosti var1 i var2 varijabli jer one ni ne postoje. U drugom primeru nema takvih problema.

Redirekcija - kombinovanje

- Moguće je i kombinovanje redirekcije ulaza i izlaza

```
ubuntu@ubuntu-VirtualBox:~/Documents$ cat fajl.txt
This is a sentence. This is anoter sentence.
ubuntu@ubuntu-VirtualBox:~/Documents$ cat greske.txt
ubuntu@ubuntu-VirtualBox:~/Documents$ spell < fajl.txt > greske.txt
ubuntu@ubuntu-VirtualBox:~/Documents$ cat greske.txt
anoter
ubuntu@ubuntu-VirtualBox:~/Documents$
```

U gornjem primeru je korišćena aplikacija **spell** za proveru ispravnosti teksta. Primer je urađen u Ubuntu distribuciji u kojoj ova aplikacija nije instalirana po difoltu, već treba dodatno da se instalira.

```
ubuntu@ubuntu-VirtualBox:~/Documents$ cat << kraj > fajl.txt
> prvi
> drugi
> kraj
ubuntu@ubuntu-VirtualBox:~/Documents$ cat fajl.txt
prvi
drugi
ubuntu@ubuntu-VirtualBox:~/Documents$ cat <<kraj >>fajl.txt
> treci
> kraj
ubuntu@ubuntu-VirtualBox:~/Documents$ cat fajl.txt
prvi
drugi
treci
ubuntu@ubuntu-VirtualBox:~/Documents$
```

Redirekcija - kombinovanje

- Redirektovan izlaz može biti i stderr

```
ubuntu@ubuntu-VirtualBox:~/Documents$ cat <<kraj 2>error.txt
> prvi
> drugi
> kraj
prvi
drugi
ubuntu@ubuntu-VirtualBox:~/Documents$ cat error.txt
ubuntu@ubuntu-VirtualBox:~/Documents$ cat <<kraj &>sve.txt
> treci
> cetvrti
> kraj
ubuntu@ubuntu-VirtualBox:~/Documents$ cat sve.txt
treci
cetvrti
```

Primetiti u prvom primeru da dolazi do ispisa unetog teksta na displej. Razlog je što stdout nije redirektovan, već samo stderr.

Redirekcija - kombinovanje

- Mogu se navesti odvojeni fajlovi za stdout i stderr

```
ubuntu@ubuntu-VirtualBox:~/Documents$ echo Tekst. >fajl.txt 2>error.txt
ubuntu@ubuntu-VirtualBox:~/Documents$ cat fajl.txt
Tekst.
ubuntu@ubuntu-VirtualBox:~/Documents$ cat error.txt
ubuntu@ubuntu-VirtualBox:~/Documents$ eccho Tekst. >fajl.txt 2>error.txt
ubuntu@ubuntu-VirtualBox:~/Documents$ cat fajl.txt
ubuntu@ubuntu-VirtualBox:~/Documents$ cat error.txt
No command 'eccho' found, did you mean:
  Command 'echo' from package 'coreutils' (main)
eccho: command not found
ubuntu@ubuntu-VirtualBox:~/Documents$
```

```
ubuntu@ubuntu-VirtualBox:~/Documents$ 2>error.txt eccho tekst >fajl.txt
ubuntu@ubuntu-VirtualBox:~/Documents$ cat fajl.txt
ubuntu@ubuntu-VirtualBox:~/Documents$ cat error.txt
No command 'eccho' found, did you mean:
  Command 'echo' from package 'coreutils' (main)
eccho: command not found
ubuntu@ubuntu-VirtualBox:~/Documents$ >fajl.txt echo tekst 2>error.txt
ubuntu@ubuntu-VirtualBox:~/Documents$ cat error.txt
ubuntu@ubuntu-VirtualBox:~/Documents$ cat fajl.txt
tekst
```

Redosled redirekcija sada može da bude proizvoljan jer svaka redirekcija ima svoj fajl. Bash ionako prvo proverava redirekcije pre samih komandi.

Pajpovi

- Pajpovi omogućavaju povezivanje komandi u lanac tj. oni omogućavaju da se izlaz (rezultat) jedne komande iskoristi kao ulaz sledeće komande u nizu
- Niz može da sadrži proizvoljan broj komandi tj. može se 'pajpovati' proizvoljan broj komandi
- Operator | se koristi za kreiranje pajpa
- Svaka komanda u pajpu se izvršava u zasebnom podshell-u (tj. shell potomku)

Primer

```
bash
bunzip2
busybox
bzcat
bzcmp
bzdiff
bzegrep
bzexe
bzfgrep
bzgrep
bzip2
bzip2recover
bzless
bzmore
cat
chacl
chgrp
chmod
chown
chvt
cp
cpio
dash
date
dd
df
:█
```

```
dir
dmesg
dnsdomainname
domainname
dumpkeys
echo
ed
egrep
false
fgconsole
fgrep
findmnt
fuser
fusermount
getfacl
grep
gunzip
gzexe
gzip
hostname
ip
journalctl
kbd_mode
kill
kmod
less
:█
```

Rezultat izvršenja komande **ls /bin | less**. Rezultat **ls** komande predstavlja ulaz za **less** komandu koja omogućava da se lista fajlova prikazuje stranu po stranu, a ne sve odjednom. Sa leve strane su prikazane prve dve strane ispisa.

Kombinovanje sa redirekcijom

- U kombinaciju se može uključiti i redirekcija
- Na primer, konačan rezultat fajla se može upisati u neki fajl kao što ilustruje primer dat na ovom slajdu
- Da redirekcija nije uključena, **sort** komanda bi rezultat ispisala u terminal
- Komanda **sort** vrši sortiranje linija

```
ubuntu@ubuntu-VirtualBox:~/Documents$ cat fajl.txt
dd
tt
aa
zz
ii
ubuntu@ubuntu-VirtualBox:~/Documents$ cat < fajl.txt | sort >sortfajl.txt
ubuntu@ubuntu-VirtualBox:~/Documents$ cat sortfajl.txt
aa
dd
ii
tt
zz
ubuntu@ubuntu-VirtualBox:~/Documents$
```

Broj pajpova

- Može se pajpovati proizvoljan broj komandi
- U primeru na ovom slajdu su povezane tri komande pajpom
- Listing sadržaja direktorijuma /bin predstavlja ulaz u komandu **head** koja uzima samo prva tri zapisa, a taj rezultat predstavlja ulaz u komandu **tail** koja od ta tri zapisa uzima poslednji i na kraju se samo on ispisuje u terminalu

```
ubuntu@ubuntu-VirtualBox:~/Documents$ ls /bin | head -3 | tail -1  
busybox  
ubuntu@ubuntu-VirtualBox:~/Documents$
```

Filtriranje

- U prethodnim primerima videli smo da pajpovi mogu biti veoma pogodno (i moćno) sredstvo za procesiranje rezultata neke komande
- Otuda se često komande koje omogućavaju filtriranje rezultata po nekom kriterijumu koriste u pajp kombinacijama

cat i tac

- Ove dve komande smo već ranije spominjali
- Kada se **cat** koristi unutar pajpa, on samo prosleđuje primljeni ulaz na izlaz (taj izlaz je sledeća komanda u pajp nizu) ili stdout izlaz (ili redirektovan izlaz) ako je **cat** poslednja karika u pajp nizu - ova napomena o izlazu generalno važi i za ostale komande)
- Komanda **tac** obrće redosled linija primljenog ulaza i to prosleđuje na izlaz

```
ubuntu@ubuntu-VirtualBox:~/Documents$ cat fajl.txt
jedan
dva
tri
cetiri
ubuntu@ubuntu-VirtualBox:~/Documents$ cat fajl.txt | cat | cat | cat
jedan
dva
tri
cetiri
ubuntu@ubuntu-VirtualBox:~/Documents$ cat fajl.txt | cat | tac | cat
cetiri
tri
dva
jedan
ubuntu@ubuntu-VirtualBox:~/Documents$ cat fajl.txt | tac | tac | cat
jedan
dva
tri
cetiri
ubuntu@ubuntu-VirtualBox:~/Documents$ █
```

Primetiti da se osim prvog poziva **cat** funkcije, u ostalim pozivima tj. fazama pajpa ne stavlja argument jer se koristi rezultat prethodne faze pajpa.

Komanda tee

- Komanda **tee** ima veoma slično ponašanje kao **cat**
- Razlika je što **tee** produkuje dva identična izlaza - jedan na stdout, a jedan u fajl (koji predstavlja argument **tee** komande)
- Cilj **tee** komande je da omogući da se u neki fajl upiše međurezultat pajpa, što može biti zgodno ako tražimo grešku u pajpu (debugujemo pajp)

Primer

```
ubuntu@ubuntu-VirtualBox:~/Documents$ cat fajl.txt | tac | tee test.txt | tac
jedan
dva
tri
cetiri
ubuntu@ubuntu-VirtualBox:~/Documents$ cat test.txt
cetiri
tri
dva
jedan
ubuntu@ubuntu-VirtualBox:~/Documents$ █
```

```
ubuntu@ubuntu-VirtualBox:~/Documents$ cat fajl.txt | tac > test2.txt | tac
ubuntu@ubuntu-VirtualBox:~/Documents$ █
```

Ako kojim slučajem stavimo redirekciju unutar pajpa, a ne u poslednju fazu dolazi do neželjenog rada pajpa. Naime, rezultat **tac** komande u drugoj fazi je završio u fajlu, a ne u stdout, pa samim tim treća faza prima prazan ulaz što nije ono što želimo. Naravno, u poslednjoj fazi svejedno je da li redirekciju stavljamo na kraj ili početak te poslednje faze.

Komanda grep

- Ova komanda je veoma popularna za filtriranje rezultata
- Omogućava da se ispišu samo one linije koje sadrže (ili ne sadrže) zadati pattern
- Postoji velik broj opcija za ovu komandu koje ćemo obraditi na narednim slajdovima
- Ako se primenjuje samostalno onda se nakon opcija i patterna (pattern ide iza opcija) navodi jedan ili više fajlova koji se pretražuju
- Ako se primenjuje kao deo fajpa što često i jeste slučaj onda se izlaz prethodne faze fajpa koristi kao ulaz koji se filtrira

Regularni izrazi

- Regularni izraz (*regular expression*) predstavlja tekstualni string koji opisuje pattern koji se traži
- Grep je skraćenica od Global regular expression print
- Postoje tri verzije sintakse za regularne izraze:
 - BRE - Basic Regular Expressions
 - ERE - Extended Regular Expressions
 - PRE - Perl Regular Expressions
- Po defaultu se koristi BRE sintaksa u **grep** komandi (opcija -G ako bi se eksplicitno navodila)
- Opcijom -E koristi se ERE sintaksa, a opcijom -P koristi se PRE sintaksa
- Opcija -F forsira da se string traži onako kako je zadat, tj. ne primenjuje se sintaksa regularnog izraza

Operatori u regularnim izrazima

- $.$ - predstavlja džoker koji se poklapa sa bilo kojim karakterom (sem null)
- $?$ - prethodnik je opcion tj. treba da se pojavi nula puta ili jednom
- $*$ - prethodnik treba da se pojavi 0 ili više puta
- $+$ - prethodnik treba da se pojavi 1 ili više puta
- $\{n\}$ - prethodnik treba da se pojavi tačno n puta
- $\{n,\}$ - prethodnik treba da se pojavi tačno n ili više puta
- $\{n, m\}$ - prethodnik treba da se pojavi bar n puta, ali ne više od m puta
- $^$ - ako se koristi na početku samog regularnog izraza ima ulogu *anchor*-a tj. navedeni regularni izraz mora da se javi na početku linije, a ako se koristi na početku tzv. *bracket expression* onda ima ulogu negacije skupa

Operatori u regularnim izrazima

- \$ - ako se koristi na kraju samog regularnog izraza ima ulogu *anchor*-a tj. navedeni regularni izraz mora da se javi na kraju linije
- \< - regularni izraz mora da se poklopi sa početkom reči
- \> - regularni izraz mora da se poklopi sa krajem reči
- \b - regularni izraz mora da se poklopi sa ivicom reči (ako se stavi ispred izraza onda je u pitanju početak reči, a ako je iza izraza u pitanju je kraj reči - moguće je staviti operator i ispred i iza izraza)
- \B - regularni izraz ne može da bude na navedenoj ivici (princip isti kao u prethodnoj stavci)
- | - ovo je I|I operator koji traži poklapanje sa jednim ili drugim izrazom koji se nalaze oko operatora (tzv. alternacija)
- Obične zagrade se mogu koristiti za grupisanje tako da prethodnik ne bude jedan karakter već string

BRE i ERE

- U BRE karakteri ?, +, {, |, (i) gube svoje specijalno značenje (zagrada i karakter | se navode sa \ ispred da bi se tumačili kao što je navedeno na prethodnim slajdovima, dok se ? i + uvek tumače kao obični karakteri)
- Detalji o BRE i ERE se mogu videti na linkovima http://pubs.opengroup.org/onlinepubs/009696899/basedefs/xbd_chap09.html#tag_09_03 i https://en.wikibooks.org/wiki/Regular_Expressions/POSIX_Basic_Regular_Expressions
- Uvek je poželjno proveriti dokumentaciju komande koja koristi regularne izraze (poput **grep** komande) i videti da li postoje neke modifikacije ili ekstenzije u odnosu na standard

Bracket expression

- *Bracket expression* predstavlja izraz koji se navodi u uglastim zagradama [], a dotični izraz predstavlja jedan karakter iz skupa karaktera koji su predstavljeni izrazom u uglastoj zagradi
- Unutar zagrade se navode karakteri skupa jedan za drugim pri čemu redosled nije bitan
- Unutar zagrade se može navesti opseg karaktera korišćenjem - pri čemu ispred - ide početak opsega, a iza - ide kraj opsega (na primer, a-f predstavlja niz malih slova od a do f)
- Ako se navodi više opsega, ne može da kraj jednog opsega bude istovremeno početak drugog opsega tj. nepravilno je pisati a-f-m kao zamenu za opsege a-f i f-m

Bracket expression

- Može se navesti klasa karaktera - ona se navodi unutar `[:naziv_klase:]` pri čemu nazivi klase mogu biti `alnum`, `alpha`, `digit`, `blank`, `cntrl`, `graph`, `lower`, `upper`, `punct`, `print`, `xdigit`, `space` (navedene klase bi uvek trebale biti podržane)
- Ako se na početku *bracket expression* stavi `^` onda je u pitanju negacija skupa tj. traži se karakter koji ne pripada skupu definisanom u *bracket expression*

Primeri

```
ubuntu@ubuntu-VirtualBox:~/Documents$ cat primer.txt
123
abc
zec
Mis
promisliti
misolovka
skocimis
mimikrija
ubuntu@ubuntu-VirtualBox:~/Documents$ grep mi primer.txt
promisliti
misolovka
skocimis
mimikrija
ubuntu@ubuntu-VirtualBox:~/Documents$ grep [[:digit:]] primer.txt
123
ubuntu@ubuntu-VirtualBox:~/Documents$ grep s[m-p] primer.txt
misolovka
ubuntu@ubuntu-VirtualBox:~/Documents$ grep ^mi primer.txt
misolovka
mimikrija
ubuntu@ubuntu-VirtualBox:~/Documents$ grep c$ primer.txt
abc
zec
ubuntu@ubuntu-VirtualBox:~/Documents$ █
```

Primeri

```
1223
12322
122242
ubuntu@ubuntu-VirtualBox:~/Documents$ grep 3*4 primer.txt
122242
ubuntu@ubuntu-VirtualBox:~/Documents$ grep -E 3*4 primer.txt
122242
ubuntu@ubuntu-VirtualBox:~/Documents$ grep 3+4 primer.txt
ubuntu@ubuntu-VirtualBox:~/Documents$ grep -E 3+4 primer.txt
ubuntu@ubuntu-VirtualBox:~/Documents$ grep 2+4 primer.txt
ubuntu@ubuntu-VirtualBox:~/Documents$ grep -E 2+4 primer.txt
122242
ubuntu@ubuntu-VirtualBox:~/Documents$ grep 3?4 primer.txt
ubuntu@ubuntu-VirtualBox:~/Documents$ grep -E 3?4 primer.txt
122242
ubuntu@ubuntu-VirtualBox:~/Documents$ █
```

ERE koristi ? i + operatore, a BRE ne kao što se može videti iz gornjeg primera. Ni stavljanje \ karaktera ispred + ne bi pomoglo.

```
ubuntu@ubuntu-VirtualBox:~/Documents$ grep z.c primer.txt
zec
ubuntu@ubuntu-VirtualBox:~/Documents$ grep c. primer.txt
skocimis
ubuntu@ubuntu-VirtualBox:~/Documents$ grep .c primer.txt
abc
zec
skocimis
```

Primeri

Primetiti upotrebu () za grupisanje stringa 12. Regularni izraz je stavljen pod jednostruke navodnike da ne dođe do greške kad bash tumači liniju (u regularnim izrazima je korišćen escape karakter \). U suštini, ako se želi biti siguran nije loše uvek staviti regularni izraz pod jednostruke navodnike. Poslednji primer koristi nestandardizovanu formu {,m} koja ipak radi u bash-u ali potencijalni problem ako se koristi samostalno je što dozvoljava i 0 poklapanja.

```
ubuntu@ubuntu-VirtualBox:~/Documents$ cat primer2.txt
123
23121233
231121212341212127781212
45121254678
5544121255588
2377754
ubuntu@ubuntu-VirtualBox:~/Documents$ grep '(12)\{2\}' primer2.txt
23121233
231121212341212127781212
45121254678
5544121255588
ubuntu@ubuntu-VirtualBox:~/Documents$ grep '(12)\{3\}' primer2.txt
231121212341212127781212
ubuntu@ubuntu-VirtualBox:~/Documents$ grep '(7)\{2,3\}' primer2.txt
231121212341212127781212
2377754
ubuntu@ubuntu-VirtualBox:~/Documents$ grep '7\{2,3\}' primer2.txt
231121212341212127781212
2377754
ubuntu@ubuntu-VirtualBox:~/Documents$ grep '(12)\{2,\}' primer2.txt
23121233
231121212341212127781212
45121254678
5544121255588
ubuntu@ubuntu-VirtualBox:~/Documents$ grep '(12)\{,3\}' primer2.txt
123
23121233
231121212341212127781212
45121254678
5544121255588
2377754
```

Primeri

Kada se koristi ERE tada ne treba da se stavljaju \ ispred zagrada kao u BRE slučaju. Napomena za jednostruke navodnike i dalje važi.

```
ubuntu@ubuntu-VirtualBox:~/Documents$ grep -E '(12){2}' primer2.txt
23121233
231121212341212127781212
45121254678
5544121255588
ubuntu@ubuntu-VirtualBox:~/Documents$ grep -E '(12){3}' primer2.txt
231121212341212127781212
ubuntu@ubuntu-VirtualBox:~/Documents$ grep -E '(7){2,3}' primer2.txt
231121212341212127781212
2377754
ubuntu@ubuntu-VirtualBox:~/Documents$ grep -E '7{2,3}' primer2.txt
231121212341212127781212
2377754
ubuntu@ubuntu-VirtualBox:~/Documents$ grep -E '(12){2,}' primer2.txt
23121233
231121212341212127781212
45121254678
5544121255588
ubuntu@ubuntu-VirtualBox:~/Documents$ grep -E '(12){,3}' primer2.txt
123
23121233
231121212341212127781212
45121254678
5544121255588
2377754
```

Primeri

```
ubuntu@ubuntu-VirtualBox:~/Documents$ cat primer3.txt
1aba
1a2a
2b3c
22cc
de34
brdo
mana
ubuntu@ubuntu-VirtualBox:~/Documents$ grep [[:digit:]][[:digit:]] primer3.txt
22cc
de34
ubuntu@ubuntu-VirtualBox:~/Documents$ grep [[:digit:]][[:lower:]] primer3.txt
1aba
1a2a
2b3c
22cc
ubuntu@ubuntu-VirtualBox:~/Documents$ grep '[:digit:]\{2\}' primer3.txt
22cc
de34
ubuntu@ubuntu-VirtualBox:~/Documents$ grep [b-d][^aeiou] primer3.txt
2b3c
22cc
brdo
ubuntu@ubuntu-VirtualBox:~/Documents$ grep [m-rc-d][^aeiou] primer3.txt
22cc
brdo
ubuntu@ubuntu-VirtualBox:~/Documents$ grep b[^aeiou] primer3.txt
2b3c
brdo
ubuntu@ubuntu-VirtualBox:~/Documents$ grep br[^aeiou] primer3.txt
brdo
ubuntu@ubuntu-VirtualBox:~/Documents$ █
```

Primeri

```
ubuntu@ubuntu-VirtualBox:~/Documents$ cat primer4.txt
jedna rec
preradjen proizvod
redni broj
dobre namere
gore dole
ne prestajati
pozvati redare
do re mi
ubuntu@ubuntu-VirtualBox:~/Documents$ grep ^re primer4.txt
redni broj
ubuntu@ubuntu-VirtualBox:~/Documents$ grep re$ primer4.txt
dobre namere
pozvati redare
ubuntu@ubuntu-VirtualBox:~/Documents$ grep 're\b' primer4.txt
dobre namere
gore dole
pozvati redare
do re mi
ubuntu@ubuntu-VirtualBox:~/Documents$ grep '\bre' primer4.txt
jedna rec
redni broj
pozvati redare
do re mi
ubuntu@ubuntu-VirtualBox:~/Documents$ grep '\bre\b' primer4.txt
do re mi
```

Primeri

```
ubuntu@ubuntu-VirtualBox:~/Documents$ grep 're\B' primer4.txt
jedna rec
preradjen proizvod
redni broj
ne prestajati
pozvati redare
ubuntu@ubuntu-VirtualBox:~/Documents$ grep '\Bre' primer4.txt
preradjen proizvod
dobre namere
gore dole
ne prestajati
pozvati redare
ubuntu@ubuntu-VirtualBox:~/Documents$ grep '\Bre\B' primer4.txt
preradjen proizvod
ne prestajati
ubuntu@ubuntu-VirtualBox:~/Documents$ grep '\Bre\b' primer4.txt
dobre namere
gore dole
pozvati redare
```

Primeri

```
ubuntu@ubuntu-VirtualBox:~/Documents$ grep '\<re' primer4.txt
jedna rec
redni broj
pozvati redare
do re mi
ubuntu@ubuntu-VirtualBox:~/Documents$ grep 're\>' primer4.txt
dobre namere
gore dole
pozvati redare
do re mi
ubuntu@ubuntu-VirtualBox:~/Documents$ grep '\<re\>' primer4.txt
do re mi
ubuntu@ubuntu-VirtualBox:~/Documents$ grep '\<r....e\>' primer4.txt
pozvati redare
ubuntu@ubuntu-VirtualBox:~/Documents$ grep '\<r...e\>' primer4.txt
ubuntu@ubuntu-VirtualBox:~/Documents$ grep '\<.....e\>' primer4.txt
dobre namere
pozvati redare
```

```
ubuntu@ubuntu-VirtualBox:~/Documents$ grep 'do\|od' primer4.txt
preradjen proizvod
dobre namere
gore dole
do re mi
ubuntu@ubuntu-VirtualBox:~/Documents$ grep -E 'do|od' primer4.txt
preradjen proizvod
dobre namere
gore dole
do re mi
```


Ispis konteksta

- Komanda `grep` omogućava i ispis tzv. konteksta
- Pod kontekstom se podrazumevaju linije koje se nalaze oko linije koja sadrži poklapanje sa traženim terminom
- Opcija `-A` ispisuje linije iza, opcija `-B` linije ispred, a opcija `-C` linije sa obe strane - naravno, uvek se ispisuje linija koja sadrži poklapanje
- Uz ove opcije se navodi broj koji predstavlja broj linija, na primer, `-A2` bi ispisala dve linije iza
- Ovo je veoma pogodno kad se, na primer, pretražuje dokumentacija, programski kod ili konfiguracioni fajl i sl. pa želimo da vidimo okolinu linije koja sadrži traženi termin

Ispis konteksta - primer

```
ubuntu@ubuntu-VirtualBox:~/Documents$ cat primer4.txt
jedna rec
preradjen proizvod
redni broj
dobre namere
gore dole
ne prestajati
pozvati redare
do re mi
ubuntu@ubuntu-VirtualBox:~/Documents$ grep -A1 dob primer4.txt
dobre namere
gore dole
ubuntu@ubuntu-VirtualBox:~/Documents$ grep -B1 dob primer4.txt
redni broj
dobre namere
ubuntu@ubuntu-VirtualBox:~/Documents$ grep -C1 dob primer4.txt
redni broj
dobre namere
gore dole
```

Još opcija

- Opcija -i ignoriše velika i mala slova tj. case insensitive je pretraga
- Opcija -w forsira da se traženi termin poklapa se celom reči
- Opcija -x forsira da se traženi termin poklapa se celom linijom
- Opcija -v invertuje rezultat pretrage tj. prikazuju se samo linije koje ne sadrže traženi termin
- Opcija -n ispisuje redni broj linije koja sadrži traženi termin
- Opcija -m zaustavlja pretragu nakon što je nađeno *m* linija koje zadovoljavaju kriterijum pretrage
- Opcija -c ispisuje broj linija koje zadovoljavaju kriterijum pretrage (broj linija po fajlu)
- Postoje i druge opcije koje nisu ovde navedene

Još opcija

```
ubuntu@ubuntu-VirtualBox:~/Documents$ grep mi primer.txt
promisliti
misolovka
skocimis
mimikrija
ubuntu@ubuntu-VirtualBox:~/Documents$ grep -i mi primer.txt
Mis
promisliti
misolovka
skocimis
mimikrija
ubuntu@ubuntu-VirtualBox:~/Documents$ grep -v mi primer.txt
123
abc
zec
Mis
1223
12322
122242
ubuntu@ubuntu-VirtualBox:~/Documents$ grep -vi mi primer.txt
123
abc
zec
1223
12322
122242
```

Još opcija

```
ubuntu@ubuntu-VirtualBox:~/Documents$ grep -w re primer4.txt
do re mi
ubuntu@ubuntu-VirtualBox:~/Documents$ grep -x re primer4.txt
ubuntu@ubuntu-VirtualBox:~/Documents$ grep -x 'do re mi' primer4.txt
do re mi
ubuntu@ubuntu-VirtualBox:~/Documents$ grep -n mi primer.txt
5:promisliti
6:misolovka
7:skocimis
8:mimikrija
ubuntu@ubuntu-VirtualBox:~/Documents$ grep -c mi primer.txt
4
ubuntu@ubuntu-VirtualBox:~/Documents$ [grep -m2 mi primer.txt
promisliti
misolovka
ubuntu@ubuntu-VirtualBox:~/Documents$ grep -nm2 mi primer.txt
5:promisliti
6:misolovka
ubuntu@ubuntu-VirtualBox:~/Documents$ grep -m2n mi primer.txt
grep: invalid max count
ubuntu@ubuntu-VirtualBox:~/Documents$ grep -m2 -n mi primer.txt
5:promisliti
6:misolovka
```

Primetiti da bash nije mogao ispravno protumačiti opciju `-m2n` jer je iza opcije `m` mora da ide broj, a `2n` nije validan broj. Opcija `n` se ili piše zasebno ili ispred opcije `m` u ovom slučaju.

Pajp

- Grep se veoma često koristi u pajpu za filtriranje rezultata

U ovom primeru iz listinga sadržaja direktorijuma izdvajamo samo fajlove čiji naziv počinje sa tačkom. Na dnu je prikazana jedna alternativa iste komande.

```
ubuntu@ubuntu-VirtualBox:~$ ls -a | grep ^[.]
.
..
.bash_history
.bash_logout
.bashrc
.cache
.compiz
.config
.dmrc
.gconf
.gnome2
.ICEauthority
.local
.mozilla
.profile
.sudo_as_admin_successful
.Xauthority
.xsession-errors
.xsession-errors.old
ubuntu@ubuntu-VirtualBox:~$ ls -a | grep '^\.'
```

Pajp - primer 2

Primer kako se **grep** može iskoristiti za višestepeno filtriranje. U prvom primeru je filtriranje izlistalo samo fajlove koji nisu direktorijumi. U drugom primeru je dodatno filtrirana lista fajlova da se izdvoje samo oni koji počinju na tačku tj. skriveni fajlovi. Ovaj primer prikazuje i da je prilikom građenja pajpova uvek zgodno ispitati ispravnost pajpa fazu po fazu nego napisati kompletan pajp odjednom jer ako dođe do greške u radu pajpa teže ju je otkriti ako nije urađena provera korak po korak.

```
ubuntu@ubuntu-VirtualBox:~$ ls -al | grep ^[^\d]
total 128
-rw----- 1 ubuntu ubuntu 12288 Aug 16 18:43 .bash_history
-rw-r--r-- 1 ubuntu ubuntu  220 May  1 2015 .bash_logout
-rw-r--r-- 1 ubuntu ubuntu  3804 Aug  9 07:57 .bashrc
-rw-r--r-- 1 ubuntu ubuntu   25 May  1 2015 .dmrc
-rw-r--r-- 1 ubuntu ubuntu  8980 May  1 2015 examples.desktop
-rw----- 1 ubuntu ubuntu  5430 Aug 17 06:50 .ICEauthority
-rw-r--r-- 1 ubuntu ubuntu   675 May  1 2015 .profile
-rw-r--r-- 1 root  root      0 May  1 2015 .sudo_as_admin_successful
-rw----- 1 ubuntu ubuntu   62 Aug 17 06:50 .Xauthority
-rw----- 1 ubuntu ubuntu  1576 Aug 17 06:50 .xsession-errors
-rw----- 1 ubuntu ubuntu  1442 Aug 16 04:05 .xsession-errors.old
ubuntu@ubuntu-VirtualBox:~$ ls -al | grep ^[^\d] | grep ' \.'
```

Komanda cut

- Ova komanda omogućava selekciju delova linije (pod delovima se podrazumevaju bajtovi, karakteri i polja tj. kolone)
- Veoma je zgodna za selekciju kolona ako je tekst organizovan u kolone
- Difolt delimiter za kolone je TAB karakter, ali može se definisati i drugačiji, ako je drugačiji delimiter korišćen za kolone (na primer, space karakter, dvotačka, zarez,...)
- Prilikom selekcije definiše se lista delova, pri čemu se članovi liste razdvajaju zarezima
- U okviru liste se mogu definisati i opsezi (n - označava selekciju od n -tog dela do kraja linije, $-n$ označava od početka linije do n -tog dela, $n-m$ označava od n -tog do m -tog dela linije)

Komanda cut - opcije

- -b - selekcija bajtova
- -c - selekcija karaktera
- -f - selekcija polja (kolona)
- -d - definicija delimitera
- --complement - komplement selekcije
- -s - ne prikazuju se linije koje ne sadrže delimiter
- U slučaju opcija -b, -c i -f se navodi lista selekcije (npr. u slučaju karaktera navode se redni brojevi karaktera koji se žele selektovati odvojeni zarezom, pri čemu umesto rednog broja može da se navede i opseg)
- U slučaju opcije -d se navodi definicija delimitera (ako je delimiter space karakter on mora da ide pod navodnike)

Komanda cut - primeri

```
ubuntu@ubuntu-VirtualBox:~$ cat kolone.txt
jedan dva tri
cetiri pet sest
sedam osam devet
ubuntu@ubuntu-VirtualBox:~$ cut -c1,3,5-6,8 kolone.txt
jdn v
ctrip
sdm s
ubuntu@ubuntu-VirtualBox:~$ cut --complement -c1,3,5-6,8 kolone.txt
eada tri
ei et sest
eaoam devet
ubuntu@ubuntu-VirtualBox:~$ cut -c3- kolone.txt
dan dva tri
tiri pet sest
dam osam devet
ubuntu@ubuntu-VirtualBox:~$ cut -c-3 kolone.txt
jed
cet
sed
ubuntu@ubuntu-VirtualBox:~$ cut -d" " -f1-2 kolone.txt
jedan dva
cetiri pet
sedam osam
ubuntu@ubuntu-VirtualBox:~$ cut -d" " -f1,3 kolone.txt
jedan tri
cetiri sest
sedam devet
ubuntu@ubuntu-VirtualBox:~$
```

Komanda tr

- Ova komanda omogućava translaciju karaktera tj. zamenu određenog tipa karaktera za neku drugu vrednost
- Prvo se navodi šta se menja (skup 1), a potom čime se menja (skup 2)
- Opcija -d omogućava brisanje određenih karaktera
- Opcija -c invertuje zadati skup
- Ulaz komande je standardni ulaz (stdin), a izlaz je standardni izlaz (stdout) - naravno, može se koristiti redirekcija, na primer, da bi se izlaz upisao u neki fajl, a ne na terminal (**tee** komanda u sledećoj fazi pajpa može da uradi ispis i na terminal i u fajl)

Komanda tr - primeri

```
ubuntu@ubuntu-VirtualBox:~/Documents$ cat doc.txt
jedan
dva
tri
cetiri
ubuntu@ubuntu-VirtualBox:~/Documents$ cat doc.txt | tr 'a' 'x'
jedxn
dvx
tri
cetiri
ubuntu@ubuntu-VirtualBox:~/Documents$ cat doc.txt | tr 'a-z' 'A-Z'
JEDAN
DVA
TRI
CETIRI
ubuntu@ubuntu-VirtualBox:~/Documents$ cat doc.txt | tr 'a-f' 'A-F'
jEDAn
DvA
tri
CEtiri
ubuntu@ubuntu-VirtualBox:~/Documents$ cat doc.txt | tr -d 'a'
jedn
dv
tri
cetiri
ubuntu@ubuntu-VirtualBox:~/Documents$ cat doc.txt | tr -d 'ae'
jdn
dv
tri
ctiri
ubuntu@ubuntu-VirtualBox:~/Documents$ cat doc.txt | tr [:lower:] [:upper:]
JEDAN
DVA
TRI
CETIRI
```

Komanda wc

- Ova komanda daje ispis broja linija, reči i karaktera (tim redom)
- Ulaz može biti stdin, ali i fajl (može i više fajlova da se navede)
- Opcija -l ispisuje broj linija
- Opcija -m ispisuje broj karaktera
- Opcija -w ispisuje broj reči
- Opcija -L ispisuje dužinu najduže linije

Komanda wc - primeri

U slučaju više fajlova, navodi se suma karaktera, reči i linija svih fajlova gledano zbirno. U slučaju dužine najduže linije ispisuje se max sa stanovišta svih fajlova.

```
ubuntu@ubuntu-VirtualBox:~/Documents$ cat primer4.txt
jedna rec
preradjen proizvod
redni broj
dobre namere
gore dole
ne prestajati
pozvati redare
do re mi
ubuntu@ubuntu-VirtualBox:~/Documents$ wc primer4.txt
  8  17 101 primer4.txt
ubuntu@ubuntu-VirtualBox:~/Documents$ wc -L primer4.txt
18 primer4.txt
ubuntu@ubuntu-VirtualBox:~/Documents$ wc -w primer4.txt
17 primer4.txt
ubuntu@ubuntu-VirtualBox:~/Documents$ wc -mw primer4.txt
17 101 primer4.txt
ubuntu@ubuntu-VirtualBox:~/Documents$ wc -mw primer4.txt primer3.txt
17 101 primer4.txt
 7  35 primer3.txt
24 136 total
ubuntu@ubuntu-VirtualBox:~/Documents$ wc -mwL primer4.txt primer3.txt
17 101  18 primer4.txt
 7  35   4 primer3.txt
24 136  18 total
```

Komanda sort

- Ova komanda vrši sortiranje linija iz navedenih fajlova ili stdin ulaza i vrši ispis rezultata na stdout izlaz (ili ako se radi redirekcija u fajl)
- Postoji velik broj opcija za sortiranje
- -r - vrši obrtanje rezultata sortiranja
- -b - ignoriše eventualne space karaktere na početku linije
- -d - sortiranje se radi samo na osnovu space karaktera i alfanumeričkih znakova
- -n - numeričko sortiranje
- -R - random sortiranje
- -i - uzimaju se u obzir samo karakteri koji se mogu štampati (*printable* karakteri)
- -f - ignoriše se *case* tj. veličina kod slova (*case insensitive*) - sva slova se tretiraju kao velika slova

Komanda sort - primeri

```
ubuntu@ubuntu-VirtualBox:~/Documents$ sort brojevi.txt
105
11
23
2845
76
ubuntu@ubuntu-VirtualBox:~/Documents$ sort -n brojevi.txt
11
23
76
105
2845
ubuntu@ubuntu-VirtualBox:~/Documents$ sort doc.txt
cetiri
dva
jedan
tri
ubuntu@ubuntu-VirtualBox:~/Documents$ sort -r doc.txt
tri
jedan
dva
cetiri
ubuntu@ubuntu-VirtualBox:~/Documents$ sort -R doc.txt
cetiri
dva
tri
jedan
```


Komanda sort

Opcija -k omogućava sortiranje po kolonama. Uz k se navodi ključ po kome se vrši sortiranje, u ovom slučaju redni broj kolone. Ako se želi sortiranje vršiti po više kolona, onda se mora zatvoriti opseg kolona. Kada nije zatvoren opseg uzimaju se redom sve kolone do kraja.

```
ubuntu@ubuntu-VirtualBox:~/Documents$ sort brojevi.txt doc.txt
105
11
23
2845
76
cetiri
dva
jedan
tri
ubuntu@ubuntu-VirtualBox:~/Documents$ sort -k2 kolone.txt
jedan cetiri sest
jedan dva osam
jedan dva tri
sedam osam devet
cetiri pet sest
ubuntu@ubuntu-VirtualBox:~/Documents$ sort -k1 kolone.txt
cetiri pet sest
jedan cetiri sest
jedan dva osam
jedan dva tri
sedam osam devet
ubuntu@ubuntu-VirtualBox:~/Documents$ sort -k1,1 -k3,3 kolone.txt
cetiri pet sest
jedan dva osam
jedan cetiri sest
jedan dva tri
sedam osam devet
```

Komanda sort – dodatna napomena

- U slučaju da je delimiter između kolona različit od space ili tab karaktera može se definisati preko opcije -t
- Na primer -t “;” ako je ; delimiter
- Dodatno, moguće je da različite kolone imaju potrebu za različitim tipovima sortiranja, na primer, jedna kolona zahteva numeričko sortiranja, a druga alfanumeričko – tada se na kraju opsega kolona definiše i kriterijum sortiranja

Komanda sort – dodatna napomena

```
ubuntu@ubuntu2004:~/term5$ cat kolone3
2014;0256;8
2014;123;7
2015;11;10
2013;0223;7
2012;0435;9
2016;320;6
2015;111;10
ubuntu@ubuntu2004:~/term5$ sort -n -k2,2 kolone3
2012;0435;9
2013;0223;7
2014;0256;8
2014;123;7
2015;11;10
2015;111;10
2016;320;6
ubuntu@ubuntu2004:~/term5$ sort -t ";" -n -k2,2 kolone3
2015;11;10
2015;111;10
2014;123;7
2013;0223;7
2014;0256;8
2016;320;6
2012;0435;9
```

Komanda sort – dodatna napomena

```
ubuntu@ubuntu2004:~/term5$ cat kolone1
Studenti
Jovan Jovic 2015 0264
Mile Milic 2016 0357
Jovana Panic 2015 0145
Jelena Maric 2016 0023
Petar Popovic 2016 0043
Igor Kovic 2014 0434
Rade Dabic 2015 0333
ubuntu@ubuntu2004:~/term5$ sort -k3,3n -k2,2 kolone1
Studenti
Igor Kovic 2014 0434
Rade Dabic 2015 0333
Jovan Jovic 2015 0264
Jovana Panic 2015 0145
Jelena Maric 2016 0023
Mile Milic 2016 0357
Petar Popovic 2016 0043
```

```
ubuntu@ubuntu2004:~/term5$ sort -k3,3n -k2,2r kolone1
Studenti
Igor Kovic 2014 0434
Jovana Panic 2015 0145
Jovan Jovic 2015 0264
Rade Dabic 2015 0333
Petar Popovic 2016 0043
Mile Milic 2016 0357
Jelena Maric 2016 0023
```

Komanda uniq

- Ova komanda vrši eliminisanje duplikata linija iz sortiranog ulaza (stdin ulaz)
- Ulaz mora biti sortiran jer **uniq** komanda upoređuje samo susedne linije, ako su duplikati razdvojeni onda **uniq** neće prepoznati duplikate
- Opcija -i ignoriše veličinu slova
- Opcija -s omogućava da se preskoči prvih n karaktera u liniji prilikom poređenja
- Opcija -d štampa samo linije koje imaju duplikat
- Opcija -c ispisuje pored linije broj duplikata (uključujući i original)

Komanda uniq - primeri

```
ubuntu@ubuntu-VirtualBox:~/Documents$ sort dupli.txt | uniq
cetiri
dva
jedan
Jedan
pet
tri
ubuntu@ubuntu-VirtualBox:~/Documents$ sort dupli.txt | uniq -i
cetiri
dva
jedan
pet
tri
ubuntu@ubuntu-VirtualBox:~/Documents$ sort dupli.txt | uniq -c
      1 cetiri
      2 dva
      2 jedan
      1 Jedan
      1 pet
      2 tri
ubuntu@ubuntu-VirtualBox:~/Documents$ sort dupli.txt | uniq -ci
      1 cetiri
      2 dva
      3 jedan
      1 pet
      2 tri
```

Komanda uniq - primeri

Primetiti da **cat** komanda nije poređala duplikate pa **uniq** prikazuje iste reči više puta. Zato je zgodno uraditi **sort** pre **uniq**.

Komanda **sort** ima opciju **-u** koja takođe daje prikaz bez duplikata. Ali, kombinovanje opcija **-ui** ne eliminiše prikaz reči Jedan

```
ubuntu@ubuntu-VirtualBox:~/Documents$ cat dupli.txt | uniq -i
jedan
dva
tri
cetiri
dva
pet
tri
ubuntu@ubuntu-VirtualBox:~/Documents$ sort dupli.txt | uniq -d
dva
jedan
tri
ubuntu@ubuntu-VirtualBox:~/Documents$ sort -u dupli.txt
cetiri
dva
jedan
Jedan
pet
tri
ubuntu@ubuntu-VirtualBox:~/Documents$ sort -ui dupli.txt
cetiri
dva
jedan
Jedan
pet
tri
```

Komanda comm

- Ova komanda vrši poređenje dva fajla i ispisuje rezultate poređenja u tri kolone
- Prva kolona sadrži samo linije jedinstvene fajlu 1
- Druga kolona sadrži samo linije jedinstvene fajlu 2
- Treća kolona sadrži linije koje se pojavljuju u oba fajla
- Ako se želi sprečiti prikaz neke od kolona onda se navodi njen broj kao opcija
- Na primer, -1 potiskuje kolonu 1, a -13 kolone 1 i 3

Komanda comm - primeri

```
ubuntu@ubuntu-VirtualBox:~/Documents$ comm lista1.txt lista2.txt
devet
jedan
osam
pet
sedam
sest
tri
ubuntu@ubuntu-VirtualBox:~/Documents$ comm -23 lista1.txt lista2.txt
jedan
tri
ubuntu@ubuntu-VirtualBox:~/Documents$ comm -1 lista1.txt lista2.txt
devet
osam
pet
sedam
sest
ubuntu@ubuntu-VirtualBox:~/Documents$ comm -2 lista1.txt lista2.txt
jedan
pet
sedam
tri
```

Komanda od

- Ova komanda vrši ispis bajtova umesto karaktera
- Format ispisa se može podesiti opcijama
- Na primer, opcija -x kreira heksadecimalni ispis, a opcija -i decimalni integer ispisa

```
ubuntu@ubuntu-VirtualBox:~/Documents$ cat lista2.txt
devet
osam
pet
sedam
sest
ubuntu@ubuntu-VirtualBox:~/Documents$ od lista2.txt
0000000 062544 062566 005164 071557 066541 070012 072145 071412
0000020 062145 066541 071412 071545 005164
0000032
ubuntu@ubuntu-VirtualBox:~/Documents$ od -i lista2.txt
0000000 1702258020 1936657012 1879731553 1930064997
0000020 1835099237 1936028426 2676
0000032
ubuntu@ubuntu-VirtualBox:~/Documents$ od -x lista2.txt
0000000 6564 6576 0a74 736f 6d61 700a 7465 730a
0000020 6465 6d61 730a 7365 0a74
0000032
```

date

- Komanda za prikaz i postavljanje vremena i datuma
- Ako se navede bez opcija onda daje prikaz trenutnog datuma i vremena (u nastavku će biti navođen samo termin vreme)
- Pored opcija moguće je kontrolisati i format ispisa
- Lista formata se može naći u man za **date** komandu
- Opcija *-d opisni_string* daje prikaz vremena za opisni string
- Opisni string može biti next Monday, 2 days ago i sl.
- Opcija *-f fajl* vrši za svaku liniju navedenog fajla primenu opcije *-d*
- Opcija *-r fajl* daje vreme modifikacije navedenog fajla
- Opcija *-u* daje prikaz UTC (*Coordinated Universal Time*) vremena

date

- Opcija `-s novo_vreme` omogućava podešavanje vremena na `novo_vreme`
- Napomena: u novijim distribucijama ova opcija neće uraditi promenu vremena, tada se za promenu vremena mora koristiti `timedatectl` komanda
- Pošto se menja sistemsko vreme, kada se želi promeniti vreme mora se koristiti administratorski nalog
- Format vremena se navodi na kraju i on utiče na prikaz informacije o vremenu
- Neke od opcija u formatu su: `%H` (čas 0-23), `%M` (minut), `%m` (mesec numerički prikaz), `%a` i `%A` (naziv dana, skraćen i pun), `%b` i `%B` (naziv meseca, skraćen i pun), `%d` (dan), `%Y` i `%y` (godina 4, odnosno 2 cifre) `%D` (ekvivalent `%m/%d/%y`), `%F` (ekvivalent `%Y-%m-%d`), `%S` (sekunda), `%R` (sat i minut 24h format), `%r` (sat i minut 12h format),...

date - primeri

```
ubuntu@ubuntu-VirtualBox:~$ date
Sun Aug 21 12:55:10 EDT 2016
ubuntu@ubuntu-VirtualBox:~$ date -ud 'next Friday'
Fri Aug 26 00:00:00 UTC 2016
ubuntu@ubuntu-VirtualBox:~$ date -d '100 hours ago'
Wed Aug 17 08:55:57 EDT 2016
ubuntu@ubuntu-VirtualBox:~$ date -d '100 hours ago' '+ %D %R'
08/17/16 08:57
ubuntu@ubuntu-VirtualBox:~$ date -d '100 hours ago' '+ %d.%m.%Y %R'
17.08.2016 08:58
```

U trećoj liniji je 12 protumačen kao sat. Mogu se koristiti i fraze kao next month, last Thursday i sl.

```
ubuntu@ubuntu-VirtualBox:~$ cat datumi.txt
01/01/15 10:38
Sep 12 2014
Apr 23 12
08/12/12
02-03-11
ubuntu@ubuntu-VirtualBox:~$ date -f datumi.txt
Thu Jan  1 10:38:00 EST 2015
Fri Sep 12 00:00:00 EDT 2014
Sat Apr 23 12:00:00 EDT 2016
Sun Aug 12 00:00:00 EDT 2012
Mon Mar 11 00:00:00 EST 2002
```

timedatectl

- U novijim distribucijama ova komanda se koristi za podešavanje vremena, kao i za prikaz trenutnih podešavanja
- Potrebne su administratorske privilegije za podešavanja
- **timedatectl set-time *vreme*** vrši podešavanje vremena

Primer komande za promenu vremena je dat na kraju. U ovom slučaju komanda neće biti uspešno izvršena jer je aktivirana NTP sinhronizacija.

```
ubuntu@ubuntu-VirtualBox:~$ timedatectl
      Local time: Sun 2016-08-21 13:10:15 EDT
      Universal time: Sun 2016-08-21 17:10:15 UTC
      RTC time: Sun 2016-08-21 17:10:11
      Time zone: America/New_York (EDT, -0400)
      NTP enabled: yes
      NTP synchronized: yes
      RTC in local TZ: no
      DST active: yes
      Last DST change: DST began at
                          Sun 2016-03-13 01:59:59 EST
                          Sun 2016-03-13 03:00:00 EDT
      Next DST change: DST ends (the clock jumps one hour backwards) at
                          Sun 2016-11-06 01:59:59 EDT
                          Sun 2016-11-06 01:00:00 EST
ubuntu@ubuntu-VirtualBox:~$ timedatectl set-time '2016-08-22'
```

cal i ncal

- Komande za prikaz kalendara
- Po defaultu prikazuje se kalendar za tekući mesec sa obeleženim trenutnim danom
- Opcija *-m mesec* prikazuje kalendar navedenog meseca tekuće godine (mesec se navodi imenom ili brojem)
- Opcija *-y godina* prikazuje kalendar čitave godine koja je navedena
- Opcija *-e* i *-o* prikazuju datum Uskrsa (*-o* opcija je za pravoslavni Uskrs) - ne radi za **cal** komandu
- Opcija *-d yyyy-mm* daje prikaz kalendara za navedeni mesec i godinu
- Opcija *-A* i *-B* prikazuju kalendare za mesece iza i ispred tekućeg meseca - zgodno za definisanje opsega meseci (*A* je od after, a *B* je od before)

cal i ncal primeri

```
ubuntu@ubuntu-VirtualBox:~$ cal
      August 2016
Su Mo Tu We Th Fr Sa
      1  2  3  4  5  6
  7  8  9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31

ubuntu@ubuntu-VirtualBox:~$ cal -d 2014-04
      April 2014
Su Mo Tu We Th Fr Sa
      1  2  3  4  5
  6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30
```

```
ubuntu@ubuntu-VirtualBox:~$ cal -m April
      April 2016
Su Mo Tu We Th Fr Sa
              1  2
  3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30

ubuntu@ubuntu-VirtualBox:~$ cal -m 4
      April 2016
Su Mo Tu We Th Fr Sa
              1  2
  3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
```


cal i ncal primeri

```
ubuntu@ubuntu-VirtualBox:~$ cal
      August 2016
Su     7 14 21 28
Mo    1  8 15 22 29
Tu    2  9 16 23 30
We    3 10 17 24 31
Th    4 11 18 25
Fr    5 12 19 26
Sa    6 13 20 27
ubuntu@ubuntu-VirtualBox:~$ cal -m 4
      April 2016
Su     3 10 17 24
Mo     4 11 18 25
Tu     5 12 19 26
We     6 13 20 27
Th     7 14 21 28
Fr     1  8 15 22 29
Sa     2  9 16 23 30
ubuntu@ubuntu-VirtualBox:~$ cal -o
May 1 2016
ubuntu@ubuntu-VirtualBox:~$ cal -o -y 2014
April 20 2014
```

```
ubuntu@ubuntu-VirtualBox:~$ cal -A1 -B1
                2016
           July                August                September
Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa
                1 2           1 2 3 4 5 6           1 2 3
3 4 5 6 7 8 9 7 8 9 10 11 12 13 4 5 6 7 8 9 10
10 11 12 13 14 15 16 14 15 16 17 18 19 20 11 12 13 14 15 16 17
17 18 19 20 21 22 23 21 22 23 24 25 26 27 18 19 20 21 22 23 24
24 25 26 27 28 29 30 28 29 30 31 25 26 27 28 29 30
31
ubuntu@ubuntu-VirtualBox:~$ cal -A1 -B1 -m 3
           February 2016           March 2016           April 2016
Su     7 14 21 28           6 13 20 27           3 10 17 24
Mo    1  8 15 22 29           7 14 21 28           4 11 18 25
Tu    2  9 16 23           1  8 15 22 29           5 12 19 26
We    3 10 17 24           2  9 16 23 30           6 13 20 27
Th    4 11 18 25           3 10 17 24 31           7 14 21 28
Fr    5 12 19 26           4 11 18 25           1  8 15 22 29
Sa    6 13 20 27           5 12 19 26           2  9 16 23 30
```

sleep

- Komanda kašnjenja za određeno vreme
- Može biti zgodna u skriptama za odlaganje izvršenja neke komande
- Iza komande se navodi vreme kašnjenja
- Ako se ne navedu jedinice onda su u pitanju sekunde
- Oznake sufiksa (tj. jedinica vremena koje se pišu iza numeričke vrednosti) su s (sekunde), m (minute), h (časovi) i d (dani)
- Ako se navede više argumenata njihov zbir se računa kao kašnjenje (zgodno za kombinovanje različitih jedinica)
- Kao numerička vrednost se može staviti i racionalan broj (*float*)

```
ubuntu@ubuntu-VirtualBox:~$ sleep 2
ubuntu@ubuntu-VirtualBox:~$ sleep 2s
ubuntu@ubuntu-VirtualBox:~$ sleep 2s 1s
ubuntu@ubuntu-VirtualBox:~$ sleep 2s 3s
ubuntu@ubuntu-VirtualBox:~$ sleep 2s 1
ubuntu@ubuntu-VirtualBox:~$ sleep 2s 1.3
```

time

- Komanda kojom se pokreće neki drugi program i potom se ispisuje statistika izvršavanja pokrenutog programa (dužina izvršavanja, opterećenost procesora i sl.)
- Opcije -o i -a omogućavaju ispis statistike u fajl umesto na terminal (opcija -o će da prebriše fajl, dok će -a da dopiše sadržaj u fajl)
- Prilikom ispisa prikazuje se *real* (vreme izvršenja), *user* (vreme izvršenja u user modu), *sys* (vreme izvršenja u kernel modu) rezultati

time primeri

```
ubuntu@ubuntu-VirtualBox:~$ time ls
datumi.txt Documents examples.desktop Music Public Videos
Desktop Downloads ispis.txt Pictures Templates

real    0m0.023s
user    0m0.000s
sys     0m0.000s
ubuntu@ubuntu-VirtualBox:~$ time sleep 1

real    0m1.008s
user    0m0.000s
sys     0m0.000s
ubuntu@ubuntu-VirtualBox:~$ time sudo updatedb
[sudo] password for ubuntu:

real    0m2.745s
user    0m0.016s
sys     0m0.028s
ubuntu@ubuntu-VirtualBox:~$ time cat datumi.txt
01/01/15 10:38
Sep 12 2014
Apr 23 12
08/12/12
02-03-11
next Thursday

real    0m0.010s
user    0m0.000s
sys     0m0.004s
```

gzip, gunzip i zcat

- Komanda **gzip** služi za kompresovanje fajlova, a **gunzip** za dekompresiju komprimovanih fajlova (koji su komprimovani gzip alatom)
- **zcat** komanda omogućava da se pročita sadržaj fajla iz gzip komprimovanog fajla (u suštini **cat** iz komprimovanog fajla)
- Po defaultu kad se zipuje fajl on se automatski i briše (ostaje samo arhiviran fajl)
- Isto važi i kad se vrši dekompresija (arhiva se briše)
- Opcija **-k** u oba slučaja čuva fajlove, odnosno arhive od brisanja
- Opcija **-r** omogućava rekurziju u okviru kompresije, tj. ako se naiđe na direktorijum onda se njegov sadržaj tj. fajlovi komprimuju
- Sam **gzip** ne komprimuje folder, to radi **tar** alat, pa se zato često koriste u paru **gzip** i **tar**

gzip, gunzip i zcat - primeri

```
ubuntu@ubuntu-VirtualBox:~/Documents/test$ ls
fajl1 fajl2
ubuntu@ubuntu-VirtualBox:~/Documents/test$ gzip fajl1 fajl2
ubuntu@ubuntu-VirtualBox:~/Documents/test$ ls
fajl1.gz fajl2.gz
ubuntu@ubuntu-VirtualBox:~/Documents/test$ zcat fajl1.gz
./xx.txt
./dir11/nn.txt
./dir11/dir22/ccc.txt
./dir11/dir21/aa.txt
ubuntu@ubuntu-VirtualBox:~/Documents/test$ gunzip fajl1.gz fajl2.gz
ubuntu@ubuntu-VirtualBox:~/Documents/test$ ls
fajl1 fajl2
ubuntu@ubuntu-VirtualBox:~/Documents/test$ cat fajl1
./xx.txt
./dir11/nn.txt
./dir11/dir22/ccc.txt
./dir11/dir21/aa.txt
```

Postoji i **zmore** komanda koja je slična **zcat** komandi. Razlika je što **zmore** omogućava ispis stranu po stranu.

gzip, gunzip i zcat - primeri

```
ubuntu@ubuntu-VirtualBox:~/Documents/test$ ls
fajl1 fajl2
ubuntu@ubuntu-VirtualBox:~/Documents/test$ gzip -k fajl1 fajl2
ubuntu@ubuntu-VirtualBox:~/Documents/test$ ls
fajl1 fajl1.gz fajl2 fajl2.gz
ubuntu@ubuntu-VirtualBox:~/Documents/test$ rm fajl1
ubuntu@ubuntu-VirtualBox:~/Documents/test$ ls
fajl1.gz fajl2 fajl2.gz
ubuntu@ubuntu-VirtualBox:~/Documents/test$ gunzip -k fajl1.gz
ubuntu@ubuntu-VirtualBox:~/Documents/test$ ls
fajl1 fajl1.gz fajl2 fajl2.gz
```

gzip, gunzip i zcat - primeri

```
ubuntu@ubuntu-VirtualBox:~/Documents/test$ ls
fajl1 fajl2 test2
ubuntu@ubuntu-VirtualBox:~/Documents/test$ cd ..
ubuntu@ubuntu-VirtualBox:~/Documents$ gzip test
gzip: test is a directory -- ignored
ubuntu@ubuntu-VirtualBox:~/Documents$ gzip -r test
ubuntu@ubuntu-VirtualBox:~/Documents$ cd test/
ubuntu@ubuntu-VirtualBox:~/Documents/test$ ls
fajl1.gz fajl2.gz test2
ubuntu@ubuntu-VirtualBox:~/Documents/test$ cd test2/
ubuntu@ubuntu-VirtualBox:~/Documents/test/test2$ ls
fajl3.gz
ubuntu@ubuntu-VirtualBox:~/Documents/test/test2$ cd ../..
ubuntu@ubuntu-VirtualBox:~/Documents$ gunzip -r test
ubuntu@ubuntu-VirtualBox:~/Documents$ cd test/
ubuntu@ubuntu-VirtualBox:~/Documents/test$ ls
fajl1 fajl2 test2
ubuntu@ubuntu-VirtualBox:~/Documents/test$ cd test2/
ubuntu@ubuntu-VirtualBox:~/Documents/test/test2$ ls
fajl3
```


gzip, gunzip i zcat - primeri

```
ubuntu@ubuntu-VirtualBox:~/Documents$ tar -zcvf test_arhiva.tar.gz test
test/
test/test2/
test/test2/fajl3.gz
test/fajl1
test/fajl2
ubuntu@ubuntu-VirtualBox:~/Documents$ ls
brojevi.txt  dupli.txt  kolone.txt  primer2.txt  primer.txt  test_arhiva.tar.gz
dir0        fajl1     lista1.txt  primer3.txt  reci.txt
doc.txt     fajl2     lista2.txt  primer4.txt  test
ubuntu@ubuntu-VirtualBox:~/Documents$ rm -r test
ubuntu@ubuntu-VirtualBox:~/Documents$ ls
brojevi.txt  dupli.txt  kolone.txt  primer2.txt  primer.txt
dir0        fajl1     lista1.txt  primer3.txt  reci.txt
doc.txt     fajl2     lista2.txt  primer4.txt  test_arhiva.tar.gz
ubuntu@ubuntu-VirtualBox:~/Documents$ tar -zxvf test_arhiva.tar.gz
test/
test/test2/
test/test2/fajl3.gz
test/fajl1
test/fajl2
ubuntu@ubuntu-VirtualBox:~/Documents$ ls
brojevi.txt  dupli.txt  kolone.txt  primer2.txt  primer.txt  test_arhiva.tar.gz
dir0        fajl1     lista1.txt  primer3.txt  reci.txt
doc.txt     fajl2     lista2.txt  primer4.txt  test
ubuntu@ubuntu-VirtualBox:~/Documents$ cd test
ubuntu@ubuntu-VirtualBox:~/Documents/test$ ls
fajl1  fajl2  test2
```

Ovde je prikazana upotreba **tar** komande za kompresiju direktorijuma. Opcija **-c** govori da se kreira arhiva, opcija **-z** da se koristi gzip alat, opcija **-f** da se arhiva smesti u fajl, a opcija **-v** da se ispiše sadržaj kompresije. Opcija **-x** je za proces dekompresije. Sve opcije **tar** komande se mogu videti u man za **tar** komandu.

bzip2, bunzip2 i bzipcat

- Identično kao i gzip alati (postoji i **bzmore**)
- Razlika je u algoritmu kompresije
- **bzip2** postiže bolje rezultate kompresije, ali je nešto sporiji od **gzip**
- Opcija -k postoji i ovde, ali ne postoji opcija -r (moraju da se navedu fajlovi)
- U slučaju upotrebe kao deo **tar** komande umesto opcije -z navodi se opcija -j

Komanda sed

- Komanda **sed** se koristi za editovanje strima u pajpu, ali može se koristiti i za editovanje sadržaja fajla
- Najčešće se koristi za *replace* funkciju, tj. da se jedan string zameni nekom drugom vrednošću
- Međutim, ova komanda ima široku upotrebnu vrednost
- <https://www.gnu.org/software/sed/manual/sed.txt> sadrži zvaničnu dokumentaciju, a link <http://www.grymoire.com/Unix/Sed.html> sadrži malo lepša objašnjenja mogućnosti **sed** komande

Substitution

- Kada se koristi *replace* funkcionalnost, tada se koristi **s** (*substitution*) funkcija **sed** komande u formatu:
 - **sed s/regexp/nova_vrednost/**
- Karakter / predstavlja delimiter i umesto njega se može koristiti, na primer, : ili _ radi bolje preglednosti u nekim situacijama
- Regularni izraz predstavlja regexp
- Zamena (*replace*) se odvija po defaultu samo na prvom nađenom poklapanju u liniji (**sed** radi na nivou linija)
- Ako se želi zamena uraditi na kompletnoj liniji, koristi se opcija **g**, a konkretna brojna vrednost radi zamenu na navedenom mestu pojavljivanja regularnog izraza
- Ako se navede broj ispred **g** onda se zamena radi na čitavoj liniji počev od navedenog pojavljivanja regularnog izraza

Substitution

- Opcija **i** radi case insensitive poklapanje
- Karakter **&** se može iskoristiti kao referenca na zadati regularni izraz (korisno ako, na primer, želimo da umnožimo traženi string)
- Opcija **w** navodi fajl u koji će biti upisane urađene modifikacije (ako ima više opcija ova se navodi na kraju) - zapisuju se samo modifikovane linije
- Napomena: opcije koje su navedene bez - se odnose na opcije u **s** funkciji i one se navode na kraju **s** funkcije
- Opcija **-r** mora da se koristi ako želimo da se regularni izraz tumači po ERE sintaksi
- Opcija **-e** se koristi kada želimo više **sed** komandi da vežemo (u suštini može i **pajp** da se uradi kao alternativa)

Substitution primeri

U trećem primeru je naveden redni broj pojavljivanja poklapanja na kom treba uraditi zamenu. Primetiti upotrebu & kao referencu na regularni izraz u pretposlednja tri primera. Poslednji primer pokazuje kako se pomoću escape karaktera, karakter & tumači kao običan.

```
ubuntu@ubuntu-VirtualBox:~$ echo jedan dva jedan | sed 's/je/aa/'
aadan dva jedan
ubuntu@ubuntu-VirtualBox:~$ echo jedan dva jedan | sed 's/je/aa/g'
aadan dva aadan
ubuntu@ubuntu-VirtualBox:~$ echo jedan dva jedan | sed 's/je/aa/2'
jedan dva aadan
ubuntu@ubuntu-VirtualBox:~$ echo Jedan dva jedan | sed 's/je/aa/'
Jedan dva aadan
ubuntu@ubuntu-VirtualBox:~$ echo Jedan dva jedan | sed 's/je/aa/i'
aadan dva jedan
ubuntu@ubuntu-VirtualBox:~$ echo Jedan dva jedan | sed 's/je/aa/ig'
aadan dva aadan
ubuntu@ubuntu-VirtualBox:~$ echo jedan dva jedan | sed 's/je/##/'
jejedan dva jedan
ubuntu@ubuntu-VirtualBox:~$ echo jedan dva jedan | sed 's/je/##/g'
jejedan dva jejedan
ubuntu@ubuntu-VirtualBox:~$ echo jedan dva jedan | sed 's/je/&t/g'
jetdan dva jetdan
ubuntu@ubuntu-VirtualBox:~$ echo jedan dva jedan | sed 's/je/\\&t/g'
&tdan dva &tdan
```

Substitution primeri

Komanda **sed** može da prihvati i fajl kao ulaz. Ako se koristi opcija **w**, onda se modifikovane linije upisuju u fajl koji je naveden nakon **w** opcije.

```
ubuntu@ubuntu-VirtualBox:~$ echo jedan dva | sed -e 's/je/aa/' -e 's/v/x/'
aadan dxa
ubuntu@ubuntu-VirtualBox:~$ sed 's/a/x/' Documents/lista1.txt
devet
osxm
pet
sedxm
sest
ubuntu@ubuntu-VirtualBox:~$ sed 's/a/x/w ispis.txt' Documents/lista1.txt
devet
osxm
pet
sedxm
sest
ubuntu@ubuntu-VirtualBox:~$ cat ispis.txt
osxm
sedxm
```

Označavanje regularnog izraza

- Regularni izrazi se mogu grupisati i indeksirati
- Na taj način je moguće pristupiti regularnim izrazima i postaviti ih u modifikovanoj liniji u proizvoljnom redosledu
- Mogu se na ovaj način i umnožiti indeksirani regularni izrazi
- Mogu se izdvojiti i samo delovi od interesa u regularnom izrazu
- Indeks se navodi kao $\backslash n$ gde je n konkretan broj (indeks regularnog izraza)

Označavanje primeri

```
ubuntu@ubuntu2004:~$ echo 2011 0125 ocena 10 | sed -r 's/([[[:digit:]]{1,}) ([[[:digit:]]{1,})/\2 \1/'
0125 2011 ocena 10
ubuntu@ubuntu2004:~$ echo aabb | sed -r 's/^(.{2})(.{2})/\2\1/'
bbaa
ubuntu@ubuntu2004:~$ echo aabbcc | sed -r 's/^(.{2})(.{2})/\2\1/'
bbaacc
ubuntu@ubuntu2004:~$ echo aabbcc | sed -r 's/^(.{2})(.{2})(.{2})/\3\2\1/'
ccbbaa
ubuntu@ubuntu2004:~$ echo aabbcc | sed -r 's/^(.{2})(.{2})(.{2})/\3\2\3\1/'
ccbcca
ubuntu@ubuntu2004:~$ echo aabbcc | sed -r 's/^(.{2}).{2}(.{2})/\1\2/'
aacc
ubuntu@ubuntu2004:~$ echo aabbcc | sed -r 's/^(.{2}).{2}(.{2})/\2\1/'
ccaa
```

Opcije p i n

- Opcija p se koristi za dupliranje linije koja je modifikovana
- Opcija -n se koristi za potiskivanje ispisa sem ako se ne naide na zahtev za ispisom (opcija p)
- Kombinovanjem p i n opcije se može dobiti funkcionalnost grep komande

Opcije p i n - primeri

```
ubuntu@ubuntu2004:~$ cat ekv
123
456
a
cdf
saq
aa22 jfjf 5
s34
cvfn_+
qwerty
1+2
ubuntu@ubuntu2004:~$ cat ekv | sed -r 's/[+]/&/p'
123
456
a
cdf
saq
aa22 jfjf 5
s34
cvfn_+
cvfn_+
qwerty
1+2
1+2
```

Opcije p i n - primeri

```
ubuntu@ubuntu2004:~$ cat ekv | sed -n -r 's/[+]/&/p'  
cvfn_+  
1+2  
ubuntu@ubuntu2004:~$ cat ekv | grep -E '[+]'  
cvfn_+  
1+2  
ubuntu@ubuntu2004:~$ cat ekv | sed -n -r 's/[+]/&&/p'  
cvfn_++  
1++2  
ubuntu@ubuntu2004:~$ cat ekv | sed -n -r 's/[+]/&-/p'  
cvfn_+-  
1+-2  
ubuntu@ubuntu2004:~$ cat ekv | sed -n -r 's/[+]/-/p'  
cvfn_-  
1-2  
ubuntu@ubuntu2004:~$ cat ekv | sed -n -r 's/[+]/-/'  
ubuntu@ubuntu2004:~$
```

Opcije p i n - primeri

```
ubuntu@ubuntu2004:~$ cat ekv | sed -r -n ' /[+]/ p'  
cvfn_+  
1+2  
ubuntu@ubuntu2004:~$ cat ekv | sed -r -n ' /[+]/ !p'  
123  
456  
a  
cdf  
saq  
aa22 jfjf 5  
s34  
qwerty
```

Zadavanje sed komandi preko fajla

- Same sed komande poput substitucije se mogu navesti i preko fajla
- Naročito pogodno ako ima više sed komandi koje treba uvezati
- Isto pogodno ako se koristi isti skup sed komandi radi smanjenja ponovnog pisanja istog sadržaja komande
- Opcijom -f se poziva fajl iz kog potom sed komanda čita šta treba da uradi
- U fajlu svaka komanda mora biti u zasebnoj liniji
- Naravno, redosled komandi može biti bitan i uticati na konačni rezultat

Primeri

```
ubuntu@ubuntu2004:~$ cat sedkom
s/a/A/g
s/e/E/g
s/[+]/&&&/
/[[:digit:]]/d
ubuntu@ubuntu2004:~$ cat ekv | sed -f sedkom
A
cdf
sAq
cvfn_+++
qwerty
```

Substitution na zadatoj liniji

- Omogućeno je i da se specificira na kojoj liniji ili opsegu linija se želi raditi substitucija
- To se radi tako što se navede broj linije ili opseg linija ispred same **s** komande
- Opseg se navodi u formatu **a,b** gde je **a** redni broj početne linije opsega, a **b** redni broj poslednje linije opsega
- Ako se za kraj opsega stavi \$ onda je poslednja linija editovanog teksta ujedno i kraj opsega
- Umesto rednog broja ili opsega, može se zadati regularni izraz – tada se s komanda radi samo na linijama koje sadrže zadati regularni izraz
- Regularni izraz se zadaje između dve /
- Može i između dva zareza ali ispred prvog treba escape karakter

Primeri

```
ubuntu@ubuntu2004:~$ cat kod
# pocetak programa
init
if start then
# glavni deo
nesto
else
prijavi gresku # ispis poruke
end
ubuntu@ubuntu2004:~$ cat kod | sed -r '1,4 s/#/###/'
### pocetak programa
init
if start then
### glavni deo
nesto
else
prijavi gresku # ispis poruke
end
ubuntu@ubuntu2004:~$ cat kod | sed -n -r '2,4 s/#/###/p'
### glavni deo
ubuntu@ubuntu2004:~$ cat kod | sed -n -r '1,4 s/#/###/p'
### pocetak programa
### glavni deo
ubuntu@ubuntu2004:~$ cat kod | sed -n -r '4 s/#/###/p'
### glavni deo
```

Primeri

```
ubuntu@ubuntu2004:~$ cat kod | sed '/#/ s/#/&&&/'
### pocetak programa
init
if start then
### glavni deo
nesto
else
prijavi gresku ### ispis poruke
end
ubuntu@ubuntu2004:~$ cat kod | sed '/^#/ s/#/&&&/'
### pocetak programa
init
if start then
### glavni deo
nesto
else
prijavi gresku # ispis poruke
end
ubuntu@ubuntu2004:~$ cat kod | sed -n '\,^#, s/#/&&&/p'
### pocetak programa
### glavni deo
```

Substitution na zadatoj liniji

- Moguće je čak navesti i opseg u vidu dva regularna izraza
- Prvi regularni izraz definiše liniju koja započinje opseg
- Drugi regularni izraz definiše liniju koja završava opseg
- Između njih se nalazi zarez

```
ubuntu@ubuntu2004:~$ cat kod
# pocetak programa
init
if start then
# glavni deo
nesto
else
prijavi gresku # ispis poruke
end
ubuntu@ubuntu2004:~$ cat kod | sed -n '/init/,/else/ s/#/###/p'
### glavni deo
ubuntu@ubuntu2004:~$ cat kod | sed -n '/^#/ ,/^e/ s/#/###/p'
### pocetak programa
### glavni deo
ubuntu@ubuntu2004:~$ cat kod | sed -n '/e/,/e/ s/#/###/p'
### pocetak programa
### glavni deo
prijavi gresku ### ispis poruke
```

Dodavanje linija

- Moguće je dodati linije ispred i iza linije koja sadrži neki regularni izraz (navodi se isto kao i kod substitucije zadatah linija)
- Isto je moguće i kompletno zameniti liniju koja sadrži neki regularni izraz
- Opcija **a** se koristi za dodavanje linije iza
- Opcija **i** za dodavanje linije ispred
- Opcija **c** za modifikovanje linije
- Moguće je dodavati i više linija ispred ili iza, kao i modifikovanu liniju zameniti sa više linija – tada se na kraju svake linije kuca \ radi prelaska u sledeći red
- Moguće je kombinovati sve tri opcije ali se tada kombinovani izraz stavlja pod vitičaste zagrade

Primeri

```
ubuntu@ubuntu2004:~$ cat kod | sed '/^#/ a\Linija iza'  
# pocetak programa  
Linija iza  
init  
if start then  
# glavni deo  
Linija iza  
nesto  
else  
prijavi gresku # ispis poruke  
end  
ubuntu@ubuntu2004:~$ cat kod | sed '/^#/ i\Linija ispred'  
Linija ispred  
# pocetak programa  
init  
if start then  
Linija ispred  
# glavni deo  
nesto  
else  
prijavi gresku # ispis poruke  
end
```

Primeri

```
ubuntu@ubuntu2004:~$ cat kod | sed '/^#/ c\Linija promenjena'  
Linija promenjena  
init  
if start then  
Linija promenjena  
nesto  
else  
prijavi gresku # ispis poruke  
end
```

```
ubuntu@ubuntu2004:~$ cat kod | sed '/^#/ i\Linija ispred 1\  
> Linija ispred 2'  
Linija ispred 1  
Linija ispred 2  
# pocetak programa  
init  
if start then  
Linija ispred 1  
Linija ispred 2  
# glavni deo  
nesto  
else  
prijavi gresku # ispis poruke  
end
```

Primeri

```
ubuntu@ubuntu2004:~$ cat kod | sed ' /^#/ {
a\MM
i\AA
c\SS
}'
AA
SS
MM
init
if start then
AA
SS
MM
nesto
else
prijavi gresku # ispis poruke
end
```

Dodavanje linija

- Umesto regularnog izraza moguće je navesti opseg linija numerički
- A može se navesti i opseg preko dva regularna izraza (za početak opsega i za kraj opsega) u istom maniru kao kod navođenja opsega kod substitucije

```
ubuntu@ubuntu2004:~$ cat kod | sed ' /^#/ ,/^#/ i\menjano'
```

```
menjano
# pocetak programa
menjano
init
menjano
if start then
menjano
# glavni deo
nesto
else
prijavi gresku # ispis poruke
end
ubuntu@ubuntu2004:~$ cat kod | sed ' /^#/ ,/^#/ c\menjano'
```

```
menjano
nesto
else
prijavi gresku # ispis poruke
end
```

```
ubuntu@ubuntu2004:~$ cat kod | sed ' 3,5 c\menjano'
```

```
# pocetak programa
init
menjano
else
prijavi gresku # ispis poruke
end
```

```
ubuntu@ubuntu2004:~$ cat kod | sed ' 3 c\menjano'
```

```
# pocetak programa
init
# glavni deo
nesto
else
prijavi gresku # ispis poruke
end
```


Dodavanje linija

```
ubuntu@ubuntu2004:~$ cat kod | sed ' /^#/ ,/^#/ i\menjano'  
menjano  
# pocetak programa  
menjano  
init  
menjano  
if start then  
menjano  
# glavni deo  
nesto  
else  
prijavi gresku # ispis poruke  
end  
ubuntu@ubuntu2004:~$ cat kod | sed ' /^#/ ,/^#/ c\menjano'  
menjano  
nesto  
else  
prijavi gresku # ispis poruke  
end
```

```
ubuntu@ubuntu2004:~$ cat kod | sed ' 3,5 c\menjano'  
# pocetak programa  
init  
menjano  
else  
prijavi gresku # ispis poruke  
end  
ubuntu@ubuntu2004:~$ cat kod | sed ' 3 c\menjano'  
# pocetak programa  
init  
menjano  
# glavni deo  
nesto  
else  
prijavi gresku # ispis poruke  
end
```

Brisanje linija

- Opcija **d** se koristi za brisanje linija
- Kriterijum za brisanje može biti regularni izraz, ali i zadati opseg linija

```
ubuntu@ubuntu2004:~$ cat ekv
123
456
a
cdf
saq
aa22 jfjf 5
s34
cvfn_+
qwerty
1+2
ubuntu@ubuntu2004:~$ cat ekv | sed ' /[[[:alpha:]]]/d'
123
456
1+2
ubuntu@ubuntu2004:~$ cat ekv | sed ' /[[[:alpha:]]]/ d'
123
456
1+2
ubuntu@ubuntu2004:~$ cat ekv | sed ' 3,$ d'
123
456
```

Brisanje - primeri

```
ubuntu@ubuntu2004:~$ cat ekv
123
456
a
cdf
saq
aa22 jfjf 5
s34
cvfn_+
qwerty
1+2
ubuntu@ubuntu2004:~$ cat ekv | sed '3,8 d'
123
456
qwerty
1+2
ubuntu@ubuntu2004:~$ cat ekv | sed '3 d'
123
456
cdf
saq
aa22 jfjf 5
s34
cvfn_+
qwerty
1+2
```

```
ubuntu@ubuntu2004:~$ cat ekv | sed -r ' /[+]/ !d'
cvfn_+
1+2
ubuntu@ubuntu2004:~$ cat ekv | sed -r ' /[+]/ d'
123
456
a
cdf
saq
aa22 jfjf 5
s34
qwerty
```

Brisanje - primeri

```
ubuntu@ubuntu-VirtualBox:~/Documents$ cat brojevi.txt
105
11
23
2845
76
ubuntu@ubuntu-VirtualBox:~/Documents$ cat brojevi.txt | sed '/2/d'
105
11
76
ubuntu@ubuntu-VirtualBox:~/Documents$ sed '/2/d' brojevi.txt
105
11
76
```

Vezivanje s i d

```
ubuntu@ubuntu-VirtualBox:~/Documents$ sed -e '/2/d' -e 's/0/a/' brojevi.txt
1a5
11
76
ubuntu@ubuntu-VirtualBox:~/Documents$ sed -e 's/0/a/' -e '/2/d' brojevi.txt
1a5
11
76
ubuntu@ubuntu-VirtualBox:~/Documents$ sed -e 's/0/2/' -e '/2/d' brojevi.txt
11
76
ubuntu@ubuntu-VirtualBox:~/Documents$ sed -e '/2/d' -e 's/0/2/' brojevi.txt
125
11
76
```

Ovde je pokazano kombinovanje *d* i *s* funkcija. Može se uočiti da je redosled funkcija bitan za konačan rezultat.

Opcija q

```
ubuntu@ubuntu2004:~$ cat ekv | sed ' 3 q'
```

```
123
```

```
456
```

```
a
```

```
ubuntu@ubuntu2004:~$ cat ekv | sed -e ' 3 q' -e 's/1/7/'
```

```
723
```

```
456
```

```
a
```

Komanda sed – ispis u fajl

```
ubuntu@ubuntu-VirtualBox:~$ cat ispis.txt
prva vrsta
druga vrsta
treca vrsta
ubuntu@ubuntu-VirtualBox:~$ sed 's\vrsta\rec\' ispis.txt > noviispis.txt
ubuntu@ubuntu-VirtualBox:~$ cat noviispis.txt
prva rec
druga rec
treca rec
ubuntu@ubuntu-VirtualBox:~$ cat ispis.txt
prva vrsta
druga vrsta
treca vrsta
ubuntu@ubuntu-VirtualBox:~$
```

U ovom primeru je editovan fajl ispis.txt a potom je redirekcijom izvršen ispis editovanja u novi fajl.