

# Osnove korištenja operacijskog sustava Linux

## 07. Pretraživanja, filtri i cjevovodi

Nino Nikola Stanušić i Goran Cetušić  
Nositelj: dr. sc. Stjepan Groš

Sveučilište u Zagrebu  
Fakultet elektrotehnike i računarstva

30.10.2017

# Sadržaj

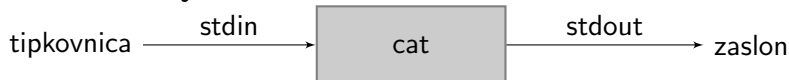
- 1 Pojam standardnog izlaza i ulaza
- 2 Filtri
- 3 Preusmjeravanje
- 4 Često korištene filter naredbe
  - Brojanje znakova, riječi i linija
  - Pretraživanje tekstualnog sadržaja
  - Sortiranje tekstualnog sadržaja
  - Uklanjanje duplih linija
  - Izdvajanje pojedinih polja linije
- 5 Pretraživanje datotečnog sustava
- 6 Pregled Naredbi

# Pojam standardnog izlaza i ulaza

- ▶ Svaki program na Linuxu ima definirane sljedeće ulaze i izlaze
  - Standardni ulaz (stdin)
  - Standardni izlaz (stdout)
  - Standardni izlaz za greške (stderr)
- ▶ Svi ti ulazi i izlazi su vezani na terminal
  - Ako ih nismo preusmjerili uz pomoć specijalnih operatora
- ▶ Svi su predstavljeni kao datoteke
  - Imaju svoje opisnike datoteka (engl. *file descriptor*)
  - Označeni su brojevima 0, 1 i 2

# Primjer naredbe cat

## ► Grafička ilustracija ulaza i izlaza



## ► Naredba cat je filter!

- Preuzima nešto na ulazu
- Filtrira preuzete podatke
- Prosljeđuje rezultat na standardni izlaz

# Filtri

- ▶ Približna definicija filtara bi mogla biti

Svaki program koji ulazne podatke čita sa standardnog ulaza, obrađuje ih na odgovarajući način te rezultat obrade prosljeđuje na standardni izlaz

- ▶ Moguće je vrlo složeno kombiniranje filtara uz pomoć odgovarajućih operatora
  - Ostaje pitanje, zbog čega standardni izlaz za greške?
  - Služi kako bi smo mogli uočiti greške u podacima

# Preusmjeravanje u datoteku (1)

- ▶ Kako bi preusmjerili izlaz neke naredbe u datoteku koristimo operator ">"
  - Pogledajmo razliku između izvršavanja sljedeće dvije naredbe:

```
$ ls -l /  
$ ls -l / > /tmp/test
```
- ▶ Zaključak
  - Druga varijanta naredbe preusmjerava izlaz u datoteku /tmp/test
  - Što ako datoteka već postoji?

## Preusmjeravanje u datoteku (2)

- ▶ Što ako želimo dodati sadržaj u datoteku?
  - Koristit ćemo operator >>
- ▶ Primjer
  - Potrebno je u jednu datoteku dobiti ispis direktorija `/bin`, `/sbin`, `/usr/bin` i `/usr/sbin`
    - Napomena: za svaki direktorij mora se posebno pozivati naredba `ls`!

## Preusmjeravanje u datoteku (3)

### ► Riješenje

```
$ ls -l /tmp/  
$ ls -l /bin/ >> /tmp/output  
$ ls -l /sbin/ >> /tmp/output  
$ ls -l /usr/bin/ >> /tmp/output  
$ ls -l /usr/sbin/ >> /tmp/output
```



# Preusmjeravanje iz datoteke

- ▶ Ako želimo sadržaj neke datoteke preusmjeriti u neki program, koristimo operator `<`
- ▶ Recimo, želimo li pogledati sadržaj datoteke `/etc/passwd`, to možemo obaviti na sljedeći način:  

```
$ cat < /etc/passwd
```

Kod ove naredbe efekt je isti kao i bez korištenja `<`
- ▶ `<` je korisniji kod naredbi koje rade isključivo sa `stdin` ulazom

# Preusmjerenje ulaza i izlaza (1)

- ▶ Moguće je istovremeno preusmjerenje izlaza i ulaza
- ▶ Primjer kopiranja datoteke `/etc/passwd` u `/tmp/passwd` uz pomoć naredbe `cat`

```
$ cat < /etc/passwd > /tmp/passwd
```

```
$ cat > /tmp/passwd < /etc/passwd
```

- ▶ Nije bitan redoslijed operatora preusmjerenja

## Preusmjeravanje ulaza i izlaza (2)

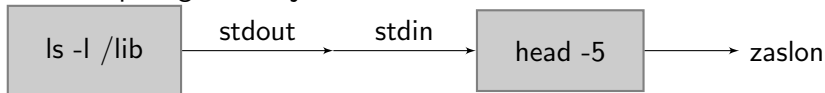
- ▶ Vrlo često programi primaju putanju do ulazne ili izlazne datoteke kroz opcije
- ▶ Oznaka - u tom kontekstu predstavlja `stdin` ili `stdout`
- ▶ Primjer  
`wget -O - http://ftp.hr.debian.org/README`

# Preusmjeravanje između programa (1)

- ▶ Do sada smo isključivo obavljali preusmjeravanje iz i u datoteku
  - Ograničena funkcionalnost
  - Loša efikasnost, ako podatke između programa prebacujemo preko datoteka
- ▶ Način povezivanja programa je upotrebom operatora `|` (engl. *pipe*)
- ▶ Primjer
  - Ispisati prvih pet linija ispisa direktorija `/lib`  
`$ ls -l /lib/ | head -5`
- ▶ Primjer
  - Ispisati sadržaj direktorija `/usr/share` u *pageru*  
`$ ls -l /usr/share | less`

## Preusmjerenje između programa (2)

- ▶ Efektivno, postigli smo sljedeće



- ▶ Zadatak
  - Odredite koji je 10. zapis u datoteci `/etc/group`

# Standardni izlaz za greške (1)

- ▶ Što je sa standardnim izlazom za greške?
- ▶ Pogledajmo ovu naredbu:  

```
$ ls -l /lib1/  
$ ls -l /lib1/ > /tmp/lib5
```
- ▶ Što bi se trebalo dogoditi?
- ▶ Što se dogodilo?
  - Zašto?

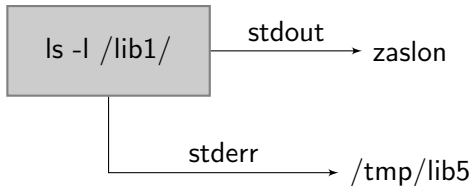
## Standardni izlaz za greške (2)

- ▶ Izlaz za greške možemo preusmjeriti operatorom 2>

```
$ ls -l /lib1/ 2> /tmp/lib5
```

- ▶ Taj operator ne preusmjerava standardni izlaz

```
$ ls -l /lib1/ 2> /tmp/lib5
```

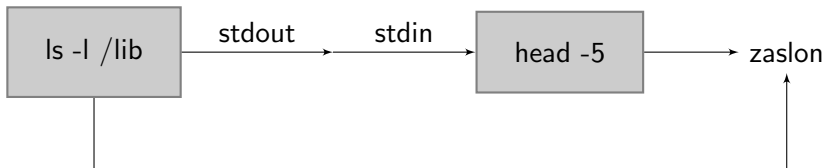


## Standardni izlaz za greške (3)

- ▶ Pogledajmo ovu naredbu:

```
$ ls -l /lib1/ | head -5 2> /tmp/lib5
```

- ▶ Što bi se trebalo dogoditi?
- ▶ Što se dogodilo?
  - Zašto?
  - Kako biste osigurali spremanje svih ispisanih grešaka u datoteku?





## Standardni izlaz za greške (4)

- ▶ Pokušajmo sada sljedeće

```
$ ls -l /lib1/ | head -5 >2 /tmp/lib5
```

- ▶ Što bi se trebalo dogoditi?
- ▶ Što se dogodilo?
  - Zašto?

# Naprednije preusmjerenje (1)

- ▶ Zajedničko preusmjerenje stdout i stderr

```
$ ls &> /dev/null
```

ili 

```
$ ls >& /dev/null
```

- ▶ Preusmjerenje stderr na stdin

```
$ ls 2>&1
```

- ▶ Ljuska nudi vrlo napredne mogućnosti preusmjerenja
  - Proučiti man stranicu :)

## Naprednije preusmjerenje (2)

### ► Primjeri s wget

```
wget -O - http://ftp.hr.debian.org/README > out.txt
```

```
wget -O - http://ftp.hr.debian.org/README 2> err.txt
```

```
wget -O - http://ftp.hr.debian.org/README 2> err.txt > out.txt
```

```
wget -O - http://ftp.hr.debian.org/README &> both.txt
```

## Naprednije preusmjerenje (3)

tee

- ▶ Naredba preusmjerava stdin na stdout i zadanu datoteku
- ▶ `tee --append`
  - Nadopisuje na trenutni sadržaj datoteke
- ▶ Primjer
  - `ls -l /usr/share | tee lista.txt`
  - `ls -l /usr/share | tee -`
    - Ovisno o distribuciji, naredba rezultira ili dvostrukim ili jednostrukim ispisom.

# Here document (1)

- ▶ Koncept koji se vrlo često koristi u skriptama
- ▶ Dio skriptnog koda koji se tretira kao zasebna datoteka
- ▶ Omogućuje navođenja sadržaja cijelih datoteka *inline* u skripti

```
cat << EOF  
Nova datoteka  
linija 2  
...  
linija n  
EOF
```

## Here document (2)

- ▶ Preusmjeravanje u datoteku

```
cat > file.txt << EOF
```

```
Here doc
```

```
EOF
```

- ili 

```
cat << EOF > file.txt
```

```
Here doc
```

```
EOF
```

- ▶ Na sličan način funkcioniraju i *here strings*

```
cat <<< 'Text string'
```

# Uvjetno izvođenje naredbi

- ▶ Naredbe se mogu ulančavati tako da se njihovo izvršavanje uvjetuje
  - \$ true && echo 1
- ▶ drugi niz naredbi izvršava se samo ako je naredba (naredbe) ispred operatora znakova && vratila izlazni status "0"
- ▶ Primjeri

```
$ false && echo 1
```

```
$ false || echo 1
```

# Brojanje znakova, riječi i linija (1)

- ▶ Za prebrojavanje znakova, riječi i linija u tekstu koristimo naredbu `wc` (engl. *word count*)
- ▶ Broji riječi, linije i znakove u datoteci ili na standardnom ulaz
- ▶ Primjer, broj znakova, riječi i linija u datoteci `/etc/passwd`  
`$ wc /etc/passwd`
- ▶ U ispisu prvo je broj linija, potom riječi i na kraju broj znakova



## Brojanje znakova, riječi i linija (2)

- ▶ Opća sintaksa naredbe je

```
wc [opcije] [<datoteka>]
```

- ▶ Opcije su

- w broji samo riječi

- l broji samo linije

- c broji samo znakove

- ▶ Ako se ne navede datoteka, tada se brojanje obavlja u podacima koji pristižu na standardni ulaz, a izlaz je uvijek stdout

# Brojanje znakova, riječi i linija (3)

## ► Zadatak 1

- Prebrojite linije u datoteci `/usr/include/stdio.h` koristeći
  - a) naredbu `wc` i navođenjem datoteke izravno
  - b) operator preusmjeravanja `stdin`
- Razmislite i prokomentirajte rezultat

## ► Zadatak 2

- Koliko datoteka i direktorija ima u direktoriju `/usr/bin`, a koliko ih je u `/usr/include`?

# Pretraživanje teksta (1)

- ▶ Za pretraživanje tekstualnih datoteka koristi se naredba `grep`
  - Traži uzorak u datoteci ili u podacima sa standardnog ulaza
  - Ispisuje na zaslon liniju u kojoj je uzorak pronađen
- ▶ Mnoštvo opcija koje omogućavaju promjenu ponašanja

## Pretraživanje teksta (2)

### ► Primjer upotrebe

- Tražimo niz "root" u datoteci /etc/passwd  
\$ grep root /etc/passwd

### ► U općem slučaju, naredba ima sintaksu

```
grep [opcije] traženi_uzorak [<datoteka>]
```

### ► Neke često korištene opcije

- -v ispisuje sve linije gdje se ne pojavljuje zadani uzorak
- -i pretraživanje bez razlike u malim i velikim slovima

## Pretraživanje teksta (3)

### ► Zadaci

- Pronaći svoje korisničko ime u datoteci `/etc/passwd`
- Potražiti konstantu `SEEK_SET` u datoteci `/usr/include/stdio.h`
- Pronaći sve datoteke u `/usr/lib` koje sadrže `X11` u svom nazivu

# Sortiranje tekstualnih linija (1)

- ▶ Za sortiranje tekstualnih i numeričkih podataka koristimo naredbu `sort`
- ▶ Primjeri
  - Sortirajmo datoteku `/usr/include/stdio.h`  
`$ sort /usr/include/stdio.h`
  - Sortirajmo sve linije iz `/usr/include/stdio.h` koje sadrže riječ `#define`  
`$ grep \#define /usr/include/stdio.h | sort`

## Sortiranje tekstualnih linija (2)

- ▶ Sintaksa naredbe sort je

```
sort [opcije] [<datoteka>]
```

- ▶ Ako se datoteka ne navede tada se sortiraju podaci koji pristižu sa standardnog ulaza. Izlazi se sa ^D nakon sto su uneseni svi elementi.
- ▶ Rezultat se ispisuje na standardni izlaz

# Uklanjanje duplih linija (1)

- ▶ Za traženje i manipulaciju duplim linijama koristi se naredba `uniq`
  - Sintaksa naredbe je  
`uniq [opcije] [<datoteka>]`
- ▶ Češće korištene opcije su
  - u Ispisuje linije koje se ne ponavljaju
  - d Ispisuje linije koje se ponavljaju barem jednom
  - c Ispisuje za svaku liniju koliko puta se ponavlja



## Uklanjanje duplih linija (2)

- ▶ Linije moraju biti sortirane prije korištenja ove naredbe!
- ▶ Primjer, provjerimo ima li duplih linija u jednoj od datoteka u direktoriju `/usr/share/dict/`

npr. `$ uniq -d /usr/share/dict/words`

# Izdvajanje pojedinih polja linije (1)

- ▶ Za izdvajanje pojedinih polja u liniji koristimo naredbu `cut`
  - ▶ Podrazumijevani znak za razdvajanje polja je razmak
    - Primjer datoteke s poljima `/etc/passwd`
    - Primjer, ispis korisničkih imena na sustavu
- ```
$ cut -f1 -d: /etc/passwd
```

## Izdvajanje pojedinih polja linije (2)

- ▶ Sintaksa naredbe je sljedeća  
`cut [opcije] [<datoteka>]`
- ▶ Neke češće korištene opcije su
  - d definira znak koji razgraničava polja
  - f definira polja koja je potrebno propustiti na izlaz
  - c ispisuje po kolonama
- ▶ Ako nije navedena datoteka tada se obrada obavlja na podacima sa standardnog ulaza

## Izdvajanje pojedinih polja linije (3)

### ► Primjer izrezivanja po kolonama

```
$ cut -c2-5,10-12 /etc/passwd
```

### ► Zadatak

- Potrebno je generirati popis svih različitih vrijednosti koje se javljaju u zadnjem polju datoteke /etc/passwd
- Koliko puta se svaka vrijednost ponavlja?

# Pretraživanje datotečnog sustava (1)

- ▶ Za pretraživanje datoteka i direktorija koristimo naredbu `find`
  - Vrlo kompleksna naredba s nizom mogućnosti
- ▶ Sintaksa naredbe je

```
find [opcije] [<direktoriji>] <uvjeti>
```
- ▶ `<direktoriji>` je popis direktorija koje je potrebno pretražiti
- ▶ `<uvjeti>` definiraju uvjete pretraživanja

## Pretraživanje datotečnog sustava (2)

- ▶ Primjeri nekih češće korištenih uvjeta
  - name <ime> Traži datoteku sa zadanim imenom
  - type <tip> Traži datoteku zadanog tipa
  - size <tip> Traži datoteku zadane veličine
  - print Ispisuje ime zadane datoteke

# Pretraživanje datotečnog sustava (3)

## ► Primjeri

- pronadimo datoteku passwd u direktoriju /etc  
`$ find /etc -name passwd`
- pronadimo sve datoteke u /etc veće od 1M  
`$ find /etc -size +1M`
- pronadimo sve datoteke u /etc manje od 1k  
`$ find /etc -size -1k`

# Pretraživanje datotečnog sustava (4)

## ► Primjeri

- pronađimo sve direktorije u /etc direktoriju  
`$ find /etc -type d`
- pronađimo sve datoteke u /etc direktoriju  
`$ find /etc -type f`
- pronađimo sve datoteke direktorije veće od 10k  
`$ find /etc -type d -a -size +10k`
- Operator -a se podrazumijeva



## Pretraživanje datotečnog sustava (5)

### ► Primjeri

- pronadimo sve direktorije i datoteke veće od 1M u /etc direktoriju  
`$ find /etc -type d -o -size +1M`
- isto kao i prethodni primjer, ali izlistava pronađene datoteke/direktorije  
`$ find /etc -type d -o -size +1M -ls`

`man find`

- Pogledati opcije naredbe `find` za MAC vremena

# Naredba whereis

- ▶ Pronalazi gdje se nalazi izvorni kod naredbe, binarni zapis ili stranica priručnika
- ▶ Sintaksa
  - `whereis [<opcije>] <datoteka>`
- ▶ Neke od opcija su
  - b traži binarne zapise
  - m traži stranice priručnika
  - s traži izvorni kod

# Naredba locate (1)

- ▶ Traži datoteku koja u svojem imenu sadrži zadani uzorak – preporučeno je njeno korištenje umjesto naredbe `find`
  - Manje opterećuje disk i brža je
  - Koristi internu bazu podataka za pretragu
- ▶ Sintaksa naredbe
  - `$ locate [<opcije>] <uzorak>`
- ▶ Korisna opcija je `-i` za ignoriranje razlika u slovima

## Naredba locate (2)

### ► Primjer

- Želimo pronaći sve datoteke passwd
- ```
$ locate passwd
```

### ► Zadatak

- Kreirati datoteku ovo\_je\_primjer u matičnom direktoriju
- Pokušati je pronaći naredbom locate
- Što se dogodilo?
- Kako ispraviti problem?

# Pregled naredbi

>	preusmjeravanja u datoteku
>>	preusmjeravanje i nadodovanje
<	preusmjeravanje iz datoteke
	cijevovod
wc	brojanje znakova, rijeci i linija
grep	pretrazivanje teksta
sort	sortiranje
uniq	uklanjanje duplih linija
cut	izdvajanje pojedinih linija
find	pretrazivanje datoteka i direktorija
locate	pretrazivanje datoteka i direktorija
whereis	pronalaži izvorni kod naredbe