

SVEUČILIŠTE U MOSTARU
FAKULTET STROJARSTVA, RAČUNARSTVA I ELEKTROTEHNIKE
PREDDIPLOMSKI STUDIJ RAČUNARSTVA

OPERACIJSKI SUSTAVI

VJEŽBE

Nastavnik: prof.dr.sc. Sven Gotovac
gotovac@fesb.hr

Asistent: doc.dr.sc. Željko Šeremet
zeljko.seremet@fsre.sum.ba

MOSTAR, TRAVANJ 2024.

PRETRAŽIVANJA, FILTRI I CJEVOVODI



SADRŽAJ

- Pojmovi standardni ulaz, standardni izlaz, standardni izlaz za greške, filter
- Preusmjeravanja
- Često korištene filter naredbe
 - Brojanje karaktera, riječi i linija
 - Pretraživanje tekstualnog sadržaja
 - Sortiranje tekstualnog sadržaja
 - ...
- Traženje datoteka i direktorija



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



OPISNICI IZLAZA I ULAZA

- Svi su predstavljeni kao datoteke
 - Imaju svoje opisnike datoteka (file descriptor)
 - Označeni su brojevima 0, 1 i 2
- Zadatak
 - Izlistajte podatke o datoteci /dev/stdin
 - Na što pokazuje?
 - Koja je krajnja datoteka?



PRIMJER NAREDBE cat

- Grafička ilustracija izlaza i ulaza



- 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 proslijeđuje 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/`. Što ako datoteka već postoji?



PREUSMJERAVANJE U DATOTEKU (2)

- Što ako želimo dodati sadržaj u datoteku?
 - Koristit ćemo operator >>
- Zadatak
 - 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 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
```

- Međutim, navedeni primjer nije dobar
 - Naredba `cat` ionako koristi `stdin` ili joj možemo dati ime datoteke pa je preusmjeravanje višak



PREUSMJERAVANJE ULAZA I IZLAZA

- Moguće je istovremeno preusmjeravanje 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 preusmjeravanja



PREUSMJERAVANJE IZMEĐU PROGRAMA

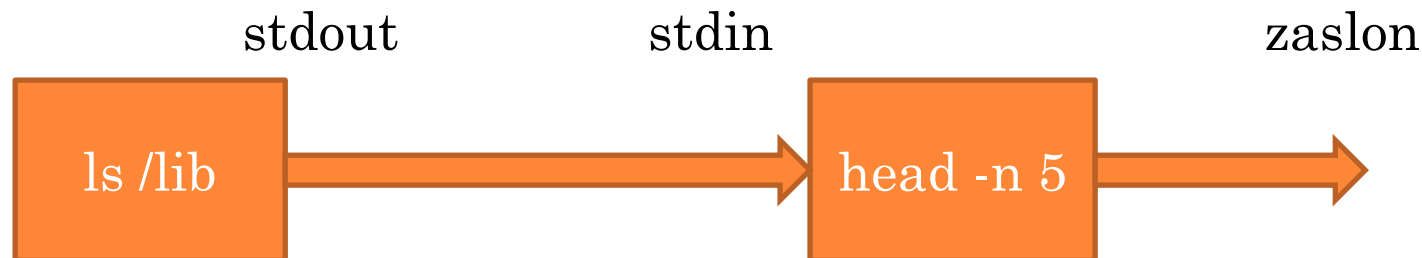
- Do sada smo isključivo obavljali preusmjerenje iz u datoteku
 - Ograničena funkcionalnost
 - Loša efikasnost ako podatke između programa prebacujemo preko datoteka
- Način povezivanja programa je upotrebom operatora `|` (pipe)
- Primjer, kako bi vidjeli prvih pet linija ispisa direktorija `/lib`, možemo napraviti sljedeće:

```
$ ls /lib | head -n 5
```



PREUSMJERAVANJE IZMEĐU PROGRAMA

- 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 /lib1 > /tmp/lib5
```

- Što bi se trebalo desiti?
- Što se desilo?
 - Zašto?



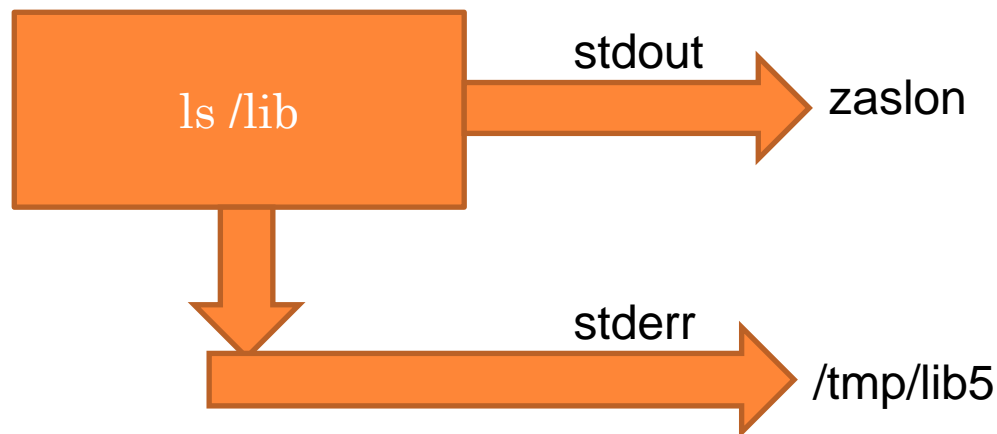
STANDARDNI IZLAZ ZA GREŠKE (2)

- Izlaz za greške možemo preusmjeriti operatorom **2>**

\$ ls /lib1 2> /tmp/lib5

- Taj operator ne preusmjerava standardni izlaz

\$ ls /lib1 2> /tmp/lib5



STANDARDNI IZLAZ ZA GREŠKE (3)

- Pogledajmo ovu naredbu:

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

- Što bi se trebalo desiti?
- Što se desilo?
 - Zašto?

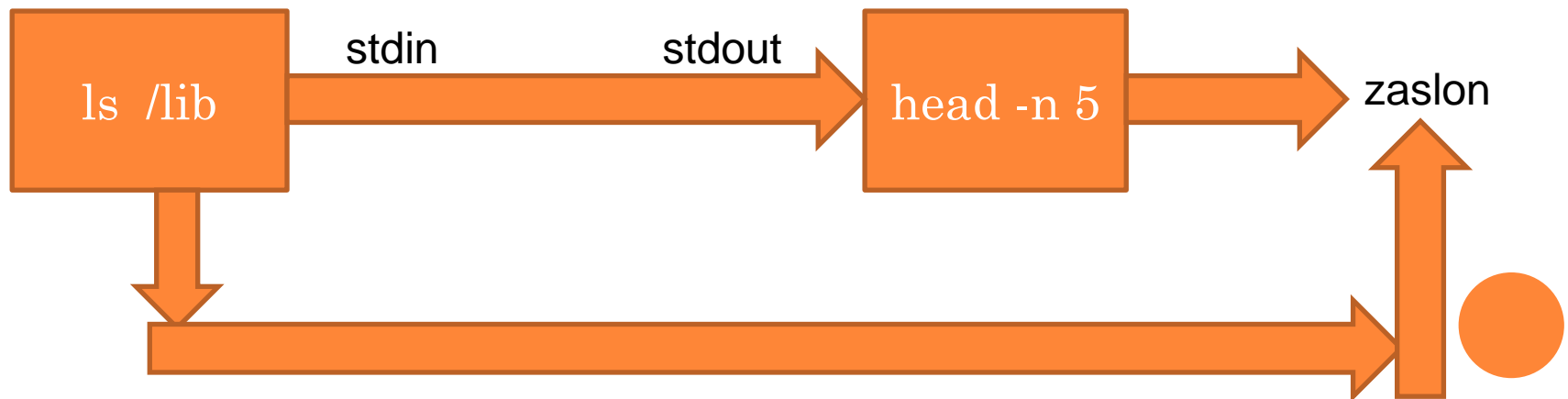


STANDARDNI IZLAZ ZA GREŠKE (4)

- Pokušajmo sada sljedeću varijantu

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

- Što bi se trebalo desiti?
- Što se desilo?
 - Zašto?



NAPREDNIJE PREUSMJERAVANJE (1)

- Sve se temelji na konceptu deskriptora datoteka (file handler)
 - stdin, deskriptor 0
 - stdout, deskriptor 1
 - stderr, deskriptor 2
- Programi pišu u odgovarajuće deskriptore
 - OS uvijek bira najniži slobodni deskriptor



NAPREDNIJE PREUSMJERAVANJE (2)

○ Primjeri

- Zajedničko preusmjerenje **stdout** i **stderr**
\$ ls2 &> /dev/null
- Preusmjerenje **stderr** na **stdin**
\$ ls2 2>&1

○ Ljuska nudi vrlo napredne mogućnosti preusmjerenja

- Proučiti man stranicu :)



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



BROJANJE KARAKTERA, RIJEČI I LINIJA (3)

○ Zadaci

- Koliko linija ima u datoteci `/usr/include/clif.h`
- 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 `INITREQ_H` u datoteci `/usr/include/initreq.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/wordexp.h**
\$ sort /usr/include/wordexp.h
 - Sortirajmo sve linije iz **/usr/include/wordexp.h** koje sadrže riječ **#define**
\$
grep \#define /usr/include/wordexp.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
- Rezultat se ispisuje na standardni izlaz
- DZ
 - Pogledati opcije f,n,k



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 datoteci **/usr/share/dict/words**
\$ 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

- pronađimo 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
```

○ DZ

- 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 ju pronaći naredbom **locate**
- Kojom naredbom se ažurira baza?



KRAJ

