

Osnove korištenja operacijskog sustava Linux

05. Varijable i kontrola toka

Mateo Stjepanović

Leonard Volarić Horvat

Nositelj: dr. sc. Stjepan Groš

Sveučilište u Zagrebu

Fakultet elektrotehnike i računarstva

04.11.2017

Sadržaj

- 1 Varijable
- 2 Uvjeti
- 3 Operatori usporedbe
- 4 Kontrola toka
- 5 Petlje
- 6 Zamjenski znakovi

Variable

- ▶ Variable se mogu koristiti kao u bilo kojem drugom jeziku
- ▶ Nije potrebno deklarirati varijablu (sličnost s Pythonom)
- ▶ Svaka je varijabla tipa *string*
- ▶ Dvije vrste varijabli
 - Globalne varijable (Vidljivo iz više shellova)
 - Lokalne varijable (Vidljive samo u lokalnom shellu)
- ▶ Mogućnosti rada s varijablama
 - Spremanje vrijednosti u varijablu
 - Čitanje vrijednosti iz varijable

Primjer postavljanja i čitanja varijabli

- ▶ Pridruživanje je jednostavnog oblika: `varijabla=vrijednost`
 - `hello="Hello World"`
 - `num=2`
 - VAŽNO: razmaka ne smije biti ni s jedne strane znaka jednakosti
- ▶ `varijabla=$(naredba)` - preusmjeravanje izlaza naredbe u varijablu
- ▶ `export varijabla` - pretvaranje varijable u globalnu za **child** procese
- ▶ `echo $varijabla`
 - `echo $hello`
 - `echo $num`
- ▶ Lokalna varijabla unutar skripte - `local`
- ▶ Perzistentno globalne varijable treba spremiti na neki drugi način:
 - `~/.bashrc`
 - `~/.profile`
 - `/etc/environment`

```
#!/bin/bash
HELLO=Hello
function hello {
    local HELLO=World
    echo $HELLO
}
echo $HELLO
hello
echo $HELLO
```

Uvjeti

- ▶ Prije ulaska u kontrolu toka potrebno je navesti uvjet kontrole
- ▶ Uvjeti (*tests*) su označeni s []
- ▶ Vraćaju 0 ili 1 ovisno o tome zadovoljava li test zadani uvjet
 - [1 -eq 1]
- ▶ NB: potreban razmak između zagrada i uvijeta
- ▶ echo \$? ispisuje rezultat testa
 - ako je vraćena 0, to znači True (sjetite se C-ovskog *return 0*;))

Operatori usporedbe

- ▶ Operatori određuju uvjete kontrole toka
- ▶ Vrste operatora:
 - Operatori integera
 - Operatori stringa
 - Operatori datoteka

Operatori integera

- ▶ -lt (manje od)
- ▶ -gt (veće od)
- ▶ -le (manje od ili jednako)
- ▶ -ge (veće od ili jednako)
- ▶ -eq (jednako)
- ▶ -ne (različito)

Operatori stringa

- ▶ <(manje od)
- ▶ >(veće od)
- ▶ = (jednako)
- ▶ != (različito)
- ▶ -n (vraća 0 ako niz ima duljinu)
- ▶ -z (vraća 0 ako je niz prazan)

Operatori datoteka

- ▶ -e (datoteka postoji)
- ▶ -s (datoteka ima veličinu veću od 0)
- ▶ -d (datoteka je direktorij)

Kontrola toka

- ▶ Sada se možemo upustiti u kontrolu toka
- ▶ Tri naredbe za kontrolu toka:
 - `if`
 - `else`
 - `elif`
- ▶ Znajući gore navedene operatore i testove idemo prikazati primjere kontrole toka

► IF - struktura i primjer

```
#!/bin/bash
HELLO="hello"
if [ $HELLO="hello" ]
then
    echo "$HELLO world"
fi
```

► ELSE - struktura i primjer

```
#!/bin/bash
if [ -e datoteka.txt ]
then
    cat datoteka.txt
else
    echo "Datoteka ne postoji!"
fi
```

► ELIF - struktura i primjer

```
#!/bin/bash
pom="kul"
if [ pom="kul" ]
then
    echo "Linux je $pom"
elif [ pom="bezveze" ]
then
    echo "Linux masterrace"
else
    echo "Linux je OS intelektualne elite"
fi
```

Petlje

- ▶ Bash, naravno, podržava i petlje
- ▶ `while` i `for` petlje konceptualno identične analogima iz drugih jezika
- ▶ Uz njih, postoje još i `until` i `select`
 - `for`: iterira po nekoj listi i **za** svaki član liste radi nešto
 - `while`: izvršava se **sve dok** je neki uvjet ispunjen
 - `until`: izvršava se **sve dok** se neki uvjet **ne ispuni**
 - `select`: generira jednostavan **izbornik** na temelju predane liste

- ▶ FOR - struktura i primjer
- ▶ "C-ovski" for:

```
echo "Brojevi od 42 do 27 djeljivi s 3: "  
for (( i=42; i>=27; i=i-3)) do  
    echo $i  
done
```

- ▶ "pythonovski" for (suštinski foreach):

```
names='Marvin Shooty Hactar Magrathea'  
for name in $names; do  
    echo $name;  
done
```

► WHILE - struktura i primjer

```
cnt=0
while [ $cnt -le 10 ]; do
    echo $cnt;
    ((cnt++)); # expression expansion
done
```

► obrada datoteke liniju po liniju

```
while read line; do
    echo $line; # ispisi liniju
    echo $line | rev; # obrni liniju
done
```

► UNTIL - struktura i primjer

```
cnt=15
until [ $cnt -lt 10 ]; do
    echo $cnt;
    ((cnt--));
done
```

► SELECT - struktura i primjer

```
distros='Ubuntu Arch CentOS Gentoo Quit'
select distro in $distros; do
    if [ $distro == 'Quit' ]; then
        break;
    fi;
    echo Tvoja je omiljena distribucija $distro
done
echo "Bilo mi je drago"
```

Zamjenski znakovi

- ▶ engl. wildcards
- ▶ Koriste se za brzo i efikasno pretraživanje i izvršavanje naredbi
- ▶ Mogu se koristiti kod svih naredbi koje prihvaćaju datoteke ili direktorije kao argument!
 - Ljuska **prije** pokretanja naredbi uklanja zamjenske znakove, pronalazi sve datoteke koje odgovaraju izrazu i postavlja ih umjesto zamjenskog izraza kao da su direktno uneseni
 - Potom pokreće naredbu koja ne dobiva nikakve informacije o zamjenskim znakovima!

Zamjenski znakovi

Osnovni zamjenski znakovi

- ? Odgovara točno jednom znaku
- * Odgovara bilo kojem broju znakova (ili nijednom)
- [xyz] Odgovara točno jednom znaku iz skupa xyz

► Primjer 1

Izlistajte sve datoteke čiji naziv počinje s b i nalaze se u /bin direktoriju

```
$ ls -l /bin/b*
```

► Primjer 2

Izlistati sve naredbe u /bin direktoriju koje se sastoje od točno dva znaka

```
$ ls -l /bin/??
```

Zamjenski znakovi

► Primjer 3

Izlistati sve datoteke u `/bin` direktoriju koje završavaju slovom `d`

```
$ ls -l /bin/*d
```

► Primjer 4

Ispisati sve datoteke u `/bin` direktoriju koje započinju s `a`, `b` ili `c`

- Jedna mogućnost

```
$ ls -l /bin/a* /bin/b* /bin/c*
```

- Kraće

```
$ ls -l /bin/[abc]*
```

Zamjenski znakovi

► Primjer 5

Izlistati sve datoteke u `/bin` direktoriju koje u sebi sadrže barem jednu znamenku

```
$ ls -l /bin/*[0123456789]*
```

- Efikasnije: Zadavanjem **raspona znakova**

```
$ ls -l /bin/*[0-9]*
```

Zamjenski znakovi

Invertiranje skupa

- ▶ Korištenjem znaka \wedge moguće je invertirati skup znakova u zagradi

- ▶ **Primjer 6**

Izlistati sve datoteke u direktoriju `/usr/bin` koje ne započinju sa malim slovom abecede

```
$ ls -l /usr/bin/[^a-z]*
```

- ▶ Ako znakove \wedge i `-` trebamo koristiti kao dio traženog skupa znakova tada \wedge ne smije biti naveden na prvom mjestu u grupi, a `-` mora biti naveden kao prvi ili zadnji znak

Zamjenski znakovi

Isključenje značenja posebnih znakova

- ▶ Ponekad ne želimo posebno značenje zamjenskih znakova
 - Što ako baš imamo datoteku koja se zove * ?
 - U tom slučaju upotrebljavamo navodnike ili znak \

```
$ ls -l "/bin/b*"
```

```
$ ls -l /bin/b\*
```

Literatura

<http://blog.tgrrrtt.com/bash-101>

http://linuxcommand.org/lc3_wss0110.php

[https:](https://www.cyberciti.biz/faq/set-environment-variable-linux/)

[//www.cyberciti.biz/faq/set-environment-variable-linux/](https://www.cyberciti.biz/faq/set-environment-variable-linux/)

[https://ryanstutorials.net/bash-scripting-tutorial/
bash-variables.php](https://ryanstutorials.net/bash-scripting-tutorial/bash-variables.php)

[http:](http://tldp.org/LDP/Bash-Beginners-Guide/html/sect_03_02.html)

[//tldp.org/LDP/Bash-Beginners-Guide/html/sect_03_02.html](http://tldp.org/LDP/Bash-Beginners-Guide/html/sect_03_02.html)

<http://tldp.org/HOWTO/Bash-Prog-Intro-HOWTO-5.html>