

# Otvoreni operativni sistemi

Marko Dimitrijević

## Unix/ Linux Shell

- **Shell** je komandni interpreter, koji omogućava korisniku da efikasno pokreće i kontroliše procese, radi sa fajl sistemom i hardverom.
- Shell je proces koga sistem tretira kao i druge procese.
- Shell se pokreće kada se korisnik uloguje na sistem. Predstavlja **parent** proces za sve procese koji se pokreću iz shell-a, i od njega nasleđuju redirekciju standardnog ulaza i izlaza, kao i dijagnostičkog izlaza.
- Postoji više različitih shell-ova, korisniku je data mogućnost da izabere shell

## Funkcije i mogućnosti shell-a

- Omogućuje pisanje i izvršavanja skripta sa konstrukcijama kao što su petlje, uslovne naredbe i slično.
- Mogućnost praćenja signala koje šalje UNIX.
- Mogućnost praćenja kodova koje šalju procesi (**echo \$?**).
- U cilju povećanja sigurnosti sistema, nekim korisnicima se dodeljuje shell sa ograničenim mogućnostima.
- Korisniku stoji na raspolaganju mogućnost da preimenuje neke komande ili da više komandi ujedini u jednu (**alias**).
- Mogućnost da se komande smeštaju u bafer, kasnije modifikuju i ponovo izvršavaju (**history**).

## Funkcije i mogućnosti shell-a

- Mogućnost da se u komandnoj liniji automatski dovrši ime započetog fajla (**autocomplete**).
- Mogućnost obavljanja aritmetičkih operacija nad promenljivima u skriptu bez UNIX komandi.
- Mogućnost praćenja i pristupa poslovima koji se odvijaju u pozadini.
- Mogućnost da se prethodne ili tekuća komandna linija modifikuju tekst editorom.

## Tipovi

- Bourne Shell **/bin/sh**
- Bourne-Again Shell **/bin/bash**
- C Shell **/bin/csh**
- Korn Shell **/bin/ksh**
- Turbo C Shell **/bin/tcsh**
- Z Shell **/bin/zsh**
- No login **/sbin/nologin**
  
- **Korisnik može promeniti shell comandom**

**chsh -s /bin/shell**

## Variable

- Variable su definisane u okviru svakog shell-a.
- Utiču na ponašanje programa ili sistema u celini.
- Mogu se prikazati komandom **env**
- Variable počinju simbolom **\$**: **\$USER**, **\$SHELL**, **\$PWD**,...
- Pojedinačne varijable se mogu prikazati komandom **echo**:

```
echo $USER
```

```
echo $SHELL
```

```
echo $PWD
```

```
echo $HOME
```

## Nastavljanje i razdvajanje komandi

Simbol	
;	<p>Razdvajanje komandi.</p> <p>Primer:</p> <pre>% ls -l; rm test.txt &lt;Enter&gt;</pre>
\	<p>Nastavljanje komande u više redova.</p> <p>Primer:</p> <pre>% cat test.txt \ grep "linux" \ more &lt;Enter&gt;</pre>

## Specijalni znaci i navodnice

Simbol	
<code>\</code>	Sprečava specijalno značenje narednog znaka. Primer - brisanje fajla ime":  <b>% rm ime\"</b>
<code>' ... '</code>	Shell tretira uokvireni tekst kao jednu reč.
<code>" ... "</code>	Uokvireni tekst se tretira kao jedna reč ali se ! i \$ tretiraju kao specijalni karakteri. Primer:  <b>% echo '\$HOME \$PWD' \$HOME \$PWD</b> <b>% echo "\$HOME \$PWD" \$HOME \$PWD</b>
<code>`command`</code>	Zamena komande sa rezultatom. Primer:  <b>% rm `find / -name core`</b>



## Bafer komandi - History

Simbol	
<b>!!</b>	Prethodna komanda. Primer:  % <b>!!</b> <Enter>  Izvršava ponovo prethodnu komandu.
<b>!n</b>	Komandna linija broj <i>n</i> . Broj <i>n</i> se može otkriti sa <b>history</b> .
<b>!-n</b>	Komandna linija sa brojem za <i>n</i> manji od tekućeg.
<b>!string</b>	Zadnja komandna linija koja počinje sa <i>string</i> . Primer:  % cc proba.c pr.c obj.o -o proba -lm ... % <b>!cc</b>
<b>!?string?</b>	Zadnja komandna linija koja sadrži <i>string</i> .

## Bafer komandi - History

Simbol	
<b>!<i>n</i>*</b>	Svi argumenti <i>n</i> -te komandne linije. Ako se ne zada <i>n</i> podrazumeva se prethodna linija. Primer:  <b>% cc proba.o pr.o obj.o -o proba</b> <b>% rm !*</b>
<b>!<i>n</i>^</b>	Prvi argument <i>n</i> -te komandne linije.
<b>!<i>n</i>\$</b>	Zadnji argument <i>n</i> -te komandne linije.
<b>!<i>i</i>:<i>j</i></b>	<i>j</i> -ti argument <i>i</i> -te komandne linije
<b>^<i>str1</i>^<i>str2</i></b>	U zadnjoj komandi zameni <i>str1</i> sa <i>str2</i> .
<b>!<i>n</i>:<i>s</i>/<i>str1</i>/<i>str2</i></b>	U <i>n</i> -toj komandnoj liniji zameni <i>str1</i> sa <i>str2</i> .

# Regularni izrazi

Simbol	
?	Zamenjuje samo jedan znak. Primer:  <code>% ls prim?.c</code> <code>prim1.c prim2.c</code>
*	Zamena bilo kog stringa. Primer:  <code>% ls *.c</code> <code>prim1.c prim2.c primer.c</code>
[...]	Zamenjuje se bilo koji znak iz zagrade. Primer:  <code>% ls prim[abc1].c</code> <code>prima.c primc.c prim1.c</code>

# Regularni izrazi

Simbol	
<b>[x-y]</b>	Zamenjuje bilo koji znak u opsegu od x do y. Primer:  % ls primer <b>[a-g].c</b> primerb.c primerd.c
<b>{ string..}</b>	Grupisanje. Primer:  % ls <b>{ x *y* ?z* }</b> x myfile yourfile xza yza

# Poslovi

<b>&amp;</b>	Izvršavanje komande u pozadini ( <i>background</i> ). Primer:  <b>% find / -name core &gt; zauzece &amp;</b>
<b>&lt;Ctrl&gt; Z</b>	Privremeno zaustavljanje (suspendovanje) procesa
<b>jobs</b>	Ispisuje listu svih procesa koji su suspendovani ili se izvršavaju u pozadini
<b>bg [%job]</b>	Komanda za prebacivanje procesa u pozadinu. Ako nema argumenata prebacuje se tekući posao (zadnje prekinuti). Primer:  <b>% infinite &lt;Ctrl&gt; Z</b> <b>% bg %1</b>

# Poslovi

<b>fg [%job]</b>	Komanda za prebacivanje procesa iz pozadine napred ( <i>foreground</i> ). Primer:  <b>% infinite &amp; % fg %1</b>
<b>stop [%job]</b>	Prekida proces koji se izvršava u pozadini. Primer:  <b>% stop %1</b>
<Ctrl> C	Prekidanje tekuceg procesa (kill).

## Kontrola i informacije o procesima

- Informacije o aktivnim procesima koji se izvršavaju na sistemu se može dobiti komandom **ps**.
- Opcije:
  - ps -e** - ispisuje informacije o svim procesima
  - ps -f** - ispisuje potpunu informaciju o procesima
  - ps -u user** - ispisuje informacije o procesima korisnika

## Kontrola i informacije o procesima

- Svaki proces ima jedinstveni identifikacioni broj PID (process ID)
- Komanda **ps -fe** daje informacije o:
  - Korisniku (**UID**),
  - Identifikacionom broju procesa (**PID**),
  - PID-u procesa koji je pokrenuo proces (parent PID, **PPID**),
  - Procenat iskorišćenja procesora (**C**),
  - Vreme kada je proces pokrenut (**STIME**),
  - Konzola kojoj pripada proces (**TTY**),
  - Ukupno iskorišćenje procesorskog vremena (**TIME**),
  - Komanda koja je pokrenula proces (**CMD**)



# Kontrola i informacije o procesima

UID	PID	PPID	C	STIME	TTY	TIME	CMD
root	3047	1	0	Oct15	tty6	00:00:00	/sbin/mingetty /dev/tty6
root	3050	591	0	Oct15	?	00:00:00	/sbin/udevd -d
root	3051	591	0	Oct15	?	00:00:00	/sbin/udevd -d
root	3074	3034	0	Oct15	?	00:00:00	/usr/libexec/gdm-simple-slave --display-id ...
root	3077	3074	0	Oct15	tty1	00:07:45	/usr/bin/Xorg :0 -nr -verbose -audit 4 .....
root	3099	1	0	Oct15	?	00:00:00	/usr/sbin/console-kit-daemon --no-daemon
gdm	3169	1	0	Oct15	?	00:00:00	/usr/bin/dbus-launch --exit-with-session
gdm	3170	1	0	Oct15	?	00:00:00	/bin/dbus-daemon --fork --print-pid 5 .....
gdm	3171	3074	0	Oct15	?	00:00:00	/usr/bin/gnome-session --autostart=/us.....
root	3174	1	0	Oct15	?	00:00:00	/usr/libexec/devkit-power-daemon
gdm	3178	1	0	Oct15	?	00:00:08	/usr/libexec/gconfd-2
gdm	3210	3171	0	Oct15	?	00:00:00	/usr/libexec/polkit-gnome-authentication-agent
gdm	3211	3171	0	Oct15	?	00:01:24	/usr/libexec/gdm-simple-greeter
root	3213	1	0	Oct15	?	00:00:00	/usr/libexec/polkit-1/polkitd
gdm	3214	3171	0	Oct15	?	00:00:25	gnome-power-manager
gdm	3217	3171	0	Oct15	?	00:00:00	plymouth-log-viewer --icon
root	3225	3074	0	Oct15	?	00:00:00	pam: gdm-password
root	10811	1	0	Oct15	?	00:00:00	squid -f /etc/squid/squid.conf
squid	10813	10811	0	Oct15	?	00:45:25	(squid) -f /etc/squid/squid.conf
squid	10821	10813	0	Oct15	?	00:00:00	(unlinkd)
postfix	12647	2340	0	00:21	?	00:00:00	pickup -l -t fifo -u
root	18487	2239	0	01:43	?	00:00:00	sshd: root@pts/0
root	18489	2239	0	01:43	?	00:00:00	sshd: root@notty
root	18491	18487	0	01:43	pts/0	00:00:00	-zsh
root	18511	18489	0	01:43	?	00:00:00	/usr/libexec/openssh/sftp-server

## Kontrola i informacije o procesima

- Interaktivna informacija o aktivnim procesima koji se izvršavaju na sistemu se može dobiti komandom **top**.
- Opcije:
  - top -n num** - ponavlja se **n** iteracija
  - top -a** - sortira procese prema utrošku memorije
  - top -pPID** - ispisuje informacije samo o procesu **PID**
  - top -u user** - ispisuje informacije o procesima korisnika
- Top daje informacije o sistemu,

# Signali

- Signali su oblik među procesne komunikacije (**Inter-Process Communication, IPC**)
- Predstavljaju asinhronone notifikacije koje se šalju procesima ili nitima (delovi procesa koji se paralelno izvršavaju)
- Signali se mogu procesu poslati od strane drugih procesa, ili od strane korisnika.

# Signali

Signal	Broj signala	Opis
<b>SIGHUP</b>	1	<b>Hang up</b> – terminal zatvoren ili parent proces okončan
<b>SIGINT</b>	2	<b>Interrupt</b> – proces prekinut (Ctrl + C)
<b>SIGQUIT</b>	3	<b>Quit</b> – Korisnik je izašao iz aplikacije (Ctrl + D)
<b>SIGFPE</b>	8	Nedozvoljena matematička operacija
<b>SIGKILL</b>	9	<b>Kill</b> – proces se trenutno zaustavlja i ne radi cleanup
<b>SIGALRM</b>	14	<b>Alarm</b> – clock signal (koristi se kod tajmera)
<b>SIGTSTP</b>	20	<b>Terminal stop</b> – suspenduje proces (Ctrl + Z)

## Signali

- Signali se određenom procesu može poslati komandom **kill**:

**kill [-SIG] PID**

Komanda šalje signal **SIG** procesu **PID**.

- Spisak svih signala koje podržava sistem:

**kill -l**

## Signali

- Signali se grupi procesa može poslati komandom **killall**:

**killall [-SIG] CMD**

Komanda šalje signal **SIG** procesima koji su pokrenuti komandom **CMD**.

- Opcije:

**killall -SIG -u USER**

-šalje signal procesima koje je startovao USER