



SHELL - rad iz komandne linije

Linux ▼

Grafičko radno okruženje opterećuje procesor i povećava rizik u smislu sigurnosti sistema, tako da se, po pravilu, ne instalira na serverima. Tada sistem administratorima na raspolaganju ostaje komandni interpreter (shell) i prateći skup alata za rad sa fajlovima.

KOMANDNI INTERPRETER (shell)

Shell je interfejs između korisnika i kernela, odnosno jezgra OS-a. Shell prihvata komande koje korisnik zadaje, zatim ih interpretira i potom ih izvršava, pri čemu po potrebi pokreće odgovarajuće programe. Na UNIX sistemima postoji više različitih komandnih interpretera, a korisnici u toku rada po potrebi mogu preći iz jednog u drugi.

Komandni interpreter je proces koji obavlja sledeće funkcije:

- interpretaciju komandne linije,
- pokretanje programa,
- redirekciju ulaza i izlaza,
- povezivanje komandi u pipeline,
- rad sa fajlovima iz komandne linije
- zamenu imena datoteka,
- rukovanje promenljivama i kontrolu okoline (environment),
- shell programiranje.

Interpretiranje komandne linije

Kad se korisnik prijavi na sistem u kontekstu tekućeg login procesa izvršava se proces shell, odnosno komandni interpreter. Na ekranu se prikazuje komandni prompt (shell prompt), a to je najčešće znak **\$**, **ukoliko se na sistem prijavi običan korisnik, odnosno #, ukoliko se na sistem prijavi root**. Kada korisnik zada neku komandu (odnosno otkuca neki tekst i pritisne Enter), shell to pokušava da interpretira. Tekst unet u shell prompt naziva se komandna linija (command line), čiji je opšti oblik:

```
$ command [opcije] [argumenti]
```

Znak \$ je odzivni znak komandnog interpretera (shell prompt). Komanda može biti **interna (ugrađena u shell) ili eksterna (realizovana kao poseban program koji se nalazi u sistemskoj putanji)**. Opcije i argumenti su parametri koje shell prenosi komandi, pri čemu su argumenti najčešće obavezni i predstavljaju ime nekog fajla, direktorijuma, korisnika ili, na primer, identifikator procesa.

Ime komande, opcije i argumenti razdvajaju se razmakom. Shell interpretira razmak kao graničnik i na osnovu toga razdvaja argumente i opcije od imena komande. U jednu komandnu liniju može se uneti **najviše 256 karaktera**. Imena većine UNIX komandi po pravilu se formiraju od malih slova. Više UNIX komandi mogu se navesti u istoj komandnoj liniji ukoliko su razdvojene znakom tačka-zarez.

```
$ cal          # samo komanda
$ df /dev/sda  # komanda (fd) i argument (/dev/sda)
$ cp 1.txt 2.txt # komanda (cp) i dva argumenta (1.txt i 2.txt)
$ date -u      # komanda (date) i opcija (-u)
$ ls -l /etc   # komanda (ls), opcija (-l) i argument (/etc)
$ clear ; date # dve komande koje se izvršavaju jedna za drugom
```

Opcije su osetljive na velika i mala slova (case-sensitive) i mogu se navesti na dva načina:

- -x znak minus (-) praćen jednim slovom,
- --option dva znaka minus (--) praćena punim imenom opcije.

echo

Jedna od često korišćenih komandi je **echo** koja prikazuje tekst ili vrednost promenljive na ekranu. Sintaksa komande echo je:

```
$ echo [opcije] [string, promenljive...]
```

Opcije:

- -n ova opcija ne prebacuje kursor u novi red, nakon izvršenja echo komande
- -e omogućava interpretaciju sledećih karaktera u kombinaciji sa obrnutom kosom crtom:
 - \a upozorenje (alert bell)
 - \b povratak unazad (backspace)
 - \c ne prelaziti u novi red (suppress trailing new line)
 - \n novi red (new line)
 - \r povratak na početak reda (carriage return)
 - \t horizontalni tabulator (horizontal tab)
 - \ obrnuta kosa crta (backslash)

Primer.

```
$ echo -e "Petar\n\t\t Petrovic"
```

```
Petar
```

```
Petrovic
```

pwd (Print Working Directory)

Sve u Linux-u je fajl i ova konstatacija će postajati sve jasnija kako se budemo detaljnije upoznavali sa Linux-om. Za sada je bitno da se svi fajlovi nalaze unutar direktorijuma (foldera) koji su organizovani u hijerarhiji stabla. Prvi direktorijum (koren stabla) se naziva root direktorijum. Root direktorijum ima fajlove i poddirektorijume koji mogu da sadrže fajlove i direktorijume...

```
/
|-- bin
|   |-- file1
|   |-- file2
|-- etc
|   |-- file3
|   |-- directory1
|       |-- file4
|       |-- file5
|-- home
|-- var
```

Lokacija fajla ili direktorijuma se naziva **putanja**. Ako imamo direktorijum koji se zove *home* i koji sadrži poddirektorijum *marko* koji u sebi ima folder *Movies*, onda je putanja do tog direktorijuma **/home/marko/Movies**.

Pri kretanju po sistemu fajlova (fajl sistemu), kao i u realnom životu, korisno je znati gde se trenutno nalazimo i gde hoćemo da idemo. Da bi videli gde se trenutno nalazimo možemo koristiti komandu **pwd**, skraćeno od Print Working Directory (Štampa Radnog Direktorijuma), koja prikazuje u kom direktorijumu se trenutno nalazimo. Putanja je prikazana od root direktorijuma.

```
$ pwd
```

cd (Change Directory)

Kada znamo gde se nalazimo (**pwd**) možemo da naučimo i kao da se krećemo. Da bi se kretali moramo da koristimo putanje. Postoje dva načina da se odredi putanja:

- **Apsolutna putanja** je putanja od root direktorijuma. Root direktorijum se obično prikazuje kao *slash (/)* karakter. Svaki putanja koja počinje */* je putanja koja počinje od root direktorijuma. Npr. */home/marko/Desktop*.
- **Relativna putanja** je putanja od trenutne lokacije. Ako se nalazimo u direktorijumu */home/marko/Documents* i želimo da stignemo u direktorijum *porez* koji se nalazi unutar direktorijuma *Documents* nije potrebno navoditi celu putanju */home/marko/Documents/porez*, možemo da koristimo samo *porez/*.

Komanda za kretanje po direktorijumima je **cd** (change directory).

```
$ cd /home/marko/Pictures
```

Sada smo promenili lokaciju na */home/marko/Pictures*.

Ako u direktorijum *Pictures* imamo direktorijum *Zurke* možemo do tog direktorijuma stići komandom

```
$ cd Zurke
```

Pored navigacije apsolutnim i relativnim putanjama komanda **cd** ima i skraćenice

- *.* - trenutni direktorijum,
- *..* - direktorijum roditelj,
- *~* - home direktorijum,
- *-* - prethodni direktorijum.

```
$ cd .
```

```
$ cd ..
```

```
$ cd ~
```

```
$ cd -
```

ls (List Directory)

Kada znamo da se krećemo po sistemu potrebno je da znamo i šta nam je dostupno. Za to možemo da koristimo komandu **ls** koja lista sadržaj direktorijuma. Komanda **ls**, bez argumenata, lista sadržaj direktorijuma u kome se trenutno nalazimo, ali je moguće je i uneti putanju do direktorijuma za koji hoćemo da dobijemo sadržaj.

```
$ ls
```

```
$ ls /home/marko
```

Komanda **ls** neće prikazati sve fajlove. Ako ime fajla počinje sa *.* taj fajl je skriven. Ako prosledimo opciju *-a* onda komanda **ls** prikazuje i skrivene fajlove.

```
$ ls -a
```

Još jedna korisna opcija je *-l*, koja prikazuje detaljnu listu fajlova. Detaljna lista sadrži dozvole za fajlove, broj linkova, ime vlasnika, vlasničku grupu, veličinu fajla, vreme kreiranja ili poslednje modifikacije i ime fajla/direktorijuma.

```
$ ls -l
```

Moguće je proslediti jednoj komandi više opcija odjednom. Za komandu **ls** moguće je proslediti *-la*. Redosled navođenja opcija određuje i redosled izvršavanja opcija. U većini slučajeva redosled nije bitan tako da je moguće komandi **ls** proslediti opciju *-la* i *-al*.

```
$ ls -la
```

touch

Jedan od načina za kreiranje fajlova je komanda **touch**. **Touch** kreira novi prazan fajl.

```
$ touch prazanfajl
```

Touch takođe menja vreme poslednje modifikacije postojećeg fajla.

file

Na linux sistemima ime fajla ne mora da reprezentuje sadržaj fajla (kao u Windows sistemima). Moguće je kreirati fajl funny.gif koji u stvari nije GIF.

Da bi saznali koji je tip fajla, možemo da koristimo komandu **file**.

```
$ file banana.jpg
```

cat

Osnovna komanda za prikaz sadržaja fajla je komanda **cat**, skraćeno od concatenate. Ona može i da kombinuje više fajlova i prikže njihov sadržaj.

```
$ cat dogfile birdfile
```

Komanda cat nije pogodna za prikaz velikih fajlova.

less

Ako je potrebno prikazati sadržaj velikog tekstualnog fajla možemo koristiti komandu less.

```
$ less /home/marko/Documents/text1
```

Za navigaciju se koriste sledeće komande:

- q - Izlazak iz less i povratak u shell,
- Page up, Page down, Up i Down - Navigacija po dokumentu,
- g - Pozicionira se na početak fajla,
- G - Pozicionira se na kraj fajla,
- /reč - Pronalazi zadatu reč u dokumentu.
- h - poziva pomoć za less.

Ponavljanje komandne linije (history)

Komandni interpreter bash upisuje svaku komandnu liniju u history fajl. Ovo omogućava da se prethodne komande ponove, pri čemu se pre ponovnog izvršavanja mogu i izmeniti. Komande se takođe mogu ponavljati na osnovu rednog broja koji im je pridružen u history datoteci. Bash shell history datoteku smešta u home direktorijum korisnika (~/.bash_history), i u njoj podrazumevano čuva 1000 prethodno izvršenih komandi. Broj komandi koje se mogu smestiti u ovu datoteku može se promeniti pomoću promenljive HISTSIZE - na primer, ako je HISTSIZE=500, to znači da se u datoteku ~/.bash_history mogu smestiti 500 prethodno izvršenih komandi. Komanda history u bash shellu prikazuje prethodno izvršene komande:

```
$ history 3
331 finger
332 mail
333 history 5
```

cp

Copy: kopiranje fajla/direktorijuma na specificiranu lokaciju

Primeri:

```
$ cp /home/a.a /tmp/b.b
$ cp a* /tmp
$ cp /etc/[a-d][1-5]* .
$ cp -r /etc /tmp/oldconfig
```

kopiranje direktorijuma /etc sa svim poddirektorijumima i datotekama u direktorijum /tmp/oldconfig/etc (datoteka /etc/passwd kopira se u /tmp/oldconfig/etc/passwd),

```
$ cp -r /etc/* /tmp/oldconfig
```

kopiranje kompletnog sadržaja direktorijuma /etc u direktorijum /tmp/oldconfig (datoteka /etc/passwd kopira se u

```
/tmp/oldconfig/passwd),
```

```
$ cp -r a* /tmp/mybackup
```

kopiranje datoteka čije ime počinje sa a iz tekućeg direktorijuma i svih poddirektorijuma u direktorijum /tmp/mybackup.

Zamena imena fajlova – JOKER znaci

Džoker karakteri: *, ? i []. Argument komande koji sadrži džoker karakter zamenjuje se odgovarajućom listom datoteka shodno pravilima zamene. Komandni interpreter izvršava ovu zamenu pre izvršavanja same komande, odnosno pre pokretanja programa.

```
$ echo *
```

```
myfile1 kyuss.txt file3 anotherfile3 file4
```

- karakter * menja bilo koji niz znakova proizvoljne dužine
- karakter ? menja bilo koji znak (tačno jedan znak)
- opseg [poc-kraj] menja tačno jedan znak koji pripada datom opsegu.

Opseg se ne sme zadati u opadajućem redosledu.

```
$ ls -d /etc/[a-d][a-d]*
```

```
/etc/acpi  
/etc/adduser.conf  
/etc/bash.bashrc  
/etc/bash_completion  
/etc/bash_completion.d  
/etc/ca-certificates  
/etc/ca-certificates.conf  
/etc/calendar  
/etc/dbus-1  
/etc/dconf
```

mv

Move: pomeranje fajla na drugu lokaciju ili promena imena

rm

Remove: brisanje fajla

mkdir

Make directory: kreiranje specificiranog direktorijuma

rmdir

Remove directory: brisanje direktorijuma

find

Traži fajlove čiji atributi zadovoljavaju kriterijume pretrage u direktorijumu koji je naveden kao početna tačka pretrage i svim poddirektorijumima, rekurzivno; ukoliko korisnik ne naznači komandi šta da uradi sa datotekama koje pronađe, komanda neće izvršiti nikakvu akciju.

```
$ find / -name urgent.txt -print
```

```
$ find /tmp -user jsmith -size +50 - print
```

```
$ find /home/jsmith -name "*.old" -print
```

Ostali kriterijumi pretrage su:

- **username** **uname**
- **groupname** **gname**
- **atime** **n** traže se datoteke kojima niko nije pristupio tačno n dana (n mora biti ceo broj, a dozvoljeni su i oblici -

```
n i +n);
```

- **mtime** **n** traže se datoteke koje niko nije modifikovao `--`
- **perm mode** prava pristupa zadata u oktalnom obliku
- **links** **n** traže se sve datoteke sa **n** hard linkova (**n** mora biti ceo broj, a dozvoljeni su `i -n i +n`);
- **type** **x** traže se sve datoteke koje su tipa **x**, pri čemu **x** može biti **b** (blok uređaj), **c** (karakter uređaj), **d** (direktorijum), **p** (imenovani pipe);
- **inode** **n** traže se sve datoteke čiji je **i-node** **n**;
- **newer fname** traže se sve datoteke koje su modifikovane pre datoteke **fname**;
- **local** traže se sve datoteke koje se nalaze na lokalnim diskovima.

```
$ find . -name "*.c" -exec echo {} \;
```

Dobijanje pomoći

Navođenje opcije `--help` u samoj komandi.

Na primer:

```
$ mkdir --help
```

Usage: mkdir [OPTION]... DIRECTORY...

Create the DIRECTORY(ies), if they **do** not already exist.

Mandatory arguments to long options are mandatory **for** short options too.

```
-m, --mode=MODE  set file mode (as in chmod), not a=rwx - umask
-p, --parents     no error if existing, make parent directories as needed
-v, --verbose    print a message for each created directory
-Z              set SELinux security context of each created directory
                to the default type
--context[=CTX] like -Z, or if CTX is specified then set the SELinux
                or SMACK security context to CTX
--help          display this help and exit
--version       output version information and exit
```

GNU coreutils online [help](http://www.gnu.org/software/coreutils/): <<http://www.gnu.org/software/coreutils/>>

Full documentation at: <<http://www.gnu.org/software/coreutils/mkdir>>

or available locally via: info '(coreutils) mkdir invocation'

Ispisuje na ekranu sintaksu i objašnjenja za odgovarajuće argumente i opcije, bez detaljnijeg opisa same komande. Ukoliko objašnjenje ne može stati na jedan ekran - pipeline sa komandom `less` (command `--help less`).

Man stranice

Jedan od najkompletnijih izvora pomoći (ponekad i jako komplikovan i nejasan) su stranice uputstva za korišćenje komande (manual page, odnosno man page).

```
$ man command
```

whereis

Prikazuje lokaciju izvršnih datoteka, izvornog koda i prateće dokumentacije programa

```
$ whereis [-bms] command
```

bez parametara prikazuje lokacije svih elemenata programa

- **-b** izvršne datoteke
- **-m** uputstva
- **-s** izvorni kôd

Primer:

```
$ whereis insmod
```

```
insmod: /sbin/insmod /usr/share/man/man8/insmod.8.gz
```

upotreba komande `whereis` za pronalaženje lokacije programa `insmod` (koji se koristi za dodavanje modula u

aktivno Linux jezgro)

which

Prikazuje samo lokaciju izvršnih datoteka; traži izvršnu datoteku u direktorijumima navedenim u sistemskoj putanji i ukoliko je nađe, prikazuje putanju i ime prve pronađene komande

```
$ which [-a] command
```

Primeri:

```
$ which insmod
/sbin/insmod
$ which fdisk
/sbin/fdisk
```

apropos

Na ekranu prikazuje ime i opis svih komandi koje u opisu imaju zadati string.

```
$ apropos whoami
ldapwhoami      (1) - LDAP who am i? tool
whoami          (1) - print effective userid
```

Alternativno ime komande (alias)

Alias je način dodele kraćeg imena pomoću kog se određena komanda, ili niz komandi, može pozvati iz komandnog interpretera. Na primer, može se dodeliti alias **ll** (long listing) koji izvršava komandu **ls -l**. Alias je aktivan samo u komandnom interpreteru za koji je napravljen. Za korn i bash alias se dodeljuje na sledeći način:

```
$ alias aliasname=value
```

Jednom postavljen alias se poništava komandom unalias

```
$ unalias aliasname
```

Primeri:

komandi se može dodeliti kraće alternativno ime

```
$ alias h=history
$ alias c=clear
```

jednom komandom se može zameniti sekvenca komandi

```
$ alias home="cd;ls"
```

može se kreirati jednostavno ime za izvršavanje komandi sa određenim parametrima

```
$ alias ls="ls -l"
$ alias copy="cp -i"
```