

VEŽBA 1

Pregled

Jedan od klasičnih, ali i danas veoma aktuelnih pristupa računaru jeste *shell*. To je interpreter komandi koje korisnik zadaje preko tastature, a celokupna komunikacija se obavlja tekstualnim putem preko terminala. U savremenim uslovima terminal je obično simuliran u okviru grafičkog radnog okruženja.

Osim što može da neposredno izvršava korisnikove komande, shell može da izvršava i unapred napisane nizove komandi. Komandama se pridružuju i promenljive i upravljačke strukture, pa ovakvi nizovi komandi poprimaju oblik pravih programa koji se uobičajeno nazivaju *skriptovima*. Skriptovi su uvek u formi tekstualnog fajla i za razliku od drugih programskih jezika ne prevode se na mašinski jezik. Tema ove i nekoliko narednih vežbi biće shell skriptovi i to pod verzijom shell-a poznatim kao *bash*. Bash je trenutno najrasprostranjeniji shell na Linux platformama.

Elementi shell jezika

Komande

Sve komande se shell-u zadaju direktno (upisivanjem na terminal) nakon što je shell dao znak da je spreman za prijam nove komande. Ovaj znak se uobičajeno naziva **promptom** i prepoznaje se po specijalnom znaku `$`.

```
[user@hostname ~]$
```

Pre znaka `$` se navode aktuelne informacije i to redom: korisnik koji je prijavljen na sistem, ime računara na kom se shell odvija i tekući radni direktorijum. Znak `~` označava podrazumevani radni direktorijum korisnika, njegov tzv. home.

Svaka komanda počinje ključnom rečju koja je istovremeno i **ime komande**. Ime komande se poklapa sa imenom (izvršnog) fajla koji sadrži program koji komandu izvodi. Komande imaju parametre čiji broj nije ograničen, a njihovo značenje specifično je za svaku komandu. Parametri su međusobno odvojeni znakom **praznog mesta** (*engl. space*).

```
ime_komande par_1 par_2 par_3 ...
```

Shell programi – skriptovi

Niz komandi se može upisati u tekstualni fajl. Svaka komanda se nalazi u sopstvenom novom redu. One se automatski redom izvršavaju. Na taj način nastaju shell programi, tzv. skriptovi.

U prvoj liniji svakog shell skripta treba da stoji

```
#!/bin/bash
```

Ovo omogućava da skript bude automatski prepoznat i izvršen kao shell program.

Znak `#` inače predstavlja oznaku za komentar u shell skriptovima.

Prenos promenljivih u novi proces

Shell se odvija kao proces pod operativnim sistemom (UNIX/Linux). Svaka komanda ili skript otvara novi proces u sistemu. Kada je u pitanju skript, novi proces je nova instanca shell procesa. Novim procesima se vrednosti promenljivih ne predaju podrazumevano, nego samo ukoliko su označene za „izvoz“ komandom `export`.

```
export VAR1
```

Ova komanda označiće za izvoz promenljivu `VAR1`.

```
export VAR2=VREDNOST_KOJA_SE_ZADAJE
```

U ovoj varijanti, istovremeno se promenljivoj zadaje i vrednost i označava se za izvoz.

Standardni ulaz i izlaz

Kao i u jeziku C, tekući proces poseduje ulaz i izlaz. Ovo se odnosi i na shell i sve komande/programe koji se iz njega pozivaju.

Cev (pipe)

Cev se formira navođenjem komandi u jednom redu, a između njih se stavlja karakter „vertikalna crta“ - `|`. Ovim se standardni izlaz prethodne komande automatski preusmerava u standardni ulaz naredne.

```
cmd1 | cmd2 | cmd3
```

Komanda `cat`

Naziv asocira na **concatenate** (poveži, spoj). Bez argumenata celokupan standardni ulaz preusmerava u standardni izlaz. Ako se zadaju argumenti, fajlovi čija su imena navedena se redom tretiraju kao standardni ulaz. Ako komanda nije u cevi (*engl. pipeline*), standardni ulaz će biti ono što se preko tastature unosi u terminal. Kraj unosa se u tom slučaju obeležava pritiskom na kombinaciju `Ctrl+D`.

Automatska promenljiva koja sadrži standardni izlaz komande

Često je neophodno obrađivati elemente koje je neka komanda poslala na standardni izlaz. Način da se ovo uradi je korišćenjem automatske promenljive koja sadrži taj izlaz:

```
$(<komanda sa argumentima>)
```

Zadaci

1. Napisati skript koji na standardnom izlazu (konzoli) ispisuje poruku po izboru (npr. `Hello World, shell!`). Skript treba da je moguće pozvati navođenjem imena skripta kao komande.
2. Napisati skript koji na standardnom izlazu ispisuje vrednost promenljive `SVAR`. Prethodno se promenljiva definiše neposrednim zadavanjem komande dodele shell-u. Radi li skript kako je zamišljeno? Šta nedostaje? Popraviti tako da radi kako je zamišljeno.
3. Napisati skript koji svoj prvi argument (parametar sa komandne linije) ispisuje 3 puta na standardni izlaz. Preusmeriti standardni izlaz na fajl. Koristiti odgovarajuće navodnike za navođenje strignova koji sadrže znakove za prazna mesta. Promeniti skript tako da 3 puta

ispisuje prvi argument u formi – 1. argument je: <argument br. 1>.

4. Napisati skript koji sadržaj svog standardnog ulaza dodeljuje promenljivoj STI, a onda ga ispisuje 3 puta. Generisati test fajl sa proizvoljnim tekstom. Obavezno testirati i korišćenjem preusmeravanja standardnog ulaza i operacijom pajpovanja (|).