

# *Shell* skripte

## Operativni sistemi

Fakultet tehničkih nauka, Univerzitet u Novom Sadu

17.01.2021.

# Shell skripte

- *Shell* skripte su programi koje interpretira *shell*.
- Sintaksa njihovog koda je identična komandama koje korisnik direktno unosi.
- Koriste se za automatizaciju raznih operacija, a naročito u slučaju kada je potrebno kombinovati više različitih programa.

# Pokretanje *shell* skripti

- Mogu se pokrenuti indirektno, pokretanjem novog *bash* procesa sa *shell* skriptom kao argument.
- Ukoliko prvi red skripte sadrži definiciju interpretera, mogu se pokrenuti i direktno kao bilo koji drugi program.
  - Definicija interpretera za *bash* skripte je:  
“#!/bin/bash”

# Komentari

- Tekst *shell* skripte koji se nalazi unutar jednog reda nakon simbola “#” je komentar i neće biti interpretiran od strane *shell*-a.

# Promenljive

- Promenljive u *shell* skriptama se dele na proizvoljne, automatske, i promenljive okruženja.
- Podrazumeva se da promenljive sadrže *string* vrednost.

# Promenljive

- Sintaksa za dodelu vrednosti promenljivama je “naziv=vrednost”.
  - Razmaci oko znaka “=” nisu dozvoljeni!
- Stavljanjem znaka “\$” ispred naziva promenljive se dobija vrednost te promenljive. *Shell* će izvršiti bukvalnu zamenu pojave teksta “\$naziv” sa vrednošću promenljive *naziv*.

# Promenljive okruženja

- Promenljive okruženja se razlikuju od običnih time što su dostupne programima koji su pokrenuti od strane *shell*-a.
- Komanda *env* listira promenljive okruženja.
- Komanda *export* pretvara običnu promenljivu u promenljivu okruženja.
- Promenljive okruženja se mogu i privremeno postaviti za određenu komandu, tako što se dodela vrednosti napiše pre naziva komande.

# Automatske promenljive

- Ove promenljive održava sam *shell* kako bi korisnik ili skripta mogli da ih koriste, a to su:
  - “\$\$” - PID samog *shell* procesa.
  - “\$?” - Povratni kod prethodno izvršenog programa.
  - “\$#” - Broj argumenata komandne linije.
  - “\$0”, “\$1”, “\$2”, ... - Pristup argumentima komandne linije.
  - “\$!” - PID poslednjeg procesa pokrenutog u pozadini.



# Povratna vrednost

- Nakon završetka procesa, operativni sistem čuva njegovu povratnu vrednost, sve dok je proces koji ga je pokrenuo ne pročita.
- Komanda u *shell*-u se smatra uspešnom ako ima povratnu vrednost 0.
- Povratne vrednosti različite od nule se smatraju kodom greške.

# Uslovno povezivanje komandi

- “&&” - Izvršava komandu sa desne strane samo ako se komanda sa leve strane završi uspešno.
- “||” - Izvršava komandu sa desne strane samo ako se komanda sa leve strane završi neuspešno.

# Ispitivanje uslova

- Komanda *test* ispituje izraz koji joj je dat preko komandne linije i vraća povratnu vrednost koja odgovara tačnošću datog izraza.
- *Bash* ima svoju ugrađenu varijantu *test* komande, koja se koristi postavljanjem izraza između dvostrukih uglastih zagrada, “[[“ i “]]”.

# Ispitivanje uslova

- Upoređivanje stringova se vrši sa binarnim operatorima “=” i “!=”, koji proveravaju da li su stringovi jednaki ili različiti, tim redom.
- Pri tome, mogu da se koriste specijalni karakteri “\*” i “?”, za upoređivanje stringova sa šablonom.
- *Bash* podržava i upoređivanje sa regularnim izrazima pomoću operatora “=~”.

# Ispitivanje uslova

- Upoređivanje numeričkih vrednosti se vrši sa sledećim binarnim operatorima:
  - “-eq” proverava da li su brojevi jednaki.
  - “-ne” proverava da li su brojevi različiti.
  - “-gt” proverava da li je levi broj strogo veći od desnog.
  - “-ge” proverava da li je levi broj veći ili jednak desnom.
  - “-lt” proverava da li je levi broj strogo manji od desnog.
  - “-le” proverava da li je levi broj manji ili jednak desnom.

# Ispitivanje uslova

- Postoje dodatni unarni operatori vezani za datoteke, koji se stavljaju ispred naziva datoteke:
  - “-e” proverava da datoteka postoji.
  - “-f” proverava da datoteka postoji i da je obična.
  - “-d” proverava da datoteka postoji i da je direktorijum.
  - “-s” proverava da datoteka postoji i da nije prazna.
  - Još mnogo operatora koji su detaljno opisani u uputstvu...

# Uslovno izvršavanje

- Naredba *if* se može koristiti za uslovno izvršavanje bloka komandi.

- Sintaksa ove naredbe izgleda ovako:

```
if uslovni izraz 1
then
    blok komandi 1
elif uslovni izraz 2
then
    blok komandi 2
else
    blok komandi 3
fi
```

# Uslovno izvršavanje

- Naredba *case* se takođe može koristiti za uslovno izvršavanje bloka komandi.

- Sintaksa ove naredbe izgleda ovako:

```
case izraz in
    šablon 1) blok komandi 1 ;;
    šablon 2) blok komandi 2 ;&
    šablon 3) blok komandi 3 ;;
    *) blok komandi 3 ;;
esac
```

- “;;” prekida izvršavanje *case*-a dok “;&” izvršava i sledeći blok.



# While petlja

- Naredba *while* se može koristiti za ponavljanje izvršenja bloka komandi dok je određeni uslov **tačan**.
  - Sintaksa ove naredbe izgleda ovako:  

```
while uslovni izraz  
do  
    blok komandi  
done
```

# *Until* petlja

- Naredba *until* se može koristiti za ponavljanje izvršenja bloka komandi dok je određen uslov **netačan**.
  - Sintaksa ove naredbe izgleda ovako:  

```
until uslovni izraz  
do  
    blok komandi  
done
```

# For petlja

- Naredba *for* izvršava blok naredbi za svaki element liste, dodeljujući taj element promenljivoj u svakom ciklusu.
  - Sintaksa ove naredbe izgleda ovako:

```
for promenljiva in lista
do
    blok komandi
done
```

# C-asta *For* petlja

- Drugi oblik naredbe *for* više podseća na *for* petlju u programskom jeziku C:  
for ((*početak*; *uslov*; *promena*))  
do  
    *blok komandi*  
done

# Redirekcija

- *Shell* omogućava povezivanje ulaza ili izlaza sa datotekama tokom izvršavanja komande.
  - Znak “>” prosleđuje izlaz komande u datoteku. Ako datoteka već postoji, biće prvo ispražnjena.
  - Znak “<” prosleđuje sadržaj datoteke kao ulaz komandi
  - Znak “>>” prosleđuje izlaz komande u datoteku. Ako datoteka već postoji, novi sadržaj će biti dodat na kraj.

# Deskriptori

- Postoje tri toka podataka u terminalu: standardni ulaz, standardni izlaz, i standardna greška.
- Ovim tokovima se dodeljuju numerički deskriptori 0, 1, i 2, tim redom.
- Prilikom redirekcije, za ulaz se podrazumeva standardni ulaz, dok se za izlaz podrazumeva standardni izlaz.

# Deskriptori

- Moguće je izvršiti redirekciju i samo toka standardne greške, sa “2>”.
- Deskriptori se mogu redirektovati i u drugi deskriptor. Ispred deskriptora koji zamenjuje datoteku treba da stoji znak “&”.
  - Na primer, ukoliko želimo da spojimo standardnu grešku i standardni izlaz, koristimo “2>&1”.