

Prva laboratorijska vježba

Osnove korištenja operacijskog sustava Linux

11. studenoga 2019.

Za svaki zadatak potrebno je napisati po jednu bash skriptu.

Zadatak 1.

- Stvorite direktorij **LAB1** i premjestite se u njega.
- Stvorite novi direktorij **source** i u njemu prazan file **empty**.
- U direktorij **source** kopirajte sadržaj direktorija **/boot** i direktorija **/etc**.
Napomena: *Koristite jednu naredbu.*
- Ispišite zauzeće direktorija **source** koristeći SI prefikse (potencije broja 10) mjernih jedinica.
- U direktoriju **LAB1** stvorite simboličku poveznicu **target** na direktorij **source**.
- Premjestite se u direktorij **target** bez dereferenciranja poveznice i ispišite adresu trenutnog direktorija. Pokažite da ispis adrese trenutnog direktorija daje **LAB1/target**.
- Vratite se u direktorij **LAB1** i premjestite se u direktorij **target** koristeći dereferenciranje poveznice. Pokažite da ispis adrese trenutnog direktorija daje **LAB1/source**.
- Koristeći poveznicu **target** odredite veličinu direktorija **source**.
- Koristeći naredbu **touch** stvorite praznu datoteku **source/novi** i postavite joj vrijeme izmjene (mtime) tako da bude isto kao i datoteci **source/empty**. Koristite jednu naredbu.
- Izbrišite sve stvorene direktorije i datoteke u direktoriju **LAB1** koristeći jednu naredbu.
- Izbrišite prazan direktorij **LAB1** koristeći naredbu za brisanje praznog direktorija.

Zadatak 2.

- a) Napišite bash skriptu koja čita tekst sa standardnog ulaza i ispisuje ga na standardni izlaz. Nad pročitanim podacima ne treba raditi izmjene.
- b) Koristeći **moгуćnosti upravljanja datotekama** u Linuxu ostvarite tekstualnu komunikaciju između dva korisnika na istom računalu. Pretpostavite da svaki korisnik može imati dva terminala stalno otvorena. Poruke poslane od jednog korisnika moraju odmah biti prikazane drugom korisniku. U implementaciji smijete koristiti najviše dvije datoteke. Rješenje mora biti u obliku jedne ili dvije bash skripte koje pripremaju i pokreću komunikaciju. U komentarima skripata objasnite postupak slanja i primanja poruka.

Zadatak 3.

Zanima nas koliko će se puta u nekoj godini dogoditi petak 13. Srećom, nalazimo se u Linux okruženju gdje nam na raspolaganju stoji naredba `ncal`¹, te poznamo naredbe za pretraživanje teksta koje će nam olakšati ovaj zadatak.

- Prije svega, proučite man stranice naredbe `ncal`. Mi ćemo ju koristiti u vrlo jednostavnom obliku: `ncal <godina>`.
- Koristeći mogućnosti pretraživanja teksta prije svega iz ispisa izbacite sve linije koje ne počinju² nizom znakova `Fr`.³
- Nadalje, koristeći opcije naredbe za pretraživanje teksta, prebrojite koliko **ukupno** ima ponavljanja broja 13 u tim linijama. Primijetite da u ovom trenutku skripta mora uistinu brojati koliko se puta u godini dogodio petak 13.
- Sljedeće što želimo dodati je da skripta od korisnika prima godinu za koju provjerava koliko puta će se dogoditi petak 13. i zatim ispisuje poruku na `stdout`. U predlošku je primjer koji možete koristiti, a on koristi naredbu `read` za čitanje inputa korisnika. Dok korisnik ne unese godinu, skripta se neće izvršavati.

Primjer 1: Čitanje inputa

```
#!/bin/bash
echo -n "Upisi godinu za koju te zanima koliko puta se dogodio petak 13.: "
read godina;
petkovi=$(ncal $godina | <ostatak super kul naredbe>)
echo "U godini $godina, petak 13. se dogodio $petkovi puta."
```

Napomena: detaljnije o varijablama u ljusci možete čuti u nastavku vještine, ali da bismo zadali nešto interesantniji zadatak, odlučili smo ipak već sada koristiti varijable. Zasada iznosimo sljedećih nekoliko naputaka:

- Konstruktor `$(naredba)` omogućuje korištenje vrijednosti na izlazu naredbe za npr. spremanje u neku varijablu, kao što je to ovdje varijabla `petkovi`.
- Adresiranje varijable se radi tako da se imenu varijable *prepend* \$, kao što vidite u primjeru: `$petkovi`
- Pripazite na razmake! Bash je pomalo osjetljiv i ne očekuje razmake pri pridruživanju vrijednost
Ispravan primjer: `a=5`
Pogrešan primjer: `a = 5`

- Za one koji žele znati više:** Sada kad imamo i lijep ispis zadatka, možda bi korisnika zanimalo koliko će se puta petak 13. dogoditi u narednih x godina. Koristeći istu filozofiju, nakon prvog ispisa zatražite sljedeći upis "do koje godine te zanima koliko puta će se dogoditi petak 13.: " gdje će korisnik unijeti, primjerice, godinu "2100", a program mu ispisati:

Primjer 2: Ispis

```
okosl@poseidon:~$ bash labos.sh
Upisi godinu za koju te zanima koliko puta se dogodio petak 13.:
2017
U godini 2017, petak 13. se dogodio 2 puta.
```

¹Postoji mogućnost da se naredba ne nalazi u osnovnoj instalaciji vašeg operacijskog sustava, u tom slučaju instalacijom paketa `bsdmainutils` možete preuzeti naredbu.

²Kao što ćete vidjeti u predavanju koje obrađuje regularne izraze, početak linije se označava znakom `^`, pa je regularni izraz koji provjerava je li "Fr" na početku linije: `^Fr`

³Ili, jasno, `Pe` od "Petak" ili nešto treće, ovisno o lokalizacijskim postavkama.

```

Upisi godinu za do koje te zanima koliko puta ce se dogoditi petak 13.:
2100
U godini 2017, petak 13. se dogodio 2 puta.
U godini 2018, petak 13. se dogodio 2 puta.
U godini 2019, petak 13. se dogodio 2 puta.
U godini 2020, petak 13. se dogodio 2 puta.
....
U godini 2100, petak 13. se dogodio 1 puta.

```

Potreban nam je još jedan blok koji će sada iterirati kroz zadani opseg godina

Primjer 3: Iteracija po opsegu

```

#!/bin/bash ...
echo "U godini $godina, petak 13. se dogodio $petkovi puta."
echo -n "Upisi godinu za do koje te zanima koliko puta ce se dogoditi petak 13.: "
read opseg
for godina in $(seq 2017 $opseg);
do
    petkovi=$(ncal $godina | <ostatak super kul naredbe>)
    echo "U godini $godina , petak 13. se dogodio $petkovi puta."
done

```

Napomena: kao i s varijablama, ovdje smo, pak, posegnuli za petljama. Primjer bi vam trebao biti dovoljan za upoznavanje s *for* petljom u Bashu, a detaljnije o kontroli toka kasnije.