

Skriptni jezici – zadaci za 1. ciklus laboratorijskih vježbi

ožujak 2012.

1 Uvod

U okviru prvog ciklusa laboratorijskih vježbi utvrđuju se i praktično primjenjuju znanja o osnovnim Unix alatima, baratanju regularnim izrazima, te pisanju jednostavnih skripti.

Studenti su dužni pripremiti se za laboratorijske vježbe samostalnim rješavanjem niza jednostavnih zadataka. Za zadatke koji obavljaju operacije nad datotekama (pretraživanje sadržaja, promjena imena i slično) potrebno je pripremiti datoteke prikladne za ispitivanje i demonstraciju rada programa.

Preduvjet za obavljanje laboratorijske vježbe je predaja (*upload*) datoteka s rješenjima zadataka kroz sustav Ferko (<https://ferko.fer.hr/ferko/>). Prilikom postavljanja datoteka treba poštovati upute o imenovanju datoteka, kao i o uvjetima pokretanja skripti (navođenje parametara i slično).

U terminu laboratorijskih vježbi student piše kratku provjeru znanja (blic), samostalno rješava jedan programski zadatak, te brani svoja rješenja zadataka za pripremu pred asistentom.

1.1 Protokol odrade laboratorijske vježbe

Ukratko, protokol odrade laboratorijske vježbe je:

1. student je dužan riješiti postavljene zadatke i datoteke s rješenjima predati kroz sustav Ferko;
2. student je dužan doći na laboratorijske vježbe u svom terminu;
3. na početku termina laboratorijskih vježbi studenti pišu kratku provjeru znanja (kviz-pitanja);
4. tijekom laboratorijskog termina studenti dobivaju zadatak koji trebaju riješiti na licu mjesta (izlazni test). Zadatak se rješava na računalu, te se rješenje predaje kroz sustav Ferko.
Obavezno zaključati predaju!
5. tijekom laboratorijskog termina student treba pokazati svoja rješenja zadataka za pripremu asistentu, za što ga asistent ocjenjuje.

1.2 Resursi u laboratoriju

Vježbe se obavljaju u fakultetskim laboratorijima. Na računalima je instalirana okolina Cygwin, s potrebnim alatima (*bash*, *sed*, *grep*,...). Također je instaliran i VMware Player, što omogućuje pokretanje Linux-a s ISO slike i rad na virtualnom računalu.

Zadatak 1

Upoznavanje s varijablama ljuste, te širenjem u naredbenom retku.

Uputa za upload: Naredbe svih podzadataka zapisati u jednu datoteku imena `zadatak1.sh`.

- Postaviti varijablu ljuste `proba` na vrijednost "Ovo je proba".
- Ispisati vrijednost postavljene varijable.
- U varijablu `lista_datoteka` upisati listu svih datoteka tekućeg kazala. Pritom iskoristiti širenje imena datoteke. Ispisati rezultat.
- U varijablu ljuste `proba3` upisati 3 puta nadovezanu vrijednost varijable `proba`, pri čemu na svaku rečenicu treba dodati točku i razmak.
- Varijablu `a` postaviti na vrijednost 4, varijablu `b` na 3, varijablu `c` na 7. Zatim u varijablu `d` upisati vrijednost koja se dobije izrazom $(a + 4) * b \% c$. Upotrijebiti širenje aritmetičkih izraza.
- U varijablu `broj_rijeci` upisati ukupan broj riječi u `.txt` datotekama tekućeg kazala. Upotrijebiti supstituciju naredbe i naredbu `wc`.
- Ispisati sadržaj Vašeg matičnog kazala primjenom *tilda*-ekspanzije.
- Primjenom naredbe `cut` prikazati (samo) informaciju o korisničkom imenu, matičnom kazalu i login ljusti svakog pojedinog korisnika pročitano iz datoteke `/etc/passwd`.
- Proučiti ispis naredbe `ps`. Primjenom naredbe `cut` izrezati dio ispisa koji se odnosi na prikaz identifikatora korisnika (UID), identifikatora procesa (PID), te naredbe koja je pokrenula proces.

Zadatak 2

Naredba `grep`, regularni izrazi, naredba `find`. Programske petlje.

Uputa za upload: Naredbe svih podzadataka zapisati u jednu datoteku imena `zadatak2.sh`.

- Napisati naredbu `grep` koja će u datoteci `namirnice.txt` pronaći i ispisati sve retke u kojima se pojavljuju nazivi voća (*banana, jabuka, jagoda, dinja, lubenica*), neovisno o tome jesu li napisani velikim ili malim slovima.
- Modificirati naredbu tako da se ispisuju samo retci u kojima se ne nalaze zadane riječi. Izlaz zapisati u datoteku `ne-voce.txt`
- Napisati naredbu `grep` koja će u kazalu `~/projekti/` i svim njegovim podkazalima pronaći datoteke u kojima se pojavljuje šifra u obliku tri velika slova i šesteroznamenasti broj, te ispisati retke u kojima se ta šifra pojavljuje.

- Napisati naredbu koja će ispisati imena i detaljne podatke svih datoteka u tekućem kazalu i njegovim podkazalima, koje su mijenjane prije 7 do 14 dana.
- Napisati u jednom retku `for` petlju koja će ispisati brojeve od 1 do 15. Pritom iskoristiti izraz za generiranje sekvence ili naredbu `seq`.
- Napisati `for` petlju koja će se izvršiti za svaku datoteku tekućeg kazala i ispisati redni broj datoteke, njeno ime i broj redaka, te u nastavku prva tri retka same datoteke. Primjer ispisa:

```
1: datoteka1.txt ... 27 redaka
-----
Ovo je prvi redak datoteke.
Ovo je drugi redak.
Treci redak ove datoteke.

2: datoteka2.txt ... 64 redaka
-----
Ovo je prvi redak datoteke 2.
...
```

Zadatak 3

U repozitoriju na stranici predmeta nalaze se dvije *log*-datoteke jednog web poslužitelja. To su tekst datoteke koje se generiraju svakoga dana, a datum je sadržan u imenu datoteke. Svaki redak odgovara jednom pristupanju poslužitelju.

Napisati skriptu koja će proći kroz sve datoteke generirane tijekom veljače, te ispisati podatke o akcijama koje su obavljane. Za svaku datoteku treba zapisati datum, a zatim za svaku akciju koja se pojavljuje u logovima treba ispisati koliko se puta dogodila toga dana. Podatke o akcijama sortirati prema silaznom broju ponavljanja, broj ponavljanja ispisati prije same akcije. Primjer ispisa je dan u nastavku (datum je u formatu `dd-mm-gggg`):

```
datum: 24-02-2008
-----
693 : GET /burza/b/Main.action HTTP/1.1
603 : GET /favicon.ico HTTP/1.1
567 : GET /burza/css/default.css HTTP/1.1
319 : POST /burza/b/Login.action HTTP/1.1
287 : GET /burza/b/Login.action HTTP/1.1
163 : GET / HTTP/1.1
155 : GET /zadaci-faq.html HTTP/1.1
...
```

Uputa za upload: Skriptu nazvati `zadatak3.sh`.

Zadatak 4

U jednom podkazalu tekućeg kazala nalaze se fotografije prebačene s fotoaparata. Datum izmjene odgovara datumu snimanja fotografije. Treba napisati skriptu koja će prebaciti fotografije u kazala po mjesecima (imena odredišnih kazala će biti oblika gggg-mm). Skripti se kao argumenti naredbenog retka predaju imena izvornog kazala sa slikama, te kazala unutar kojeg će se kreirati podkazala po mjesecima. Skripta treba provjeravati postoji li već kazalo odgovarajućeg imena, kreirati ga po potrebi, te premjestiti svaku datoteku u odgovarajuće kazalo. Za dobivanje datuma izmjene datoteke može se iskoristiti naredba `stat`. Primjer izvođenja:

```
$ ls -R *
po_mjesecima:

slike:
IMG_2038.jpg  IMG_2040.jpg  IMG_2042.jpg  IMG_2044.jpg  IMG_2046.jpg
IMG_2039.jpg  IMG_2041.jpg  IMG_2043.jpg  IMG_2045.jpg  IMG_2047.jpg

$ ../zadatak4.sh slike po_mjesecima
$ ls -R *
po_mjesecima:
2009-08  2009-09

po_mjesecima/2009-08:
IMG_2038.jpg  IMG_2039.jpg  IMG_2040.jpg  IMG_2041.jpg  IMG_2042.jpg

po_mjesecima/2009-09:
IMG_2043.jpg  IMG_2044.jpg  IMG_2045.jpg  IMG_2046.jpg  IMG_2047.jpg

slike:
```

Uputa za upload: Skriptu nazvati `zadatak4.sh`, kao argumente predati imena izvornog i odredišnog kazala, u skripti obaviti potrebne provjere.

Zadatak 5

U jednoj datoteci imamo zapisane podatke o rezultatima studenata na nekom predmetu, u obliku:

```
JMBAG; Prezime, Ime; Broj_bodova_1; grupa
```

U drugoj datoteci su podaci o jednoj skupini studenata, u obliku (<TAB> predstavlja oznaku za tabulator):

```
JMBAG<TAB>Broj_bodova_2
```

Napisati skriptu koja će nadopuniti podatke iz druge datoteke podacima iz prve, tako da se dobije ispis na standardni izlaz oblika:

```
JMBAG<TAB>Prezime Ime (grupa)<TAB>Broj_bodova_1<TAB>Broj_bodova_2
```

Ako u prvoj datoteci ne postoje podaci za nekog studenta iz druge, podatke koji nedostaju označiti s "--".

Uputa za upload: Skriptu nazvati `zadatak5.sh`, a kao argumente predati imena dviju tekst datoteka s podacima o studentima. U skripti obaviti potrebne provjere.

Zadatak 6

Napisati skriptu imena `zadatak6.sh` koja će obaviti dvosmjernu sinkronizaciju dva direktorija koja sadrže isključivo datoteke. Pod dvosmjernom sinkronizacijom podrazumijeva se kopiranje novih ili novijih datoteka iz jednog u drugi direktorij pri čemu su direktoriji ravnopravni. Dodatne napomene:

- poruke o greškama moraju se slati na standardni izlaz za greške
- u slučaju datoteka koje se ne mogu pročitati, skripta treba ispisati poruku o grešci ali nastaviti s izvršavanjem
- za svaku datoteku koja se kopira ispisati sljedeći niz (umjesto `dir1` i `dir2` mora pisati ime stvarnih direktorija, a umjesto `dat1` ime datoteke koja se kopira):
`dir1/dat1 -> dir2`

Primjer pozivanja i izvršavanja skripte:

```
$ ./sync.sh dir1 dir2
dir1/a -> dir2
dir1/b -> dir2
dir1/c -> dir2
dir2/f -> dir1
dir2/g -> dir1
dir2/h -> dir1
```

U navedenom primjeru iz direktorija `dir1` prekopirane su datoteke `a`, `b` i `c` u direktorij `dir2`, dok su istovremeno iz direktorija `dir2` kopirane datoteke `f`, `g` i `h` u direktorij `dir1`. Uputa: za informaciju o vremenu izmjene datoteke može se iskoristiti naredba `stat` (obratiti pažnju na opciju `-c %Y`).

Uputa za upload: Skriptu nazvati `zadatak6.sh`, a kao argumente predati imena direktorija koji se sinkroniziraju. U skripti obaviti potrebne provjere.