

Skriptni jezici – zadaci za 1. ciklus laboratorijskih vježbi

ožujak 2023.

1 Uvod

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

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 na računalu, 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 (voditi računa o roku za predaju!);
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*,...). Student vježbu može odraditi i na vlastitom računalu, ali treba voditi računa o mogućnosti povezivanja na mrežu (radi predaje zadatka kroz Ferko).

Napomena: Ako se koristi Windows editor, potrebno je postaviti oznaku kraja retka u skladu s Unixom (EOL = Line Feed), inače su mogući problemi s izvođenjem *bash* skripti.

Zadatak 1

Upoznavanje s varijablama ljsuke, te širenjem u naredbenom retku.

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

- Postaviti varijablu `ljsuke_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 `ljsuke_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. Ispisati vrijednosti varijabli `a`, `b`, `c` i `d`.
- 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.

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 prethodnu naredbu tako da se ispisuju samo retci u kojima se ne pojavljuju zadane riječi.
- 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. Šifra od ostalog teksta mora biti odvojena razmakom.
- 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`.
- Modificirati prethodnu naredbu tako da se zadnji broj raspona zadaje u varijabli `kraj`. Provjeriti je li svejedno koristi li se izraz za generiranje sekvence ili naredba `seq`.

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 dan je u nastavku (datum je u formatu dd-mm-gggg).

Skripta kao argument naredbenog retka prima ime direktorija u kojemu se nalaze log-datoteke (direktorij može biti na proizvoljnoj lokaciji). Skripta treba provjeriti postoji li zadani direktorij te u slučaju pogreške ispisati uputu o načinu korištenja i izaći iz skripte.

Primjer izvršavanja skripte:

```
$ ./zadatak3.sh ../TestPrimjeri/
```

```
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`, kao argument predati ime direktorija s log-datotekama, u skripti obaviti potrebne provjere.

Zadatak 4

U jednom direktoriju (na proizvoljnoj lokaciji) nalaze se fotografije prebačene s fotoaparata. Ime svake datoteke sadrži podatak o datumu i vremenu snimanja fotografije, u obliku GGGGMMDD_HHMM.jpg. Treba napisati skriptu koja će izlistati imena fotografija iz direktorija zadanog kao argument naredbenog retka na način da se fotografije grupiraju prema mjesecu u kojem su snimljene. Fotografije u svakoj grupi trebaju biti poredane po vremenu snimanja i numerirane, a na kraju svake grupe treba ispisati broj fotografija u toj grupi. Ispis oblikovati prema primjeru u nastavku.

Skripta kao argument naredbenog retka prima ime direktorija u kojemu se nalaze fotografije. Skripta treba provjeriti postoji li zadani direktorij, te u slučaju pogreške ispisati uputu o načinu korištenja i izaći iz skripte.

Za testiranje skripte može se koristiti direktorij s “fotografijama”, koji je zapakiran u `tar` arhivu koju možete naći u repozitoriju na stranici predmeta. Arhivu možete raspakirati u tekući direktorij naredbom:

```
tar -xvf testovi.tar
```

Primjer izvršavanja skripte:

```
$ ./zadatak4.sh ../Testovi/Slike/
```

```
01-2020 :
```

```
-----
```

1. 20200102_1025.jpg
2. 20200102_1915.jpg
3. 20200105_1439.jpg
4. 20200107_1859.jpg
5. 20200116_1538.jpg
6. 20200124_1129.jpg
7. 20200125_1327.jpg
8. 20200126_1013.jpg
9. 20200126_1236.jpg

```
--- Ukupno: 9 slika ----
```

```
02-2020 :
```

```
-----
```

1. 20200207_1706.jpg
2. 20200212_0954.jpg
3. 20200214_1234.jpg
4. 20200219_1541.jpg
5. 20200221_1134.jpg
6. 20200222_1609.jpg

```
--- Ukupno: 6 slika ----
```

```
...
```

Uputa za upload: Skriptu nazvati `zadatak4.sh`, kao argument predati ime direktorija s fotografijama, u skripti obaviti potrebne provjere.

Zadatak 5

Napisati skriptu koja će počevši od kazala čije je ime zadano kao argument u naredbenom retku, obići sva podkazala i izbrojati ukupan broj redaka u datotekama s imenima čiji je oblik (npr. `*.c`) zadani kao drugi argument. Na početku skripte ispisati argumente navedene u naredbenom retku pri pozivu skripte.

Uputa za upload: Skriptu nazvati `zadatak5.sh`, a kao argumente predati ime kazala i uzorak imena datoteka.

Zadatak 6

Napisati skriptu prikladnu za dvosmjernu sinkronizaciju dva direktorija koji se zadaju kao argumenti naredbenog retka. Radi lakšeg testiranja skripta treba samo oponašati sinkronizaciju, pri čemu se umjesto stvarnog kopiranja datoteka ispisuju poruke o smjeru kopiranja. Pod dvosmjernom sinkronizacijom podrazumijeva se kopiranje novih ili novijih datoteka iz jednog u drugi direktorij, pri čemu su direktoriji ravnopravni. Pretpostaviti da se u direktorijima nalaze isključivo datoteke (nema poddirektorija).

Za ispitivanje je li prva datoteka novija (kasnije mijenjana) od druge može se koristiti naredba `test` s operatorom `-nt` (engl. *newer than*), npr. `[file1 -nt file2]`.

Skripta kao argumente naredbenog retka prima imena dva direktorija koje treba sinkronizirati. Treba provjeriti postoje li zadani direktoriji, te u slučaju pogreške ispisati uputu o načinu korištenja i izaći iz skripte.

Za testiranje skripte mogu se koristiti direktoriji `dir1` i `dir2`, zapakirani u `tar` arhivu koja je opisana uz zadatak 4.

Primjer izvršavanja skripte:

```
$ ./zadatak6.sh ../Testovi/dir1 ../Testovi/dir2
../Testovi/dir1/aaa --> ../Testovi/dir2
../Testovi/dir1/bbb --> ../Testovi/dir2
../Testovi/dir1/fff --> ../Testovi/dir2
../Testovi/dir2/ccc --> ../Testovi/dir1
../Testovi/dir2/ggg --> ../Testovi/dir1
../Testovi/dir2/hhh --> ../Testovi/dir1
...
```

Uputa za upload: Skriptu nazvati `zadatak6.sh`, a kao argumente predati imena dva direktorija koje treba sinkronizirati. U skripti obaviti potrebne provjere.