

Programski alati
na Unix računalima

* laboratorijske vježbe *

Nikola Grgić

Split, 2019.

Uvod

Kolegij Programski alati na Unix računalima izvodi se na završnoj godini pred-diplomskog stručnog studija Informacijskih tehnologija na Sveučilišnom odjelu za stručne studije Sveučilišta u Splitu.

Laboratorijske vježbe podijeljene su u dvije cjeline. Prvih šest vježbi pokriva osnove rada u Linux ljuski: rad s datotekama, korisnička prava, usmjeravanje ulaza i izlaza, cjevovode (engl. *pipeline*), proširivanje zagradama i kontrolu procesa. Kroz drugi dio vježbi rješavaju se zadatci iz regularnih izraza, Bash skripti i programskog jezika C. U ovom, „programerskom” dijelu kolegija, naglasak je stavljen na korištenje standardnog ulaza i izlaza te argumente naredbenog retka funkcije `main`. Za prevođenje programa i za kontrolu verzija koriste se alati GCC, `make` i Git. Na kraju obje nastavne cjeline nalazi se primjer kolokvija.

Vježbe su osmišljene tako da postupno navikavaju studente na korištenje terminala koji je kao koncept osmišljen na samim počecima računalstva, a unatoč tome što nije bitno izmijenjen već duže od pola stoljeća, još uvijek predstavlja nezaobilazan alat za svakog programera i naprednog korisnika računala.

Popis laboratorijskih vježbi

| | | |
|----|--------------------------------------|----|
| 1 | Korisnička prava | 3 |
| 2 | Rad s datotekama | 8 |
| 3 | Rad u ljosci, prvi dio | 11 |
| 4 | Rad u ljosci, drugi dio | 14 |
| 5 | Alati vi, nano, wget i tar | 17 |
| 6 | Priprema za kolokvij | 20 |
| 7 | Regularni izrazi | 22 |
| 8 | Prevoditelj GCC | 24 |
| 9 | Bash skripte, prvi dio | 27 |
| 10 | Bash skripte, drugi dio | 29 |
| 11 | Git | 31 |
| 12 | Priprema za kolokvij | 33 |

Korisnička prava

Vježba 1

Zadatci u ovoj vježbi rade se pod različitim korisničkim imenima, pri čemu svaki korisnik ima svoj *home* direktorij. Prije nego krenete s rješavanjem nekog zadatka, budite sigurni da znate koji je tekući direktorij (koristite naredbu `pwd`). Sve dok vam ne bude jasna razlika između sljedećih naredbi i putanja, nemojte započinjati vježbu:

```
$ ls /vjezba
```

```
$ ls vjezba
```

```
$ ls ~/vjezba
```

```
$ ls /home/foo/vjezba
```

```
$ ls /home/bar/vjezba
```

- Iz terminala napravite direktorij `/vjezba`. Prema potrebi koristite administratorske ovlasti (naredba `sudo`):

```
$ mkdir /vjezba
```

- Ispišite naziv trenutnog korisnika naredbom `whoami`. Napravite dva nova korisnika (`foo`, `bar`):

```
$ sudo useradd -m foo
```

```
$ sudo useradd -m bar
```

- Postavite korisnicima pripadne lozinke (neka budu iste kao korisničko ime):

```
$ sudo passwd foo
```

```
$ sudo passwd bar
```

- Ispišite datoteku `/etc/passwd` i pronađite zapise vezane uz nove korisnike. Uočite brojeve koji označavaju njihov *user id* i *group id*:

```
$ cat /etc/passwd
```

- Otvorite dva nova terminala i u svakom pokrenite ljsku s drugim korisnikom:

```
$ su -l foo
```

```
$ su -l bar
```

Sada bi trebali imati tri otvorena terminalna prozora za tri postojeća korisnika od kojih samo jedan može koristiti *root* ovlasti i naredbu `sudo` (koji?). Pažljivo rasporedite prozore po ekranu. U svakom prozoru isprobajte naredbu `whoami`.

- Provjerite postoji li direktorij `/vjezba` (u *root* direktoriju). Za svakog korisnika **iz njegovog terminala** napravite direktorij naziva `/vjezba/foo`, tj. `/vjezba/bar`. Korisnici neće moći napraviti direktorije, stoga kao korisnik s *root* ovlastima prvo trebate namjestiti dozvole nad direktorijem `/vjezba` tako da svim korisnicima dozvolite obavljanje svih radnji.
- Koristeći naredbu `ls` uvjerite se da je svaki korisnik vlasnik svoga direktorija:

```
$ ls -l /vjezba
```

- Neka korisnici postavite dozvole za svoje direktorije na 700. Direktorij `/vjezba` treba izgledati ovako:

```
drwx----- 2 bar bar 4096 datum vrijeme bar
drwx----- 2 foo foo 4096 datum vrijeme foo
```

- Provjerite može li korisnik ući u „tuđi” direktorij. Pokušajte promijeniti dozvole

na tuđem direktoriju.

- Neka korisnici postave dozvole za svoje direktorije na 775. Dozvole nad direktorijima sada izgledaju ovako:

```
drwxrwxr-x 2 bar bar 4096 datum vrijeme bar
drwxrwxr-x 2 foo foo 4096 datum vrijeme foo
```

Korisnici mogu međusobno ulaziti u direktorije (dozvola `x`, naredba `cd`) i pregledavati njihov sadržaj (dozvola `r`, naredba `ls`).

- Kao korisnik `foo` zapišite poruku u datoteku `/vježba/foo/myfile`:

```
$ echo "I'm Foo, and this is my file" > /vježba/foo/myfile
```

- Kao korisnik `bar` izlistajte sadržaj direktorija `foo` i pročitajte sadržaj datoteke `myfile`. Pokušajte promijeniti sadržaj datoteke `myfile`.

```
$ ls foo
```

```
$ cat foo/myfile
```

```
$ echo "Gimme that file" > /vježba/foo/myfile
```

- Nad datotekom `myfile` postavite dozvole 777. Neka `bar` ponovo pokuša izmijeniti sadržaj datoteke.

- Neka `bar` pokuša izbrisati datoteku `myfile`:

```
$ rm /vježba/foo/myfile
```

Dozvole nad datotekom `myfile` iz njegove perspektive su `rwx`, a nad roditeljskim direktorijem `foo` su `r-x`. Uočite da se dozvola brisanja ne tumači iz dozvola datoteke koja se briše, već iz dozvola roditeljskog direktorija.

- U direktoriju `/vjezba/foo` napravite sljedeće datoteke i direktorije s odgovarajućim dozvolama (naredbe `mkdir` i `touch`):

```
----- 1 foo foo    0 Oct 18 12:17 00
d----- 2 foo foo 4096 Oct 18 12:17 01
-r----- 1 foo foo    0 Oct 18 12:17 02
--w----- 1 foo foo    0 Oct 18 12:17 03
---x----- 1 foo foo    0 Oct 18 12:17 04
----r----- 1 foo foo    0 Oct 18 12:17 05
-----w---- 1 foo foo    0 Oct 18 12:17 06
-----x--- 1 foo foo    0 Oct 18 12:17 07
-----r-- 1 foo foo    0 Oct 18 12:17 08
-----w- 1 foo foo    0 Oct 18 12:17 09
-----x 1 foo foo    0 Oct 18 12:17 10
drwxrwxrwx 2 foo foo 4096 Oct 18 12:17 11
```

- Naredbom `id` provjerite kojim grupama korisnik `foo` pripada. Kao `foo`, pokušajte napraviti direktorij `/foo` koristeći naredbu `sudo mkdir`.
- Kao administrator dodajte korisnika `foo` u grupu `sudo`:

```
$ sudo usermod -a -G sudo foo
```

Kako bi promjena imala efekta, naredbom `exit` iziđite iz aktivne `foo` ljuške. Provjerite kojim grupama `foo` sada pripada. Vratite se u `foo` ljušku (`su -l foo`) i opet isprobajte naredbu `sudo mkdir`.

PREDAVANJE VJEŽBE

Snimite povijest naredbi za svakog korisnika:

```
$ history > /vjezba/username_history
```

Spojite sve tri dobivene datoteke u jednu koristeći naredbu `cat`, tako da izlazna datoteka ima naziv u obliku `grupa_ime-prezime_vjezba`:

```
$ cat file1 file2 file3 > xx_ime-prezime_yy
```

Datoteku pošaljite elektroničkom poštom na adresu `predajem.unix@oss.unist.hr` i pričekajte odgovor. Poruka neka bude bez teksta, s predmetom (engl. *subject*) `vjezba`.

SLANJE ZADATAKA OD KUĆE

Vježba koja nije završena u laboratoriju, može se ponovo odraditi kod kuće i poslati na ocjenjivanje. Vježba se može poslati u roku od 24 sata nakon završetka vježbi, u suprotnom je treba odraditi na nadoknadama. Postupak slanja vježbe opisan je u dokumentu „Upute za rad u laboratoriju” koji je dostupan na sustavu Moodle.

Zadaci u vježbama označeni su numeracijom oblika `zadX`. Za svaki zadatak treba napraviti odvojeni direktorij naziva `~/zadX` koji će biti tekući direktorij za taj zadatak. Ovo vrijedi za sve vježbe, kao i za kolokvij i ispit.

zad1 Napravite direktorij `~/zad1` i pozicionirajte se u njega. Ispišite tekući direktorij i putanju do svog *home* direktorija:

```
$ pwd
```

```
$ echo ~
```

Izlistajte sadržaj *home* direktorija:

```
$ ls ~
```

```
$ ls -a ~
```

```
$ ls -l ~
```

```
$ ls -al ~
```

Napravite direktorij `~/zad1/vjezba`. Isprobajte program `tree`: izlistajte sadržaj *root* direktorija, svog *home* direktorija i tekućeg direktorija. Isprobajte naredbe `pwd`, `ls`, `mkdir`, `rm` i `touch`.

Predajte zadatak koristeći naredbu `predajem`:

```
$ predajem zad1
```

Na isti način predat ćete i ostale zadatke. Prije nego nastavite s vježbom, proučite dokument s uputama za vježbe koji je dostupan na sustavu Moodle.

- zad2** Napravite direktorij `~/zadX` (ubuduće se ovo podrazumijeva). U njemu napravite pomoćne direktorije i datoteke koje će vam omogućiti da isprobate sljedeće mogućnosti naredbe `mv`: preimenovanje datoteka, preimenovanje direktorija, premještanje datoteke u drugi direktorij te premještanje nekoliko datoteka istovremeno u drugi direktorij.
- zad3** U tekućem direktoriju (!) napravite pomoćne direktorije i datoteke koje će vam omogućiti da isprobate sljedeće mogućnosti naredbe `cp`: kopiranje datoteka, kopiranje direktorija, kopiranje direktorija koji imaju nekakav sadržaj, kopiranje datoteke u drugi direktorij i kopiranje nekoliko datoteka istovremeno u drugi direktorij.
- zad4** Napravite direktorij naziva `My Documents`. Promijenite naziv tog direktorija u `My old documents`.
- zad5** Napravite direktorij `tmp`. Unutar njega napravite skrivenu datoteku naziva `.hidden`. Izlistajte sadržaj direktorija pa preimenujte direktorij `tmp` u `temp`. Napravite kopiju direktorija `temp` i nazovite je `temp-copy`. Izbrišite izvorni direktorij `temp`. Izlistajte cjelokupan sadržaj direktorija `temp-copy` i uvjerite se da u njemu postoji datoteka `.hidden`.
- zad6** Napravite prazne datoteke `f1` i `f2`, te direktorije `d1` i `d2`. Kopirajte obje datoteke u `d1`. Promijenite tekući direktorij u `d2` pa napravite lokalne kopije datoteka.
- zad7** Koristeći samo jednu naredbu u liniji (engl. *bash one-liner*, ubuduće */bol/*), napravite pet datoteka naziva `a b c d e`. Napravite direktorij `characters` i kopirajte svih pet datoteka u njega. Napravite backup direktorija `characters` i nazovite ga `backup-yyyyymmdd` (s današnjim datumom u nazivu). Izbrišite direktorij `characters`.
- zad8** Kopirajte datoteku `/materijali/mint_download_links` u tekući direktorij i nazovite je `mint_dl`. */bol/*
- zad9** Kopirajte datoteku `/materijali/mint_download_links` u tekući direk-

torij (pritom nemojte izlaziti iz njega). Prilikom upisivanja putanje koristite se tipkom `TAB` za automatsko nadopunjavanje.

zad10 Napravite deset direktorija naziva 0 1 2 3 4 5 6 7 8 9. */bol/*

POVIJEST NAREDBI

Ako želite sačuvati povijest naredbi, možete je poslati na web servis s *pastebin* funkcionalnosti naredbom `sharehistory`.

PREDAVANJE VJEŽBE

Vježba se završava naredbom `predajem`. Prije toga, svi zadatci pojedinačno trebaju biti riješeni i predani:

```
$ predajem vjezbu
```

SLANJE ZADATAKA OD KUĆE

Vježba koja nije završena u laboratoriju, može se ponovo odraditi kod kuće i poslati na ocjenjivanje. Vježba se može poslati u roku od 24 sata nakon završetka vježbi, u suprotnom je treba odraditi na nadoknadama. Postupak slanja vježbe opisan je u dokumentu „Upute za rad u laboratoriju” koji je dostupan na sustavu Moodle.

zad1 Ispišite na ekran nazive svih datoteka (zajedno s putanjom) iz direktorija `/materijali` čiji nazivi počinju slovom `m`:

```
$ ls /materijali/m*
```

Ispis u jednom stupcu moguće je napraviti uz pomoć opcije `-1`:

```
$ ls -1 /materijali/m*
```

Ako promijenimo tekući direktorij u `/materijali`, bit će ispisani samo nazivi datoteka, bez putanje:

```
$ cd /materijali
```

```
$ ls m*
```

Rezultat zadnjeg ispisa treba upisati u datoteku naziva `m_start`. U tu svrhu predložene su četiri varijante iste naredbe. Isprobajte njihovo korištenje ovisno o direktoriju u kojem se nalazite.

```
$ ls m* > m_start
```

```
$ ls m* > ~/zad1/m_start
```

```
$ ls /materijali/m* > m_start
```

```
$ ls /materijali/m* > ~/zad1/m_start
```

zad2 Kopirajte direktorij `/materijali/letters` u tekući direktorij. Iz kopiranog direktorija izbrišite sve datoteke čiji se naziv sastoji od dva znaka.

- zad3** Nazive svih datoteka (bez putanja) iz direktorija `/materijali` čiji naziv završava na `.log` upišite u datoteku `list`. Provjerite što se dogodi ako umjesto naredbe `ls` u *prompt* upišete `sl`.
- zad4** Nazive svih datoteka iz direktorija `/materijali`, koje bilo gdje u nazivu imaju niz znakova `user`, upišite u datoteku `list`.
- zad5** Nazive svih datoteka čiji naziv ima oblik `/var/log/syslog.X.gz`, pri čemu `X` može biti bilo koji znak, upišite u datoteku `list`.
- zad6** Nazive svih datoteka iz direktorija `/materijali/backup` iz čijeg se naziva može zaključiti da su napravljene u 2018. godini upišite u datoteku `list`.
- zad7** Nazive svih datoteka iz direktorija `/materijali` čiji naziv predstavlja neku godinu iz 21. stoljeća upišite u datoteku `list`.
- zad8** Nazive i putanje svih datoteka iz direktorija `/materijali/20XX` čiji naziv počinje s `backup` upišite u datoteku `list`.
- zad9** Nazive svih datoteka iz direktorija `/materijali/backup` iz čijeg se naziva može zaključiti da su napravljene u studenom ili prosincu (bilo koje godine) upišite u datoteku `list`.
- zad10** Putanje i nazive svih datoteka naziva `list` koje su napravljene u sklopu prethodnih zadataka upišite u datoteku `list_all`.
- zad11** Sadržaj svih datoteka naziva `list` koje su napravljene u sklopu prethodnih zadataka upišite u datoteku `list_all_content`.

POVIJEST NAREDBI

Ako želite sačuvati povijest naredbi, možete je poslati na web servis s *pastebin* funkcionalnosti naredbom `sharehistory`.

PREDAVANJE VJEŽBE

Vježba se završava naredbom `predajem`. Prije toga, svi zadatci pojedinačno trebaju biti riješeni i predani:

`$ predajem vjezbu`

SLANJE ZADATAKA OD KUĆE

Vježba koja nije završena u laboratoriju, može se ponovo odraditi kod kuće i poslati na ocjenjivanje. Vježba se može poslati u roku od 24 sata nakon završetka vježbi, u suprotnom je treba odraditi na nadoknadama. Postupak slanja vježbe opisan je u dokumentu „Upute za rad u laboratoriju” koji je dostupan na sustavu Moodle.

Rad u ljusti, drugi dio

Vježba 4



- zad1** Sadržaj datoteke `/materijali/towns` poredajte po abecedi i spremite u datoteku `towns_sorted`. */bol/*
- zad2** Dohvatite zadnja dva retka iz datoteke `/materijali/towns` i spremite ih u `towns_last`. */bol/*
- zad3** Dohvatite treći redak po abecedi iz datoteke `/materijali/towns` i spremite ga u datoteku `towns_third`. */bol/*
- zad4** Koristeći naredbu `wc` prebrojite linije, riječi i znakove (`man wc: „newline, words and byte count”`) u svim datotekama iz direktorija `/materijali` čiji naziv počinje s `md5` i spremite izlaz u datoteku `count`. */bol/*
- zad5** Prebrojite linije, riječi i znakove iz datoteke `/materijali/towns` i spremite ih u datoteke `lines`, `words` i `bytes`. U datotekama trebaju biti spremljeni samo brojevi, bez putanja.
- zad6** Broj znakova koje ispisuje naredba `pi` upišite u datoteku `decimals_count`. */bol/*
- zad7** Izlaz naredbe `fortune` usmjerite u naredbu `cowsay`. ASCII crtež zapišite u datoteku `smart_cow`. */bol/*

U slučaju da naredbe iz prethodnog zadatka nisu dostupne na sustavu, treba ih instalirati:

```
$ sudo apt-get install cowsay fortune-mod fortune-min
```

zad8 Naredba `cowsay` može generirati različite likove:

```
$ cowsay -f tux "Linux rules"
```

```
$ cowsay -l
```

Popis datoteka sa slikama životinja iz direktorija `/usr/share/cowsay/cows` zapišite u datoteku `animals`. Zapišite samo datoteke čiji naziv završava s `.cow`.

zad9 Upišite trenutni datum i vrijeme (izlaz naredbe `date`) u datoteku `local_time`.

zad10 U tekući direktorij kopirajte rješenje iz prethodnog zadatka pa na kraj datoteke `local_time` dodajte crtež stegosaurusa bez da prebrišete postojeći sadržaj. */bol/*

zad11 U tekući direktorij kopirajte rješenje iz prethodnog zadatka pa postavite dozvole nad datotekom `local_time` tako da je može čitati samo vlasnik.

zad12 Koristeći proširivanje zagradama iscertajte kravu koja broji od 1 do 105 i spremite je u datoteku `math_cow`. */bol/*

zad13 Napravite deset direktorija naziva `dirX` (X je jednoznamenasti broj). */bol/*

zad14 Napravite direktorij `alphabet`. Bez promjene tekućeg direktorija napravite u njemu 26 datoteka s nazivima duljine jedan znak (sva velika slova engleske abecede). */bol/*

zad15 Napravite datoteke čiji nazivi su duljine tri znaka i predstavljaju sve kombinacije znakova `1, I, l` (broj 1, veliko slovo i, malo slovo L). */bol/*

- zad16** Ispišite na ekran brojeve od 0 do 100. Listu usmjerite u `wc` i odredite njenu duljinu (prebrojite članove liste). Sve napravite kao `/bol/`, a rezultat usmjerite u datoteku `num_count`.
- zad17** Koristeći samo jedan izraz za proširivanje zagrada napravite direktorije za svaki mjesec u tekućoj i sljedeće dvije godine (nazivi direktorija neka budu oblika `YYYY-MM`).
- zad18** Pokrenite program `cmatrix` pa ga pošaljite u pozadinu kombinacijom tipki `Ctrl Z`. Zapišite trenutni datum i vrijeme u datoteku `lab_end`. Vratite `cmatrix` u prvi plan naredbom `fg`. Prekinite izvođenje programa tipkama `Ctrl C`.

POVIJEST NAREDBI

Ako želite sačuvati povijest naredbi, možete je poslati na web servis s *pastebin* funkcionalnosti naredbom `sharehistory`.

PREDAVANJE VJEŽBE

Vježba se završava naredbom `predajem`. Prije toga, svi zadatci pojedinačno trebaju biti riješeni i predani:

```
$ predajem vjezbu
```

SLANJE ZADATAKA OD KUĆE

Vježba koja nije završena u laboratoriju, može se ponovo odraditi kod kuće i poslati na ocjenjivanje. Vježba se može poslati u roku od 24 sata nakon završetka vježbi, u suprotnom je treba odraditi na nadoknadama. Postupak slanja vježbe opisan je u dokumentu „Upute za rad u laboratoriju” koji je dostupan na sustavu Moodle.

Alati vi, nano, wget i tar

Vježba 5

zad1 Preuzmite datoteku `personal.oss.unist.hr/~nrgjic/nastava/unix/tutor` i učitajte je u editor `vi`. Pročitajte i odradite upute iz datoteke. Na kraju pregledajte sadržaj izmijenjene datoteke programom `less` (pomicanje po tekstu najlakše je tipkama `↑` i `↓` za up/down) .

zad2 Napišite sljedeći tekst u editoru `nano` i sačuvajte ga u datoteci `program`:

```
INTEGER MAIN()
[
  INTEGER first, second

  printf "Enter two numbers:"
  scanf "%d %d", &first, &second

  printf "%d", first + second
]
```

zad3 Kopirajte datoteku iz prethodnog zadatka. Koristeći editor `vi` prepravite programski kôd tako da zadovoljava sintaksu programskog jezika C i da bude uredno uvučen:

```
int main()
{
    int first, second;

    printf ("Enter two numbers:");
    scanf ("%d %d", &first, &second);

    printf ("%d", first + second);
}
```

- zad4** Napravite tri direktorija naziva `dir1` do `dir3` */bol/*. U svakom direktoriju napravite po pet datoteka naziva `file08` do `file12` */bol/*. Zapakirajte sve u arhivu `archive.tar.gz` pa obrišite izvorne direktorije. Spremite sadržaj arhive (listu datoteka) u datoteku `archive_list`.
- zad5** Ostanite u tekućem direktoriju pa zapakirajte datoteku `tutor` iz prvog zadatka u `tutor.tar.gz`. Poruka "Removing leading ..." koju program `tar` možda ispiše nije znak da je došlo do pogreške, već informacija o tome da su spremljene relativne putanje.
- zad6** Koristeći putanju koja počinje oznakom za korisnikov *home* direktorij (`~`) u putanji, raspakirajte datoteku `tutor.tar.gz` iz prethodnog zadatka u tekući direktorij.
- zad7** Raspakirajte datoteku `/materijali/logs1602.tar.gz` u tekući direktorij. Od dobivenih datoteka napravite novu arhivu naziva `logs.tar.gz` u kojoj će biti samo datoteke s nastavkom `.log`. */bol/*
- zad8** Zapakirajte sve direktorije iz `/materijali/atmusers` čiji se naziv sastoji od točno 7 znakova u datoteku `atmusers.tar.gz`.
- zad9** U tekući direktorij kopirajte arhive iz svih prethodnih zadataka */bol/*. Upute: sve arhive imaju isti nastavak i nalaze se u direktorijima čiji naziv ima standardan oblik. Kod kopiranja treba provući njihove putanje kroz *wildcard* uzorak.
- zad10** Izlistajte sadržaj arhive `atmusers.tar.gz` bez da je kopirate u tekući direktorij. Izlaz spremite u datoteku `archive_contents`.

POVIJEST NAREDBI

Ako želite sačuvati povijest naredbi, možete je poslati na web servis s *pastebin* funkcionalnosti naredbom `sharehistory`.

PREDAVANJE VJEŽBE

Vježba se završava naredbom `predajem`. Prije toga, svi zadatci pojedinačno trebaju biti riješeni i predani:

`$ predajem vjezbu`

SLANJE ZADATAKA OD KUĆE

Vježba koja nije završena u laboratoriju, može se ponovo odraditi kod kuće i poslati na ocjenjivanje. Vježba se može poslati u roku od 24 sata nakon završetka vježbi, u suprotnom je treba odraditi na nadoknadama. Postupak slanja vježbe opisan je u dokumentu „Upute za rad u laboratoriju” koji je dostupan na sustavu Moodle.

Prije pristupanja kolokviju studenti se moraju upoznati s pravilima za pisanje ispita koji su dostupni na sustavu Moodle ili kod nastavnika.

VREMENSKO OGRANIČENJE: prvi zadatak treba riješiti i predati u prvih 10 minuta ispita.

zad1 Napišite sljedeći program koristeći editor `vi` i snimite ga pod nazivom `sample1.c`

```
int main() {
    printf("*****\n");
    printf("*   TITLE   *\n");
    printf("*****\n");
    printf("\n");
    printf("Sample output...\n");
    return 0;
}
```

Koristeći editor `vi` izmijenite pozive funkcije `printf` s pozivima funkcije `puts`. Sačuvate izmijenjenu datoteku pod nazivom `sample2.c`. (6/30)

zad2 Kopirajte direktorij `/materijali/atmusers` zajedno s poddirektorijima i datotekama u tekući direktorij. Nad kopiranim poddirektorijima čiji naziv ima duljinu 7 znakova postavite dozvole tako da korisnik i grupa mogu čitati i pisati, a da ostali nemaju nikakva prava. (6/30)

zad3 Dohvatite datoteku `http://personal.oss.unist.hr/~nrggic/nastava/unix/movies.tar.gz` s interneta i raspakirajte je u tekući direktorij. Napravite direktorij naziva `1980s`. Pomaknite sve filmove snimljene 1980-ih u taj direktorij. */bol/*(6/30)

zad4 Kopirajte datoteku `movies.tar.gz` iz prethodnog zadatka ili je skinite s interneta i raspakirajte je. Prebrojite sve `mkv` filmove snimljene 1990-ih i rezultat (samo broj) spremite u datoteku `1990_count`. */bol/(6/30)*

zad5 Iz datoteke `/materijali/movies.lst` dohvatite petu liniju i zapišite je u datoteku `fifth_movie` (6/30)

Predajte ispit (predajem ispit) i ispunite tablicu:

| | |
|----------------|------------------|
| Ime i prezime: | Predani zadatci: |
|----------------|------------------|

Regularni izrazi

Vježba 7

U većini zadataka u vježbi koristit će se dvije datoteke (`web_access.log` i `words`). Kako bi ubrzali rješavanje zadataka, putanje do tih datoteka spremite u varijable naziva `log` i `words`:

```
$ log=/materijali/web_access.log
```

```
$ words=/usr/share/dict/words
```

Te dvije varijable koristit ćemo za brže pristupanje pojedinim datotekama, npr.:

```
$ echo $log
```

```
$ grep something $words
```

Zadaci se rješavaju tako da se prvi isproba regularni izraz (RI), a nakon toga se izlaz RI usmjeri u datoteku naziva `~/zadX/result` (za prvih 9 zadataka). Svi zadatci u vježbi treba rješavati kao */bol/*.

zad1 log: dohvatite retke koji završavaju s `terminated`

zad2 log: prebrojite prazne retke

zad3 log: prebrojite retke koji sadrže `Windows 9x` (x je bilo koja znamenka)

zad4 log: dohvatite retke koji sadrže `eth0` i ispred toga bilo koji znak koji nije razmak

zad5 log: dohvatite adrese koje imaju oblik `192.168.16.[broj]`. Dohvatite

samo adrese, ne cijele retke

- zad6** `words`: dohvatite riječi duljine 8 znakova koji završavaju na `bow`
- zad7** `words`: dohvatite riječi koje imaju tri slova s točkom (slova `i` ili `j`) zaredom. Primjer: riječ `Beijing`
- zad8** `words`: dohvatite riječi koje sadrže 7 ili više suglasnika zaredom. Neka RI isključi samoglasnike i znakove interpunkcije `[[:punct:]]`
- zad9** `words`: dohvatite riječi koje sadrže ponavljajuću sekvencu `pa` koja se ponavlja barem dva puta
- zad10** u tekućem direktoriju napravite izmijenjenu kopiju datoteke `/materijali/towns_spaced` bez praznih redaka `/bol/`

POVIJEST NAREDBI

Ako želite sačuvati povijest naredbi, možete je poslati na web servis s `pastebin` funkcionalnosti naredbom `sharehistory`.

PREDAVANJE VJEŽBE

Vježba se završava naredbom `predajem`. Prije toga, svi zadatci pojedinačno trebaju biti riješeni i predani:

```
$ predajem vjezbu
```

SLANJE ZADATAKA OD KUĆE

Vježba koja nije završena u laboratoriju, može se ponovo odraditi kod kuće i poslati na ocjenjivanje. Vježba se može poslati u roku od 24 sata nakon završetka vježbi, u suprotnom je treba odraditi na nadoknadama. Postupak slanja vježbe opisan je u dokumentu „Upute za rad u laboratoriju” koji je dostupan na sustavu Moodle.

Prilikom rješavanja nekih zadataka u izvorni kôd programa je potrebno uključiti sljedeće datoteke zaglavlja:

```
#include <stdlib.h>
#include <stdio.h>
```

zad1 Napišite program "Hello world" u programskom jeziku C i prevedite ga. Datoteka s izvornim kodom neka se zove `hello.c`, a izvršna datoteka `hello`. Pokrenite program i usmjerite izlaz u datoteku `greeting`.

zad2 Napišite program koji prima korisničko ime kao argument i pozdravlja korisnika. Izlaz programa usmjeriti u datoteku `greeting`. Za ispis koristiti sljedeću funkciju:

```
printf("%s %s\n", "Hello", argv[1])
```

zad3 Napravite datoteku `fmodul.c` s funkcijom `modul(int broj)` koja računa apsolutnu vrijednost argumenta i od nje napravite statičku biblioteku `libmodul.a`:

```
$ gcc -c fmodul.c -o fmodul.o
```

```
$ ar rcs libmodul.a fmodul.o
```

Napomena: na ispitu neće biti zadataka s izradom statičke biblioteke. Nemojte predavati rješenje dok ne riješite sljedeći zadatak i uvjerite se da funkcija radi ispravno.

zad4 Napišite i prevedite program `modul`, s datotekom izvornog koda `modul.c`,

koji ispisuje apsolutnu vrijednost broja primljenog kao parametar (koristite gotovu biblioteku iz prethodnog zadatka). Za pretvorbu stringa u cijeli broj koristite funkciju `int atoi(char *str);`

zad5 Napravite datoteku strukture *makefile* kojom će se automatizirati koraci izrade programa `modul` iz prethodnog zadatka. Datoteke s izvornim kodom (`~/zad4/modul.c` i `~/zad3/fmodul.c`) kopirajte u tekući direktorij. Ciljevi (engl. *target*) u datoteci *makefile* neka budu: `all` (stvara izvršni program naziva `modul`), `fmodul.o` (stvara objektnu datoteku), `libmodul.a` (stvara biblioteku). Napravite i *target* `clean_start` koji će obrisati sve datoteke u direktoriju osim onih potrebnih za prevođenje programa od početka. Rješenje zadatka nalazi se na dnu vježbe (prvo pokušajte samostalno riješiti zadatak).

NAPOMENA

Na ispitu će prilikom rješavanja zadataka vezanih za prevoditelj GCC biti potrebno poznavati osnove programiranja u programskom jeziku C (varijable, osnovne matematičke operacije, odluke, ispis, pozivi i povratna vrijednost funkcije, kao i prosljeđivanje argumenata funkciji `main`).

POVIJEST NAREDBI

Ako želite sačuvati povijest naredbi, možete je poslati na web servis s *pastebin* funkcionalnosti naredbom `sharehistory`.

PREDAVANJE VJEŽBE

Vježba se završava naredbom `predajem`. Prije toga, svi zadatci pojedinačno trebaju biti riješeni i predani:

```
$ predajem vjezbu
```

SLANJE ZADATAKA OD KUĆE

Vježba koja nije završena u laboratoriju, može se ponovo odraditi kod kuće i poslati na ocjenjivanje. Vježba se može poslati u roku od 24 sata nakon završetka vježbi, u suprotnom je treba odraditi na nadoknadama. Postupak slanja vježbe opisan je u dokumentu „Upute za rad u laboratoriju” koji je dostupan na sustavu Moodle.

RJEŠENJE ZADATKA 5

```
all: libmodul.a modul.c  
    gcc modul.c libmodul.a -o modul
```

```
libmodul.a: fmodul.o  
    ar rcs libmodul.a fmodul.o
```

```
fmodul.o: fmodul.c  
    gcc -c fmodul.c -o fmodul.o
```

```
clean_start:  
    rm fmodul.o libmodul.a modul
```

Bash skripte, prvi dio

Vježba 9



- zad1** Napišite Bash skriptu „*Hello world*”.
- zad2** Napišite skriptu koja ispisuje svaki drugi cijeli broj počevši od prvog argumenta (uključivo) do 100.
- zad3** Napišite skriptu `hello_user` koja ispisuje pozdrav korisniku čije ime prima kao argument.
- zad4** Napišite skriptu koja zbraja dva broja primljena kao argument. Koristite C-stil računanja aritmetičkih izraza.
- zad5** Napišite skriptu koja ispisuje zadanu poruku, ali samo u slučajevima kada je pokrenuta s najmanje jednim argumentom:

```
Pokrenuta je skripta naziva X s Y argumenata
```

- zad6** Napišite skriptu koja sprema popis datoteka iz tekućeg direktorija u varijablu koju kasnije koristi za iteraciju u petlji koja broji te datoteke. Neka izlaz izgleda ovako:

```
U tekucem direktoriju nalazi se X datoteka.
```

POVIJEST NAREDBI

Ako želite sačuvati povijest naredbi, možete je poslati na web servis s *pastebin* funkcionalnosti naredbom `sharehistory`.

PREDAVANJE VJEŽBE

Vježba se završava naredbom `predajem`. Prije toga, svi zadatci pojedinačno trebaju biti riješeni i predani:

```
$ predajem vjezbu
```

SLANJE ZADATAKA OD KUĆE

Vježba koja nije završena u laboratoriju, može se ponovo odraditi kod kuće i poslati na ocjenjivanje. Vježba se može poslati u roku od 24 sata nakon završetka vježbi, u suprotnom je treba odraditi na nadoknadama. Postupak slanja vježbe opisan je u dokumentu „Upute za rad u laboratoriju” koji je dostupan na sustavu Moodle.

Bash skripte, drugi dio

Vježba 10

- zad1** Napišite skriptu koja stvara kopije svih datoteka koje se nalaze u tekućem direktoriju. Kopije datoteka neka imaju naziv oblika `stari_naziv.bak`.
- zad2** Napišite skriptu koja prima putanju i ispisuje radi li se o datoteci ili direktoriju.
- zad3** Napišite skriptu koja broji prazne datoteke iz direktorija kojeg prima kao argument. Za testiranje možete koristiti direktorij `/materijali/file_test/`.
- zad4** Napišite skriptu koja ispisuje imena autora knjiga iz datoteke `/materijali/booklist`.
- zad5** Napišite skriptu koja naredbom `ping` provjerava dostupnost računala primljenog kao argument (u obliku URL-a ili IP adrese). Izlaz oblikujte ovako: "Računalo _____ je/nije dostupno". Skripta ne smije prikazivati izlaz naredbe `ping` ni u jednom slučaju pa izlaz prikriti usmjerenjem standardnog izlaza i standardne greške u `/dev/null`.

POVIJEST NAREDBI

Ako želite sačuvati povijest naredbi, možete je poslati na web servis s *pastebin* funkcionalnosti naredbom `sharehistory`.

PREDAVANJE VJEŽBE

Vježba se završava naredbom `predajem`. Prije toga, svi zadatci pojedinačno trebaju biti riješeni i predani:

```
$ predajem vjezbu
```

SLANJE ZADATAKA OD KUĆE

Vježba koja nije završena u laboratoriju, može se ponovo odraditi kod kuće i poslati na ocjenjivanje. Vježba se može poslati u roku od 24 sata nakon završetka vježbi, u suprotnom je treba odraditi na nadoknadama. Postupak slanja vježbe opisan je u dokumentu „Upute za rad u laboratoriju” koji je dostupan na sustavu Moodle.

-
- zad1** Promijenite izvorni kod programa iz repozitorija https://github.com/ossunix/hello_world_template.git tako da ispisuje pozdrav na hrvatskom jeziku. Prevedite program koristeći gotovi `makefile`, pokrenite ga i izlaz programa spremite u datoteku `hello`.
- zad2** U repozitoriju <https://github.com/ossunix/udvostruci.git> nalazi se izvorni kod programa koji ispisuje dvostruku vrijednost broja primljenog kao argument. Program treba prevesti pa napisati bash skriptu koja pomoću njega udvostručuje brojeve iz datoteke `/materijali/brojevi100` i upisuje ih u datoteku `brojevi200`.
- zad3** U repozitoriju https://github.com/ossunix/velika_slova.git se nalazi izvorni kod programa koji traži od korisnika da unese riječ, nakon čega je ispisuje velikim slovima. Potrebno je izmijeniti program tako da korisnik ne unosi ništa, nego da se riječ prosljeđuje programu kao argument. Napisati skriptu koja pomoću tog programa sve riječi iz datoteke `/materijali/animals` zapisuje velikim slovima u datoteku `animals_capitalized`.

POVIJEST NAREDBI

Ako želite sačuvati povijest naredbi, možete je poslati na web servis s *pastebin* funkcionalnosti naredbom `sharehistory`.

PREDAVANJE VJEŽBE

Vježba se završava naredbom `predajem`. Prije toga, svi zadatci pojedinačno trebaju biti riješeni i predani:

`$ predajem vjezbu`

SLANJE ZADATAKA OD KUĆE

Vježba koja nije završena u laboratoriju, može se ponovo odraditi kod kuće i poslati na ocjenjivanje. Vježba se može poslati u roku od 24 sata nakon završetka vježbi, u suprotnom je treba odraditi na nadoknadama. Postupak slanja vježbe opisan je u dokumentu „Upute za rad u laboratoriju” koji je dostupan na sustavu Moodle.

Priprema za kolokvij

Vježba 12

Prije pristupanja kolokviju studenti se moraju upoznati s pravilima za pisanje ispita koji su dostupni na sustavu Moodle ili kod nastavnika.

- zad1** Napravite kopiju datoteke `/materijali/devlog12a` koja će na početku svake linije sadržavati niz od četiri razmaka `/bol/(12/50)`
- zad2** Napravite skriptu u kojoj će se iterirati po linijama datoteke `/materijali/etc-passwd` i koja će ispisati korisnike koji pripadaju grupi 101. Korisnička imena su u prvom, a grupe u četvrtom polju svake linije. Prije predavanja zadatka, izlaz skripte usmjerite u datoteku `users101`. (18/50)
- zad3** U repozitoriju <https://github.com/ossunix/briskula.git> nalazi se statička biblioteka s funkcijom prototipa `void karta();` koja ispisuje slučajnu kartu za igru briškula. Napišite program u C-u koji kao argument prima broj karata koje treba ispisati. Napišite `makefile` za automatizaciju postupka prevođenja programa. Na kraju „ručno” generirajte 40 karata (mogu se ponavljati) i upišite ih u datoteku `karte`. Napomena: za pretvaranje vrijednosti stringa u cijeli broj koristite funkciju `int atoi(char *str)` iz biblioteke `stdlib.h`. (20/50)

Predajte ispit (`predajem ispit`) i ispunite tablicu:

| | |
|----------------|------------------|
| Ime i prezime: | Predani zadatci: |
|----------------|------------------|