

# Operativni sustavi

Vježbe

Ljiljana Despalatović

Copyright © 2017 Ljiljana Despalatović

Licensed under the Creative Commons Attribution 4.0 International (the “License”). You may not use this file except in compliance with the License. You may obtain a copy of the License at <https://creativecommons.org/licenses/by/4.0/>. Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

# Sadržaj

<b>1</b>	<b>Uvod .....</b>	<b>5</b>
<b>2</b>	<b>Vježba 1 .....</b>	<b>7</b>
2.1	Instaliranje Linux operativnog sustava na virtualnu mašinu	7
<b>3</b>	<b>Vježba 2 .....</b>	<b>9</b>
3.1	Command line interface (CLI)	9
3.2	Naredbe Linux operativnog sustava	10
3.3	Korištenje repozitorija	11
3.4	Kompajliranje programa za Linux platformu	12
<b>4</b>	<b>Vježba 3 .....</b>	<b>15</b>
4.1	Naredbe za navigaciju file systemom	15
4.2	Putevi <i>paths</i>	17
4.3	Direktorij <code>/proc</code>	18

<b>5</b>	<b>Vježba 4</b> .....	<b>19</b>
5.1	Wildcards	19
5.2	Naredbe za pretraživanje file systema	19
<b>6</b>	<b>Vježba 5</b> .....	<b>21</b>
6.1	Redirekcija >, >>, <	21
6.2	Naredbe grep, cut, sort, head, tail	22
<b>7</b>	<b>Vježba 6</b> .....	<b>23</b>
7.1	Filteri	24
<b>8</b>	<b>Vježba 7</b> .....	<b>25</b>
8.1	Dozvole i vlasništvo datoteka i direktorija ( <i>file permission</i> )	25
<b>9</b>	<b>Vježba 8</b> .....	<b>29</b>
9.1	Shell skripte	29
<b>10</b>	<b>Vježba 9</b> .....	<b>33</b>
10.1	Shell skripte ... nastavak	33
<b>11</b>	<b>Vježba 10</b> .....	<b>37</b>
11.1	Procesi	37

# Uvod

## Upute za vježbe

- Sve naredbe dane u primjerima treba izvršiti.
- Na početku svake vježbe obrisati stari history naredbom `history -c`.
- Kreirati direktorij `vase_prezime_vjezbaxx` i u njemu izvršavati naredbe (`mkdir vase_prezime_vjezbaxx`).
- Na kraju vježbi spremiti history u datoteku `prezime_ime_vjxx.txt` (`history > prezime_ime_vjxx.txt`).
- Nepromijenjenu datoteku `prezime_ime_vjxx.txt` uploadati na <https://moodle.oss.unist.hr/course/view.php?id=133>.
- Obrisati direktorij `vase_prezime_vjezbaxx` (`rm -rf vase_prezime_vjezbaxx`).



# Vježba 1

## 2.1 Instaliranje Linux operativnog sustava na virtualnu mašinu

Virtualna mašina (VM) je program koji simulira računalo i na koju možemo instalirati drugi operativni sustav. Operativni sustav (OS) računala zovemo *host OS*, a operativni sustav instaliran na VM zovemo *guest OS*.

U ovoj vježbi instalirat ćemo Linux OS Mint distribuciju koja je bazirana na Ubuntu distribuciji koja je pak bazirana na Debian distribuciji. Sve *Debian-based* distribucije imaju isti sustav pakiranja i instaliranja programa.

**Zadatak 2.1** Slijediti sljedeće korake instalacije:

- Odabrati Linux distribuciju, te preuzeti njenu `.iso` datoteku.
- Kopirati `.iso` datoteku na disk.
- Pokrenuti Oracle VirtualBox manager.
- Iz izbornika odabrati Machine, New.
- Upisati karakteristike virtualnog stroja: Naziv, Tip (Linux), Verzija (Ubuntu 32-bit), RAM (1500 MB)
- Napraviti virtualni hard disk (VDI, Dynamic, 16 GB)
- Odabrati ISO sliku: VM settings, Storage, označiti DVD (Empty), Choose Virtual Optical Disk File
- Instalirati operativni sustav. Odabrati **manualno** određivanje veličine particija.
- Prijedlog veličina particije

/	ext4	12GB
swap	linux swap	max 1 GB
/home	ext4	3GB

- username: student

password: student123



## Vježba 2

### 3.1 Command line interface (CLI)

Command line interface (CLI) je naziv za programe koji omogućuju unošenje naredbi u tekstualnom obliku koje potom izvršava *Command line interpreter* ili ljuska (engl. *shell*). *Shell* je sučelje između korisnika i operativnog sustava. Na Linux operativnim sustavima dostupno je nekoliko *shellova*, aktivni možemo detektirati naredbom `echo $SHELL`.

*Shell* je program koji interpretira naredbeni redak, parsira ga i pokreće procese navedene u njemu. Naredbeni redak ima odzivni znak (za *bash* je to znak `$` u korisničkom modu rada, a `#` u *root* modu). Parsiranje se vrši tako da se podijeli redak u riječi (engl. *words*) na sljedeći način:

- svako pojavljivanje navodnika označava tekst koji slijedi do sljedećeg navodnika kao jednu riječ
- svako pojavljivanje praznine označava kraj riječi
- naredbena linija završava novim redom ili znakovima `;`, `&`, `|`.

Svaka riječ u naredbenom retku je pridružena nizu stringova. Prva riječ ima indeks 0 u nizu, druga 1 itd.

Na Linux OS možemo koristiti programe *Konsole* ili *xterm* kao CLI ili instalirati neki drugi CLI (*yakuake*, *guake*, *terminator*, *eterm* itd.). Uobičajeno naziv za te programe je konzola ili terminal.

Kod korištenja terminala važan nam je podatak u kojem dijelu datotečnog sustava se nalazimo. Inicijalno je to *home* direktorij (u našem slučaju `/home/student`). Kratica za *home* direktorij je `~`.

**Primjer 3.1** Pozicioniranje u *home* direktorij

- `ls ~`
- `ls /home/student`

Osim *home* direktorija, bitan direktorij je *root* direktorij. Oznaka za njega je `/`.

**Primjer 3.2** Pozicioniranje u *root* direktorij

- `cd /`

## 3.2 Naredbe Linux operativnog sustava

Opći oblik naredbi u Linux operativnim sustavima je

```
$ command [option(s)] [argument(s)]
```

Moguće opcije i argumente te njihov opis mogu se dobiti naredbom `man`.

```
man command
```

### Opće naredbe

**Zadatak 3.1** Izvršiti sljedeće naredbe u komandnoj liniji:

- Ispisati datum. Naredba `date`.
- Proučiti opcije naredbe `cal` koristeći naredbe `man` i `info`.
- Ispisati kalendar za 2020. Ispisati kalendar za drugi mjesec 2020. Ispisati kalendar za 1., 2. i 3. mjesec 2020. koristeći naredbu `cal`.
- Obrisati sadržaj ekrana koristeći naredbu `clear`.
- Ispisati poruku koristeći naredbu `echo`. Npr. `echo Hello world`.
- Ispisati *fortune cookie*. Naredba `fortune`. Ako program `fortune` nije instaliran, instalirati ga koristeći `sudo apt-get install fortune`.

### Korištenje naredbe `history`

**Zadatak 3.2** Izvršiti sljedeće naredbe u komandnoj liniji.

- `history`
- `history 10`
- `!10`
- ponovno pokretanje naredbe iz historyja sa modifikacijom: `!echo:s/world/you`
- pretraga history-ja: `<Ctrl-R>ech`

## Tips and tricks

**Zadatak 3.3** Isprobati sljedeće u komandnoj liniji:

- Pomoću strelice gore i strelice dolje možemo pristupiti već izvršenim naredbama. Isprobati!
- Za pregled prethodnih naredbi i rezultata koristiti `shift+page-up`, `shift+page-down`.
- Naredbe i putanje se mogu dovršiti korištenjem tipke `<TAB>`. Isprobajte: `his<TAB>`
- Selektirajte neku od predhodno napisanih naredbi i kopirajte je koristeći srednji botun miša ili istovremeno oba botuna na touchpadu. Selekcija - *copy*, srednji botun - *paste*.

## Izvršavanje naredbi kao `root`

- Dva načina za izvršavanje naredbi kao `root`:
  - `su root` ili `sudo su`, password je isti (`student123`).
  - `sudo naredba`
- Uočite drugačiji komandni prompt.
- Izađite sa `exit`.

## 3.3 Korištenje repozitorija

Svaka distribucija Linuxa dolazi sa svojim **repozitorijem** tj. kolekcijom programa kompajliranih za tu distribuciju i verziju Linuxa. Repozitorij se nalazi na serveru, a lokalno je pohranjen popis dostupnih programa. Osim glavnog repozitorija, korisnik može dodati i druge repozitorije. Programi u repozitoriju dostupni su u obliku **paketa** (engl. *package*). Lokalno, na računalima, je pohranjena lista dostupnih paketa. Listu dostupnih paketa osvježavamo naredbom `apt-get update` pokrenutom kao *root*.

Za rad sa softverskim paketima (pretraga i instaliranje) može se koristiti GUI aplikacija ili se može koristiti naredbena linija. Programi za baratanje paketima zovemo *Package Managerima*. Za Ubuntu, Mint, Debian i osatle distribucije izvedene iz Debian

distribucije GUI programi su *synaptic*, Software Centar ili Software Manager te Update Manager. Program za rad iz naredbenog retka je APT (Advanced Package Tool).

Osnovne apt naredbe:

- `apt-get update` - osvježava listu dostupnih paketa (pokrenuti kao *root*)
- `apt-cache search name` - pretraga repozitorija, vraća sve pakete koji u svom imenu ili opisu imaju riječ *name*
- `apt-get install paket` - instalacija *paketa* na računalo (pokrenuti kao *root*).

## Instaliranje programa (Ubuntu, Mint i ostale Debian distribucije)

**Zadatak 3.4** Izvršiti sljedeće naredbe u komandnoj liniji:

- Pokreniti program *synaptic* (kao *root*) i instalirajte program `code::blocks` ili neki drugi C IDE. Zatvorite *synaptic*.
- Koristeći naredbu `apt-get` napravite update repozitorija (`sudo apt-get update`), a zatim instalirajte neki program (npr. `sudo apt-get install fslint`).
- Pronaći u repozitoriju programe za pisanje nota `apt-cache search "music score"` i instalirajte ga.

## 3.4 Kompajliranje programa za Linux platformu

Programi za Linux OS u najvećoj mjeri su pisani u programskim jezicima C/C++, Python, Perl i Java. Među njima, programi napisani u C/C++ se kompajliraju te ih je uglavnom moguće preuzeti u obliku izvornog kôda, te samostalno kompajlirati. Sa većinom distribucija Linux dolazi i C/C++ kompajler GNU GCC (GNU Compiler Collection) no ako već nije instaliran može se instalirati zajedno sa ostalim potrebnim alatima i bibliotekama koristeći (za distribucije izvedene iz Debiana) `sudo apt-get install build-essential`. To je ustvari kolekcija alata za C, C++, Objective-C, Fortran, Ada i Go jezike uključujući i biblioteke za te jezike. GNU GCC izdan je pod GNU licencom slobodnog softvera.

### C kompajler

**Zadatak 3.5** Kompajlirati jednostavan C program.

- Kreirati datoteku `hello.c` koristeći naredbu `touch`.
- Otvoriti datoteku programom `gedit`, `kate`, `nano`, `codeblocks` ili nekim editorom.
- Napisati neki jednostavan C program (npr. program koji ispisuje poruku "Hello

world!").

- Iskompajlirati program: `gcc hello.c -o hello`. Opcija `-o` daje ime izvršnoj datoteci. Ako te opcije nema, izvršna datoteka zvat će se `a.out`
  - Ako nedostaje standardna biblioteka instalirajte je `sudo apt-get install libc6-dev`.
  - Ako nedostaje kompajler možete ga instalirati zajedno sa pripadajućim paketima `sudo apt-get install build-essential`.
- Pokreniti izvršnu datoteku: `./hello`

Izvršna datoteka je binarna datoteka. Takvim datotekama barata OS. Binarne datoteke su datoteke koje ne sadrže samo tekst. Vrstu datoteke možemo detektirati naredbom `file`. Za izvršnu datoteku troslojna oznaka dobivena naredbom `file` bit će ELF, slika spremljena kao png imat će oznaku PNG, dok će pdf dokument imati oznaku pdf. U Linux zajednici uobičajeni je naziv *binaries* za izvršne binarne datoteke.

**Zadatak 3.6** Sadržaj binarne datoteke `hello` otvorite programom `hexdump` ili od. Naredbom `strings` ispišite sve stringove u binarnoj datoteci i zapišite ih u datoteku `vjezba1bin.zip`. Naredbom `file` provjerite format datoteke `vjezba1bin.zip`. Druge informacije o datoteci možete dobiti naredbom `stat`.



## Vježba 3

### 4.1 Naredbe za navigaciju file systemom

U radu s terminalom bitan podatak je tekući direktorij. To je trenutna pozicija u datotečnom sustavu. Datotečni sustav ima hijerarhijsku strukturu direktorija. Na vrhu je root direktorij (/), njegova djeca su `bin`, `home`, `sbin`, `var`, `proc` itd.

- Naredba `pwd` ispisuje radni ili tekući direktorij. Prilikom pokretanja shella, tekući direktorij biti će `/home/student`.
- `cd` - promjena tekućeg direktorija (*change directory*)

```
cd path
```

**Primjer 4.1** Korištenje naredbe `cd`.

- `cd ime_dir` - tekući direktorij postaje `ime_dir`
- `cd ..` - tekući direktorij postaje roditelj
- `cd /var/spool/anacron` - tekući direktorij postaje `/var/spool/anacron`
- `ls` - daje popis datoteka i direktorija u danom direktoriju (ako nema parametara, izlista se tekući direktorij)

```
ls direktorij
```

**Primjer 4.2** Primjer korištenja naredbe `ls`.

- `ls ime_dir` - sadržaj direktorija `ime_dir`

- `ls ..` - sadržaj roditeljskog direktorija
- `ls /var` - sadržaj direktorija `/var`

### Opcije

Neke od opcija naredbe `ls`

- `ls -l` prikazuje detaljniji ispis
  - `ls -a` prikazuje i datoteke koje počinju sa `.` (hidden file)
  - `ls -al` kombinacija prethodne dvije opcije
  - `ls -r` ispisuje sadržaj direktorija u obrnutom redoslijedu (leksikografski)
- `mkdir` kreira direktorij

```
mkdir ime_dir
```

- `rmdir` briše direktorij. Direktorij treba biti prazan, inače `rm -r`

```
rmdir ime_dir
```

- `cp` kopira datoteke ili direktorije

```
cp tko gdje
```

### Primjer 4.3 Primjer korištenja naredbe `cp`

- `cp ime.txt ~/dir1/` - kopira datoteku `ime.txt` iz tekućeg direktorija u direktorij `~/dir1/`
  - `cp ime.txt ../` - kopira datoteku `ime.txt` iz tekućeg direktorija u roditeljski direktorij `..`
  - `cp ime.txt ../ime2.txt` - kopira datoteku `ime.txt` iz tekućeg direktorija u roditeljski direktorij `..` s novim imenom.
  - `cp -R dir1 dir2` - kopira direktorij i sve njegove elemente u drugi direktorij
- `mv` premješta datoteke ili direktorije

```
mv tko gdje
```

### Primjer 4.4 Primjer korištenja naredbe `mv`.

- `mv ime.txt ~/dir1/` - premješta datoteku `ime.txt` iz tekućeg direktorija u direktorij `~/dir1/`
- `mv -R dir1 dir2` - premješta direktorij i sve njegove elemente u drugi direk-



torij

- `mv ime.txt ../ime2.txt` - premješta datoteku `ime.txt` iz tekućeg direktorija u roditeljski direktorij `..` s novim imenom.
- `mv -R dir1 dir2` - premješta direktorij i sve njegove elemente u drugi direktorij

#### **Zadatak 4.1** Isprobajte:

- Što rade sljedeće naredbe: `cd`, `ls`, `mkdir`, `rmdir`, `pwd`?
- Koristeći naredbu `touch` kreirati novu datoteku.
- Sadržaj nekog direktorija ispišite u novu datoteku koristeći preusmjerenje `>`.
- Koristeći naredbe `cat` i `less` prikazati sadržaj neke datoteke.

#### **Zadatak 4.2** Riješite sljedeće zadatke koristeći naredbe `cd`, `ls`, `mkdir`, `rmdir`, `mv`, `cp`, `pwd`:

- Ispitajte koji je tekući direktorij (direktorij u kojem se nalazite). Naredba `pwd`.
- Promijenite tekući direktorij. Neka novi bude novokreirani direktorij. Naredba `cd`.
- Kreirajte direktorij `tmp` i u njemu nekoliko datoteka (npr. imena `prva`, `druga` i `treca`).
- Naredbom `tree` provjerite učinjeno.
- Izlistajte popis datoteka u tekućem direktoriju. Naredba `ls`. Ispitajte opcije naredbe `ls`: `ls -a`, `ls -l`, `ls -d`, `ls -t`, `ls ..`, `ls .` i kombinacije npr. `ls -ltr`.
- Kopirajte datoteku `prva` u direktorij `vjezba3`. Naredba `cp`.
- Preimenujte datoteku `prva` u `prvatmp` koristeći naredbu `mv`.
- Pomaknite datoteku `druga` u direktorij `vjezba3` i preimenujte je koristeći samo jednu naredbu `mv`.
- Izlistajte sadržaj direktorija `tmp` u obrnutom redoslijedu.
- Kopirajte direktorij `tmp` i sve datoteke u njemu u direktorij `tmp2`. Koristite opciju `cp -R`.

## 4.2 Putevi *paths*

- Prikažite svoj `search path`. Naredba `echo $PATH`
- Koja je putanja do vašeg `home` direktorija. Naredba `pwd`.
- Idite u `/var/tmp` direktorij.
- Idite u `/var/spool/anacron` direktorij koristeći samo jednu naredbu. Koji je vaš tekući direktorij (`pwd`)?

- Vratite se u direktorij vjezba3.

### 4.3 Direktorij /proc

- Idite u /proc direktorij.
- Koji CPU koristi vaše računalo?
- Koliko RAM-a trenutno koristi?
- Koliki je swap space?
- Koje file sisteme poznaje vaš operativni sustav?

→ Popis svih pokrenutih naredbi eksportirajte u datoteku imena prezime\_ime\_vj3.txt.  
Uploadajte datoteku na <http://moodle.oss.unist.hr>.

## Vježba 4

### 5.1 Wildcards

znak	značenje	primjer
*	zamjenjuju grupu znakova bilo koje veličine	echo D*
?	zamjenjuje jedan znak	echo ?ocument
[]	lista mogućih znakova	ls *[tT]
-	raspon znakova	ls *[1-5]
!	invertira listu znakova	ls *[^1-5]

**Zadatak 5.1** Ispišite sve datoteke u direktoriju /usr/bin koje:

- završavaju sa config.
- počinju sa A ili a.
- završavaju sa znamenkom.
- imaju znamenku 1 na drugoj poziciji.
- nemaju znamenku na zadnjoj poziciji.

### 5.2 Naredbe za pretraživanje file systema

#### Pretraživanje naredbom find

Sintaksa `find <path> -name <searchstring>`.

opcija	opis	primjer
-name	pretraživanje po imenu	find /usr/bin/ -name 'v*'
-iname	pretraživanje po imenu ( <i>case insensitive</i> )	find /usr/bin/ -iname 'v*'
-type	pretraživanje po tipu	find / -type d -iname '*s'
-mtime	pretraživanje datoteka modificiranih u zadanom vremenu	find /var/backups -mtime +5
-size	pretraživanje po veličini u blokovima	find /var/backups -size +50
-newer	pretraživanje datoteka novijih od navedene	find /var/backups -newer /var/lock

## Pretraživanje naredbom locate

Naredba `locate` omogućava brzo pronalaženje indeksiranih datoteka u datotečnom sustavu. Primjer: `locate password`. Indeksiranje se vrši naredbom `updatedb`.

**Zadatak 5.2** Riješite sljedeće zadatke koristeći `find` ili `locate` naredbu:

1. Nađite sve datoteke u `/usr/games` direktoriju čija imena počinju malim slovom.
2. Nađite sve datoteke u `home` direktoriju kreirane danas.
3. Nađite sve datoteke u `home` direktoriju starije od tjedan dana.
4. Nađite lokaciju programa `rmdir`.
5. Nađite lokaciju datoteke `.bash_history`. Što je u njoj?
6. Pronađite datoteku `.bash_logout` naredbom `locate`.
7. Pronađite sve datoteke u `home` direktoriju veće od 5MB. Naredba `find . -size +5000k`.
8. Pronađite sve datoteke na file systemu koje sadrže u imenu riječ `kernel`. Naredba `find / -iname *kernel*`

## Vježba 5

### 6.1 Redirekcija `>`, `>>`, `<`

operator	opis	primjer
<code>&gt;</code>	dodavanje u datoteku ili stdout	<code>ls ~ &gt; lista.dat</code>
<code>&gt;&gt;</code>	dodavanje na kraj datoteke	<code>ls -al ~ &gt;&gt; lista.dat</code>
<code>&lt;</code>	ulaz iz datoteke ili stdin	<code>wc -l &lt; lista.dat</code>
<code>2&gt;</code>	greške (2-stderr) se ispisuju u datoteku	<code>find /etc/ -iname gr* 2&gt; err.dat</code>

- Nakon svakog primjera pogledati datoteke u koje je izlaz preusmjeren.
- Pogledati man `wc`.
- `/dev/null` je specijalni file u koji se mogu preusmjeriti neželjeni podaci. `/dev/null` je uvijek prazan.

#### Zadatak 6.1 Redirekcija:

1. Izvršite sljedeće naredbe:

- `find /etc/ -iname gr*`
- `find /etc/ -iname gr* 2> vjezba5/err.dat`
- `find /etc/ -iname gr* 2> /dev/null`

Koja je razlika između gornjih naredbi?

2. Sadržaj direktorija `/etc/init` spremite u datoteku `etc.dat` (u direktorij `vjezba5`).
3. Koristeći redirekciju i program `wc` prebrojite koliko bajtova ima datoteka `etc.dat`.
4. Provjerite rezultat uspoređujući ga sa ispisom `ls -al etc.dat`.
5. Izvršite sljedeće naredbe:
  - `cat < etc.dat`

- `cat < etc.dat > etccopy.dat`
- `cat < etc.dat > etccopy.dat 2> error.dat`

Koja je razlika između gornjih naredbi?

## 6.2 Naredbe grep, cut, sort, head, tail

naredba	opis	primjer
cut	ispisuje određene stupce odvojene delimiterima iz datoteke	<code>cut -f1 -d. etc.dat</code>
grep	ispisuje linije u kojima se pojavljuje <i>pattern</i>	<code>grep hal etc.dat</code>
sort	sortira linije u tekstualnoj datoteci	<code>sort etc.dat</code>
head	ispisuje prvih <i>n</i> linija datoteke (default je 10)	<code>head etc.dat</code>
tail	ispisuje zadnjih <i>n</i> linija datoteke (default je 10)	<code>tail etc.dat</code>

**Zadatak 6.2** Pogledajte sadržaj datoteke `/etc/passwd` (koristiti naredbu `cat` ili `less`).

1. Ispisati samo korisničko ime i home direktorij (tj. prvo i šesto polje) u datoteci `/etc/passwd`.  
`cut -f1,6 -d: /etc/passwd`  
`-d` određuje koji je delimiter koji odvaja polja u redku (u našem primjeru je :)  
`-f` određuje koja polja će se prikazati (u našem primjeru prvo i šesto)
2. Iz datoteke `/etc/passwd` ispisati samo redak koji sadrži riječ `root`. Koristiti naredbu `grep`
3. Ispisati sortiranu datoteku `/etc/passwd` koristeći naredbu `sort`.
4. Ispisati prvih 20 linija `/etc/passwd` datoteke. Koristiti naredbu `head`.
5. Ispisati zadnjih 10 linija `/etc/passwd` datoteke. Koristiti naredbu `tail`.

## Vježba 6

### Pipeline

**Pipe** ili **pipeline** je niz od dvije ili više naredbi odvojenih vertikalnom crtom `|`. Standardni output prve naredbe, standardni je input druge naredbe.

**Zadatak 7.1** Izvršite sljedeće linije

- `ls | wc -l`
- `ls | grep dat`
- `ls | tac` - ispis u obratnom redoslijedu (pogledati `man tac`)

**Zadatak 7.2** U direktoriju kreirajte direktorij `etc`.

1. U direktoriju `etc` kreirajte datoteku sa nastavkom `.txt` u kojoj su zapisane datoteke iz direktorija `/etc` koje počinju sa slovom `a` (`ls -d /etc/a*`). Uočite razliku između `etc` i `/etc`. Koristite preusmjeravanje `>` u datoteku.
  2. U direktoriju `etc` kreirajte datoteku sa nastavkom `.dat` u kojoj su zapisane datoteke iz direktorija `/etc` koje počinju sa slovom `a` u obratnom redoslijedu (`ls -dr /etc/a*`).
- `ls | grep dat`
  - Iz datoteke `/etc/passwd` izdvojite prvi i šesti stupac koji se odnose na `user-a mail`. Kombinirajte `grep` i `cut`.
  - Izlistajte datoteke i direktorije u tekućem direktoriju sortirane po veličini. Kombinirajte `ls -l` i `sort -n -k 5` (sortiranje po petom stupcu).
  - Izlistajte datoteke (samo datoteke) u tekućem direktoriju sortirane po veličini. Za

pronalaženje datoteka koristite `find . -type f -ls`.

## 7.1 Filteri

**Filter** je naredba koja čita sa standardnog ulaza (stdin), transformira ulaz i piše na standardni output. Može biti srednji dio pipelinea.

**Primjer 7.1** Isprobajte sljedeće naredbe

- `ls | grep .txt | wc -l`
- `cat etc/etc.dat etc/etc.txt | sort -n -k 5 | uniq > finaletc.list` (cat spaja datoteke (konkatenacija), one se zatim sortiraju po petom polju i sa uniq se izbacuju duplikati, te se rezultat zapiše u finaletc.list)

**Zadatak 7.3** Iz datoteke `/etc/passwd` izdvojite sve retke u kojima je shell bash (stupac 7) (naredba `grep`). Ispišite samo prvi i sedmi stupac tj. username i shell (naredba `cut`) i sortirajte po username-u (naredba `sort`).



## Vježba 7

### 8.1 Dozvole i vlasništvo datoteka i direktorija (*file permission*)

#### Korisnici i grupe

Da bi se datoteke zaštitile od neovlaštenog pristupa, Linux dozvoljava postavljanje dozvola korištenja datoteka i direktorija. Dozvole se određuju za:

- vlasnika datoteke (**owner**)
- članove grupe kojoj je datoteka dodjeljena (**group**)
- sve ostale korisnike (**other**).

**Primjer 8.1** Izvršite `ls -l`. Prvih 10 znakova označava dozvole korištenja datoteke, zatim je broj koji označava broj linkova, sljedeća polja su vlasnik, grupa, veličina, vrijeme zadnje promjene i ime datoteke.

```
-rwxr-xr-x 1 os os 10103 2010-12-15 10:58 main
-rw-r--r-- 1 os os 248 2010-12-15 10:58 main.c
```

U gornjim primjerima:

- Prvi znak - ili d označava tip datoteke (datoteka ili direktorij).
- Sljedeća tri znaka rwx ili rw- označavaju dozvole vlasnika datoteke.
- Sljedeća tri znaka r-x ili r-- označavaju dozvole za grupu.
- Sljedeća tri znaka r-x ili r-- označavaju dozvole za ostale korisnike.

**Za datoteke:**

Troslovna kombinacija slova *r*, *w* i *x* označavaju redom pravo čitanja, pisanja i izvršavanja datoteke.

**Za direktorije:**

- *r*: korisnik može vidjeti sadržaj direktorija (npr. sa `ls` naredbom).
- *w*: korisnik može mijenjati sadržaj direktorija tj. kreirati, brisati i preimenovati datoteke u direktoriju.
- *x*: korisnik može koristiti direktorij kao svoj tekući direktorij, tj. može ući u njega naredbom `cd`.

**Mijenjanje dozvola**

Sintaksa `chmod <MODE> <ime_datoteke>`

`<MODE>` može biti napisan simbolički npr. `chmod go-rx kopija.c` ili oktalno npr. `chmod 700 kopija.c`.

**Simbolički**

`chmod <TKO OPERATOR STO> <ime_datoteke>`

`<TKO>` može biti:

- **u** user
- **g** grupa
- **o** ostali
- **a** svi

`<OPERATOR>` može biti:

- **+** dodavanje prava
- **-** oduzimanje prava
- **=** skidanje svih prava i dodavanje specificiranih

`<STO>` može biti *r*, *w* i *x*.

**Primjer 8.2** Primjer simboličkog načina pridjeljivanja prava korištenja:

- `chmod u=rw,go=r kopija.c` - user dobija pravo pisanja i čitanja, a grupa i ostali samo čitanja.
- `chmod a+x main` - svi dobiju pravo izvršavanja uz već postojeća prava.

## Oktalno zapisivanje prava korištenja

```
chmod <ZZZ> <ime_datoteke>
```

	u	g	o
	754		
access	r w x	r w x	r w x
binary	4 2 1	4 2 1	4 2 1
enabled	1 1 1	1 0 1	1 0 0
result	4 2 1	4 0 1	4 0 0
total	7	5	4

Z - oktalna znamenka (znamenka između 0 i 7)

### Primjer 8.3

- `chmod 644 kopija.c` - user dobija pravo pisanja i čitanja, a grupa i ostali samo čitanja.
- `chmod 755 main` - svi dobiju pravo izvršavanja uz već postojeća prava.

**Zadatak 8.1** Nakon svake izvršene naredbe u primjerima i zadacima, izvršite naredbu `ls -l` kako bi vidjeli promjene.

- U direktoriju vjezba7 kreirajte datoteku main.c u kojoj će biti neki c program.
- Iskompajlirajte ga: `gcc -o main main.c`
- Napravite kopiju datoteke main.c. Neka se kopija zove kopi.ja.c.

**Zadatak 8.2** Datoteci `kopi.ja.c` dodijelite takva prava da samo user može mijenjati datoteku.

**Zadatak 8.3** Datoteci main dodijelite takva prava da samo user i pripadajuća grupa mogu izvršavati datoteku tj. program main.

**Zadatak 8.4** Datoteci main skinite pravo mijenjanja za usera. Pokušajte pokrenuti program ./main.

**Zadatak 8.5** Kreirajte direktorij test i u njega kopirajte datoteku main. Direktoriju test skinite pravo pisanja w. Kopirajte datoteku main.c u njega. Kakav je rezultat i zašto?



## Vježba 8

### 9.1 Shell skripte

**Skripta** je datoteka koja sadrži naredbe shella. Shell čita datoteku i izvršava naredbu po naredbu kao da je pisana putem tipkovnice. Na taj način ne moramo pamtit i naredbe, poslovi se mogu izvršavati u određeno vrijeme, ne gubimo vrijeme rješavajući problem putem GUI-a.

**Primjer 9.1** Primjeri shell skripti:

- naći sve izvršne programe u PATH-u koji odgovaraju nekom *patternu* [http://user.it.uu.se/~matkin/documents/shell/#Rename\\_files](http://user.it.uu.se/~matkin/documents/shell/#Rename_files)
- napraviti *backup* za sve datoteke koje su mijenjane u nekom danu <http://tldp.org/LDP/abs/html/special-chars.html#EX58>
- konvertirati jedan grafički format u drugi (JPEG u PNG) [http://en.wikipedia.org/wiki/Shell\\_script](http://en.wikipedia.org/wiki/Shell_script)
- obavijestiti kada se neki user ulogira ili odlogira <http://www.comp.eonworks.com/scripts/scripts.html>
- ispisati imena datoteka koje u sebi sadrže liniju dužu od  $n$  znakova <http://www.comp.eonworks.com/scripts/scripts.html>
- jednom tjedno kopirati sve datoteke na "backup server" na lokalnoj mreži. itd.

### Pisanje shell skripte

Koraci:

1. Napisati skriptu npr. `script1` u nekom editoru (npr. `gedit`). U prvom redku

napisati `#!/bin/bash`!

2. Dodijeliti pravo izvršavanja (x) skripti npr. `chmod 755 script1`
3. Smjestiti je negdje gdje je shell može naći npr. u tekući direktorij
4. Izvršiti je npr. `./script1`

**Zadatak 9.1** Napisati Hello world! skriptu i izvršiti je.

```
#!/bin/bash
echo "Hello world!"
```

## Elementi shell skripte

- **Varijable:** definiraju se ovako: `X="Hello"`, a vrijednost se dohvaća sa `$X`. U varijablu možemo spremati i vrijednost nekog aritmetičkog izraza koristeći dvostruke okrugle zagrade npr. `X=$((Y+Z))`
- **Standardne (environment) varijable:** su varijable shella (npr. `$PATH`, `$USER`, `$PWD` ili `$OLDPWD`) koje možete pogledati naredbom `env`.
- **Parametri tj. argumenti komandne linije** označeni su sa `$1`, `$2`, `$3` itd.
  - `$@` je `"$1" "$2" "$3"`
  - `$*` je `"$1 $2 $3"`
  - `$\#` je broj parametara

**Zadatak 9.2** Napišite (i proučite) skriptu prva i izvršite je sa `./prva` jen dva. Što skripta radi?

```
#!/bin/bash
echo Ime skripte: $0
echo Prvi parametar: $1
echo Prvi parametar: $2
echo Broj parametara: $#
```

## Praznine

Znak jednakosti (=) koristi se za pridjeljivanje vrijednosti i za usporedbu.

- Kod pridjeljivanja vrijednosti znak = nije omeđen prazninama: `val=33` ili `str="Neki string"`.
- Kod uspoređivanja je znak = omeđen prazninama: `if [ $str = "Nekistring" ]; then echo "je"; else echo "nije"; fi`

## Jednostruki vs. dvostruki navodnici

**strong quoting** Unutar jednostrukih navodnika ništa se ne interpretira. Primjer: `echo 'Your PATH is: $PATH'` ispisati će se `Your PATH is: $PATH`. Znakovi nemaju posebno značenje

**weak quoting** Unutar dvostukih navodnika interpretira se većina znakova, ali neki znakovi imaju posebno značenje i njih se ne interpretira:

- *space* ne interpretira kao separator riječi npr.
  - `touch prva datoteka.dat` kreirati će (ili promijeniti vrijeme pristupa) dvije datoteke: `prva` i `datoteka.dat`
  - `touch "prva datoteka.dat"` kreirati će (ili promijeniti vrijeme pristupa) datoteku: `prva datoteka.dat`
- znakove unutar jednostrukih navodnika ne interpretira kao "strong quoting" npr. `echo '$PATH'` će ispisati cijeli *path*
- wildcardove ne interpretira npr. `ls -l "prva*"` će ispisati poruku: `ls: cannot access prva*: No such file or directory`
- ekspanziju `pathnamea` i varijabli `echo "Koristimo $TERM"` će ispisati `Koristimo xterm`, odnosno ime terminala kojeg koristimo.
- zamjene za komandu (*command substitution*) npr. `echo `ls`` će izvršiti naredbu `ls`

**Zadatak 9.3** Napišite (i proučite) skriptu druga i izvršite je. Što skripta radi?

```
#!/bin/bash
echo "hello, $USER"
echo "Sadrzaj trenutnog direktorija, $PWD"
ls -l # list files
```

**Zadatak 9.4** Napišite (i proučite) skriptu treća i izvršite je. Što skripta radi?

```
#!/bin/bash
X=3
Y=4
echo $((X+Y))
echo $((X-Y))
echo $((X*Y))
echo $((X/Y))
echo $((X%Y))
```

**Zadatak 9.5** Modificirajte skriptu tako da vrši osnovne aritmetičke operacije nad

argumentima komandne linije.



## Vježba 9

### 10.1 Shell skripte ... nastavak

#### Uvjetno izvršavanje

**Uvjetno izvršavanje** je izvršavanje naredbe ili skupa naredbi u ovisnosti o povratnoj vrijednosti druge naredbe. Realizira se pomoću `if then else fi` notacije.

```
if test-commands; then
    consequent-commands;
[elif more-test-commands; then
    more-consequents;]
[else alternate-consequents;]
fi
```

Ukoliko se izvršava samo jedna naredba moguće je zamijeniti `if then fi` sa naredbama `&& i ||`.

- `command1 && command2` je ekvivalentno  
`if (command1 successful)then do (command2)`
- `command1 || command2` je ekvivalentno  
`if (command1 not successful)then do (command2)`

#### Uglate zagrade

**Sintaksa if naredbe** Uglate zagrade su samo pokratak za naredbu `test`.

Sintaksa: `test <IZRAZ>`

<IZRAZ> je kombinacija opcija i vrijednosti.

```
#!/bin/bash
# test da li /etc/passwd postoji

if test -e /etc/passwd; then
    echo "Alright man..." >&2
else
    echo "Yuck! Where is it??" >&2
    exit 1
fi
```

Drugi način je testiranje pomoću uglate zagrade [. Naredba završava zatvorenom uglatom zagradom ].

Uglati zagrada [ nije dio if naredbe.

. Sintaksa je [ <IZRAZ> ] (uočite razmak nakon otvorene uglate zagrade).

```
#!/bin/bash
# test da li /etc/passwd postoji

if [ -e /etc/passwd ]; then
    echo "Alright man..." >&2
else
    echo "Yuck! Where is it??" >&2
    exit 1
fi
```

Za test-commands pogledajte link [Bash test and comparison functions](#). Neki od test naredbi su dani u sljedećoj tablici:

Izrazi za testiranje statusa datoteke	
-f <file>	file je obična datoteka
-d <file>	file je direktorij
-r <file>	file ima pravo čitanja
-w <file>	file ima pravo pisanja
-x <file>	file ima pravo izvršavanja
-s <file>	file nije prazan (ima nenula duljinu)
-e <file>	file postoji
Izrazi za testiranje stringova	
-n <string>	string nije prazan
-z <string>	string je prazan
<string> == <string>	stringovi su jednaki
<string> != <string>	stringovi nisu jednaki
Aritmetički izrazi	
<value> -eq <value>	jednakost
<value> -ne <value>	nejednakost
<value> -lt <value>	manje od
<value> -le <value>	manje ili jednako
<value> -gt <value>	veće od
<value> -ge <value>	veće ili jednako

**Primjer 10.1** Izvršite sljedeću skriptu. Što radi?

```
#!/bin/bash
A=101
if [ "$A" -eq 10 ]
then
    echo A = 10
else
    echo A != 10
fi
```

**Zadatak 10.1** Modificirajte zadatak iz prethodne vježbe (osnovne aritmetičke operacije s parametrima) tako da se parametri zbroje ako je prvi parametar manji od drugog parametra, pomnože ako su jednaki, a oduzmu ako je prvi parametar veći od drugog parametra.

## Naredbe za ponavljanje

### Naredba while

Sintaksa:

```
while expression is true
do
    command(s)
done
```

**Primjer 10.2** Proučite i izvršite!

```
#!/bin/bash
count=0
max=10
while [ $count != $max ]; do count=`expr $count + 1`
    echo "$count"
done
```

### Naredba for

Sintaksa:

```
for identifier in list
do
    command(s) to be executed
done
```

**Primjer 10.3** Proučite i izvršite!

1. 

```
#!/bin/bash
for i in 1 2 3 4
do
    echo "$i"
done
```
2. 

```
#!/bin/bash
for i in {1..4}
do
    echo "$i"
done
```
3. 

```
#!/bin/bash
for i in {1..10..2}
do
    echo "$i"
done
```
4. 

```
#!/bin/bash
for (( i=0; i<5; i++ ))
do
    echo "$i"
done
```
5. 

```
#!/bin/bash
count=0
for d in *
do
    count=`expr $count + 1`
    echo $d
done
echo $count
```

# Vježba 10

## 11.1 Procesi

### Proces

Proces je program u izvršavanju zajedno sa svojim virtualnim adresnim prostorom. Svaki proces ima pridružene i sljedeće vrijednosti:

- ID procesa (PID)
- ID roditelja (PPID)
- ID korisnika i grupe
- environment varijable
- listu otvorenih datoteka
- tekući direktorij
- itd.

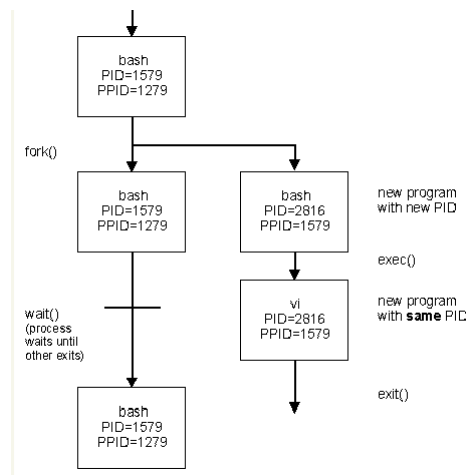
### Pokretanje procesa

Svaki proces je pokrenut od strane nekog drugog procesa. Jedini izuzetak je **init** proces koji pokreće kernel i ima PID 1. Sam shell je proces i njegov PID se može dobiti sa:  
echo \$\$

Iz shella se procesi pokreću izvršavanjem komandne linije. Npr. u komandnoj liniji napišite bash. Otvoriti će se subshell.

**Zadatak 11.1** Izvršite sljedeće akcije:

- Naći PID tekućeg shella. Izvršiti naredbu `history`.
- Pokrenite novi `bash` shell
- Naći PID novootvorenog subshellja. Izvršiti naredbu `history`. Što zaključujete?
- Izaći iz novootvorenog shella naredbom `exit`. Provjeriti PID shella u kojem se nalazite.



Procesi se mogu izvršavati na dva načina:

- **foreground** procesi
- **background** procesi

Kada se proces izvršava kao foreground proces, shell koji je pokrenuo proces treba čekati da proces završi kako bi se mogli pokretati ostali procesi. S druge strane, kada se proces izvršava kao background proces, shell ne mora čekati završetak procesa (odzivni znak se odmah pojavi u shellu). Background procesi se pokreću tako da se na kraj komandne linije doda znak ampersand (&).

Prebacivanje foreground u background:

1. Suspendirati proces (Ctrl+z)
2. Unijeti naredbu `bg`

**Zadatak 11.2** Pokreniti `gedit` kao foreground proces i prebaciti ga u background.

**Zadatak 11.3** Pokrenuti naredbu `jobs` kojom se ispisuju svi background ili suspendirani procesi.

## Nadgledanje procesa

- `ps` - ispituje sve procese (process status). Neke opcije:
  - `a` - svi procesi povezani sa terminalom

- **x** - svi ostali procesi
- **u** - detaljniji ispis
- **-e** - svi procesi
- **-f** - full format
- **-l** - kompletan ispis

Najčešći oblik korištenja **ps** naredbe je `ps aux` ili `ps -ef`

- `lsuf` - ispisuje listu otvorenih datoteka za pojedini proces
- `ps tree` - ispisuje procese u obliku stabla
- `pidof` - ispisuje PID procesa
- `top` - ispisuje trenutno pokrenute procese i informacije o memoriji i CPU.

**Zadatak 11.4** Pokrenite neki program (npr. unijeti `gedit test.txt`). Nađite PID i PPID procesa.

## Završetak procesa

Procesi završavaju zbog dva razloga:

- proces sam završava, automatski ili zbog korisničke intervencije
- drugi proces šalje signal procesu i tako ga terminira (npr. naredbom `kill`)

**Kill signal** Najvažniji signali:

Signal	Kratika	Značenje	Akcija
02	Ctrl+c	interrupt	end process
09		kill	end process (nepovratno)
15		terminate	end process

Sintaksa je `kill -SIGKILL PID`. Predefinirani signal je 15.

**Zadatak 11.5** Ubijte `gedit` iz komandne linije.