

Tečajevi naprednog računarstva u Križevcima

HACK2020 | Hub for Advanced  
Computing Križevci

## Uvod u GNU/Linux

Bilješke s tečaja održanog u sklopu projekta HACK 2020

**Andrej Dundović**

Organizator:



Pokrovitelj:



Križevci, 2020



© 2020 Andrej Dundović.

Ovaj se rad može koristiti, mijenjati i dalje distribuirati po pravilima licence Creative Commons Imenovanje 4.0:

<http://creativecommons.org/licenses/by/4.0/>

Inačica dokumenta: [git-50b6b25](#), 2020-10-26

# Sadržaj

<b>1</b>	<b>Osnove naredbenog retka</b>	<b>1</b>
1.1	Naredbeni redak GNU/Linuxa - Bash . . . . .	2
1.2	Navigacija . . . . .	4
1.2.1	Hijerarhija datotečnog sustava . . . . .	4
1.2.2	Pretraživanje datoteka i direktorija . . . . .	6
1.3	Rad s datotekama i direktorijima . . . . .	7
1.3.1	Provjera datoteka . . . . .	9
1.3.2	Arhiviranje i kompresija datoteka i direktorija . . . . .	10
1.4	Rad s tekstualnim datotekama . . . . .	11
1.5	Naredbe za podešavanje i kontrolu naredbenog retka . . . . .	11
1.6	Sustav ovlasti i dozvola . . . . .	11
1.7	Rad s procesima . . . . .	11
1.8	Rad s mrežom . . . . .	11
1.9	Rad s datotečnim sustavima . . . . .	11
1.10	Upravljanje sustavom . . . . .	11
1.11	Ostalo . . . . .	12
<b>A</b>	<b>Instalacija GNU/Linuxa</b>	<b>13</b>
A.1	Instalacija na virtualno računalo (VirtualBox) . . . . .	15
A.2	Instalacija na fizičko računalo . . . . .	18
	<b>Bibliografija</b>	<b>19</b>
	<b>Popis naredbi</b>	<b>20</b>

# 1 | Osnove naredbenog retka

Interakcija čovjeka i računala u osnovi se svodi na to da računalo prihvaca čovjekove naredbe i vraća mu rezultate tih naredbi. Komunikacijski kanal koji takvu interakciju omogućuje nosi naziv korisničko sučelje (engl. *user interface*). U terminologiji operacijskih sustava, korisničko sučelje zapravo je i zadnji, *najvanjskiji* sloj operacijskog sustava, stoga ne čudi da se naziva još i ljska (engl. *shell*). Ljska se može realizirati u stilu grafičkog korisničkog sučelja (engl. *graphical user interface*, GUI) kao što je na primjer *Windows Shell* ili kao sučelje naredbenog retka (engl. *command-line interface*, CLI). U principu su moguće i druge realizacije, poput glasovnog sučelja (engl. *voice-user interface*, VUI) koje je već prisutno u nekim oblicima (npr. kod pametnih telefona), no uistinu efikasno korištenje takvog viđeno je, barem za sada, samo u Zvjezdanim stazama.

Možda se čini arhaičnim komunicirati s računalom putem naredbenog retka, ali izgleda da takvo korisničko sučelje uporno odbija "izumrijeti", štoviše, i oni operacijski sustavi koji su ga isprva izbacivali, naknadno su ga ipak vratili kao svoj neizostavni dio (npr. *PowerShell* u MS Windows). Razlog je taj što naredbeni redak ima značajnih prednosti nad grafičkim sučeljem u određenim domenama primjene računala. Za početak, naredbeni redak zahtijeva daleko manje računalnih resursa od grafičkog sučelja te je utoliko pouzdaniji i brži, tj. responsivniji, a to pogotovo dolazi do izražaja kod uređaja i računala slabijih performansi kao na primjer kod ugrađenih sustava (engl. *embedded system*). Za ostale razlike, pa tako i prednosti i mane CLI-a u odnosu na GUI, dobro je ponoviti osnovnu logiku funkciranja oba pristupa.

Grafičko sučelje po definiciji za svaku naredbu ima pripadajuću vizualnu reprezentaciju na ekranu, bilo u obliku ikona, bilo u obliku nekih drugih simbola ili vizualnih elemenata, kako bi se mogla pokrenuti, recimo mišem ili putem ekrana osjetljivog na dodir. Očito je nemoguće prikazati na jednom ekranu sve moguće naredbe koje računalo može izvršiti kao vizualne elemente jer se u toj gužvi nitko ne bi snašao, pa su one funkcionalni i hijerarhijski grupirane u izbornike i *podekrane* ("prozore"). Tako da bi se došlo do specifičnije naredbe ili opcije, obično je potrebno otvoriti slijed od nekoliko prozora, izbornika i podizbornika.

Naredbeni redak, s druge strane, ne otkriva bjelodano koje je sve naredbe moguće unijeti, već je prikazan samo prazan redak za unos teksta, a samo određeni tekst predstavlja valjanu naredbu koja se šalje računalu pritiskom na tipku Enter. Na čovjeku je da zna koji

tekst reprezentira koje naredbe. Srećom, naredbe su u pravilu riječi engleskog jezika ili lakopamtljive kratice te slijede određenu logiku i sintaksu, tako da nije potrebno pamćenje cijelog niza nasumično nabacanih riječi, već samo sintakse i manjeg broja naredbi. Iz ovoga se može odmah zaključiti da je korištenje grafičkog sučelja u startu lakše: nije potrebno memorirati naredbe te je intuitivnije izvođenje više zadaća istovremeno koje su vizualno odvojene. Nasuprot tomu, naredbeni redak nudi brži pristup svakoj od naredbi i njihovim opcijama jednom kad se zna što se traži, a često nudi i više opcija pa time i veću kontrolu kako računalo izvršava zadaće. Naredbe se u naredbenom retku mogu povezivati u kombinacije i tako izvršavati kompleksnije zadaće, a mogu tvoriti i skripte (engl. *scripts*) što se već i naziva skriptno programiranje. Suptilnije prednosti lakše će se uočiti kroz praksu koja slijedi u nastavku.

## 1.1 Naredbeni redak GNU/Linuxa - Bash

GNU/Linux u tradiciji Unixa kao osnovni kanal komunikacije između operacijskog sustava i korisnika koristi naredbeni redak. Iako ljudska (engl. *shell*) označava širi pojam, u kontekstu Unixa ona je sinonim za naredbeni redak. Od tuda i naziv interpretatora naredbenog retka (engl. *command line interpreter*) u Unixu: sh (*shell*). Interpretator je program koji interpretira naredbe i prosljeđuje ih operacijskom sustavu te je tako i osnovni nosioc CLI-a. Gotovo sve distribucija GNU/Linuxa dolaze s prepostavljenim interpretatorom naredbenog retka **bash**, a naziv dolazi od kratice *bourne-again shell* što je referenca na originalni Unixov interpretator, sh, kojeg je napisao Steve Bourne.

Moderne distribucije GNU/Linuxa namijenjene korisnicima osobnih računala dolaze s predinstaliranim i prepostavljenim grafičkim sučeljem, a da bi se iz njega pristupilo naredbenom retku, potreban je tzv. emulator terminala. Ovisno o grafičkom sučelju i distribuciji, emulatori terminala nalaze pod imenima: Terminal (GNOME), Konsole (KDE), XTerm i sl. Pokretanjem jednog dobiva se aktivan redak za unos teksta<sup>1</sup> koji najčešće počinje sa sljedećim oznakama:

```
[user@localhost ~]$
```

gdje na mjestu user stoji korisničko ime trenutnog korisnika, na mjestu localhost mrežni naziv računala (engl. *hostname*), a kod tilde (~) nalazi se naziv direktorija u kojem je trenutno naredbeni redak aktivan (tilda je inače oznaka za *home* direktorij korisnika; u ovom slučaju to bi se interpretiralo kao /home/user/). Znak dolara (\$) označuje činjenicu da korisnik ima neprivilegirani pristup, dok bi umjesto njega, znak hasha (#) označavao privilegirani pristup (tzv. *superuser*).

Upisivanjem nepoznate naredbe ili besmislenog niza znakova i pritiskom na Enter, interpretator naredbenog retka (**bash** u ovom slučaju) javlja da takva naredba nije nađena, npr.

<sup>1</sup>Od sada pa nadalje prepostavlja se da čitatelj ima pristup GNU/Linuxu te da u praksi isprobava ono napisano u tekstu. Za pomoć oko instalacije GNU/Linuxa treba vidjeti Prilog A.

```
[user@localhost ~]$ nekabesmislenanaredba  
bash: nekabesmislenanaredba: command not found
```

nakon čega se naredbeni redak vraća u prvotno stanje i spremam je za unos nove naredbe. S druge strane, upisivanjem smislene naredbe, kao npr. **date**,

```
[user@localhost ~]$ date  
Sat 24 Oct 11:22:09 CEST 2020
```

u naredbeni redak ispisuje datum i vrijeme koje mjeri ugrađeni sat u računalu.

Prva riječ u retku uvijek predstavlja samu naredbu dok sve što slijedi naknadno, odvojeno razmacima, predstavlja argumente te naredbe. Ova sintaksa vuče korijene od matematičkih funkcija, odnosno  $f(x, y, z, \dots)$  gdje je  $f$  naziv funkcije, a  $x, y, z$  njezini argumenti. Kada sam argument sadrži razmake, da ne bi bio prepoznat kao dva ili više argumenta, potrebno ga je staviti u navodne znakove, bilo jednostrukе ('argument s razmacima'), bilo dvostrukе ("argument s razmacima"). Neke naredbe, kao navedeni **date**, ne trebaju nužno argument da bi se ispravno pozvala, ali mogu imati opcionalne argumente. U slučaju naredbe **date** može se recimo dodati argument `+%j` koji mijenja ispis pa tako ispisuje samo trenutni dan u godini:

```
[user@localhost ~]$ date +%j  
298
```

Iako ne postoji konsenzus oko naziva, u pravilu se argumenti koji počinju s crticom (-) zovu opcije ili modifikatori naredbe, a nekad i prekidači (engl. *switches*) ili zastave (engl. *flags*). Kod većine naredbi razlikujemo kratke opcije koje počinju samo s jednom crticom i sadrže samo jedno slovo iza crtice te duge opcije koje počinju s dvije crtice te sadrže više od jednog slova. Na primjeru već navedene naredbe imamo opcije `-u` i `--utc` koje su sinonimi i čine to da **date** ispisuje vrijeme u univerzalnom vremenu (engl. *Coordinated Universal Time*, UTC). Kako ovo nije standardizirano tako neke naredbe odstupaju od tog običaja i višeslovnoj opciji prethodi samo jedna crtica. Jednoslovne opcije mogu se pisati i zajedno, pa je slijed "naredba `-a -b -c -d`" moguće skratiti u "naredba `-abcd`" pod pretpostavkom da samoj opciji ne treba pridružiti neku vrijednost, npr. "naredba `-a 10 -b`" ne može se skratiti kao "naredba `-a10b`" jer će interpretator "10b" shvatiti kao jedinstvenu vrijednost. Treba posebno istaknuti da u Unixu, pa time i u GNU/Linuxu, veliko/malo slovo čini razliku pri unošenju naredbi i argumenata tako da recimo "DaTe `--uTC`" neće dati isti rezultat kao "date `--utc`".

Dobrodošla je mogućnost interpretatora pamćenje prošlih unosa, najčešće do 1000 posljednje unesenih naredbi po korisniku, u ovisnosti kako je podešeno. Ta mogućnost naziva se još i engl. *history* iz očitog razloga, a upisivanjem naredbe **history** dobiva se cjelokupni popis svih naredbi gdje su one zadnje izlistane na samom kraju. Do prošlih naredbi može se doći i tipkovničkim strelicama *up/down*, dok one *left/right* služe za pomicanje kursora unutar retka.

## 1.2 Navigacija

Interpretator naredbenog retka uvijek je aktivan u nekom direktoriju zvanom radni direktorij (engl. *working directory*), a pri pokretanju terminala to je najčešće *home* direktorij kao što je već spomenuto ranije. Upisivanjem naredbe **pwd** (od engl. *print working directory*) dobiva se i puna putanja radnog direktorija u hijerarhiji datotečnog sustava.

Hijerarhija datoteka i direktorija na Unixoidnim operacijskim sustavima organizirana je u strukturu stabla gdje je ishodišna točka tzv. *root* direktorij označen samo s / te ne postoji putanja koja izlazi izvan tog stabla. Hijerarhija datotečnog sustava u GNU/Linuxu bit će opisana u sljedećem odjeljku, 1.2.1. Naredbom **tree** može se dobiti i vizualizacija takvog ustroja jer ta naredba ispisuje sve grane (direktorije) i listove (datoteke) tog stabla polazeći iz aktivnog direktorija.

Kretanje kroz direktorije vrši se putem naredbe **cd** (od engl. *change directory*) tako da iza te naredbe slijedi argument odvojen razmakom koji predstavlja naziv direktorija ili putanje u koju se želi prijeći, npr.

```
| [user@localhost ~]$ cd /usr/bin
```

gdje argument može biti absolutna putanja (koja počinje s /) ili relativna u odnosu na aktivan direktorij (koja ne počinje s /). Bez ikakvog argumenta, **cd** vraća u *home* direktorij, baš kao i "cd ~". Umjesto samo tilde, tilda uz korisničko ime, "cd ~user", postavlja radni direktorij u *home* direktorij navedenog korisnika. Nadalje, točka . i dvije točke .. simboli su za aktivan direktorij i roditeljski direktorij. Tako npr. naredba "cd ." ne mijenja ništa jer nas premješta u direktorij u kojem već jesmo, dok "cd .." ide u jedan direktorij iznad (roditeljski) unutar hijerarhije. Nakon uzastopnog upisivanja **cd ..** nužno dolazimo do *root* direktorija koji nema nikakav direktorij izvan sebe pa daljnjim upisivanjem "cd .." aktivna pozicija ostaje na /. Na toj poziciji naredba **tree** ispisuje cijelo stablo datotečnog sustava osim dijelova kojima korisnik nema ovlasti pristupiti.

Teško je znati kuda se kretati u datotečnom sustavu dok se ne zna popis postojećih datoteka i direktorija. Upravo tu funkcionalnost omogućuje naredba **ls** (od engl. *list*). Ta naredba jednostavno ispisuje datoteke i direktorije koje se nalaze u aktivnom direktoriju, npr. "ls /usr/sbin". Ako joj se kao argument pridruži putanja, ispisuje popis sadržaj te putanje. Njezino je ponašanje korisno modificirati opcijama, na primjer, "ls -1" ispisuje gornji popis kao stupac s dodatnim informacijama o stavkama. I druge naredbe dolaze s opcijama koje modificiraju njihovo ponašanje, a kod većine njih, popis opcija i kratki opis što rade može se dobiti s --help ili nekad -h.

### 1.2.1 Hijerarhija datotečnog sustava

GNU/Linux naslijedio je i hijerarhiju (strukturu) datotečnog sustava od Unixa, iako ne baš dosljedno te se u nekim dijelovima razlikuje od distribucije do distribucije. Sama hijerarhija

propisana je tzv. standardom hijerarhije datotečnog sustava (engl. *Filesystem Hierarchy Standard*), a osim specifikacije po kojoj treba rasporediti datoteke, standard sadrži i preporuku kako rasporediti osnovne direktorije po participijama ili diskovima. Za razliku od operacijskih sustava koji diskove, participije i vanjske medije označavaju slovima abecede kao zasebna stabla (npr. MS Windows), Unix sve pristupne točke takvih prostora za pohranu i medija smješta unutar istog jedinstvenog stabla. U nastavku slijedi opis te hijerarhije u osnovnim crtama, a za detaljniji pregled vidjeti [1, poglavlje 3], zatim [2], ili dokumentaciju specifične distribucije.

Već spomenuti *root* direktorij (/) u prvoj razini ne sadrži nikakve datoteke, već samo direktorije i to abecednim redom: *bin*, *boot*, *dev*, *etc*, *home*, *lib*, *media*, *mnt*, *opt*, *proc*, *root*, *run*, *sbin*, *sys*, *tmp*, *usr* i *var*. Mogući su i neki drugi direktoriji u ovoj razini, ali to je u pravilu specifičnost distribucije ili pojedinog računala.

Direktorij *boot* sadrži datoteke nužne da bi se operacijski sustav uopće počeo učitavati. Između ostalog, tu se nalazi i sama jezgra sustava, *kernel* i to u datotekama pod nazivom *vmlinuz*.

U direktoriju *bin* nalazi se uži skup izvršnih datoteka naredbi koje su potrebne regularnim korisnicima kad se operacijski sustav pokrene. Za razliku od *bin*, *sbin* sadrži one izvršne datoteke naredbi koje nisu namijenjene regularnim korisnicima, već prvenstveno administratorima.

Zatim, direktorij *lib* sadrži biblioteke i module jezgre sustava nužne da bi se operacijski sustav uspješno startao te da bi programi unutar *bin* i *sbin* uopće radili. Prelaskom na 64-bitne sustave uveo se i *lib64* koji ima istu namijenu kao *lib*, ali za 64-bitne verzije biblioteka.

S obzirom na to da se direktoriji slične namijene *bin*, *sbin* i *lib* nalaze i u *usr*, a ne postoji jasno razgraničenje koje naredbe bi isle u */bin*, a koje u */usr/bin*, odnosno */sbin*/*/usr/sbin* i */lib*/*/usr/lib*, već je raspodjela stvar prakse pojedine distribucije, neke moderne distribucije uvele su praksu da sve stavljaju unutar direktorija *usr*, dok su *bin*, *sbin* i *lib* samo simboličke poveznice (prečaci) na pripadajuće direktorije u *usr*<sup>2</sup>.

Uz prethodno navedeno, direktorij *usr* sadrži i sve ostale korisničke programe te njima pridružene datoteke i resurse koje je osigurala distribucija GNU/Linuxa, bilo pri samoj instalaciji operacijskog sustava, bilo naknadnom instalacijom iz službenih repozitorija. Iznimka je samo poddirektorij *local*, tj. */usr/local*, koji sadrži ono što je neovisno instalirano te taj direktorij distribucija neće ni u kojem slučaju mijenjati ili prebrisati.

Postavke svih programa svojstvenih za računalo na kojem je sustav instaliran nalaze se u direktoriju *etc*. To su u pravilu posebno formatirane tekstualne datoteke koje programi i operacijski sustav koriste kako bi definirali svoj način rada.

Direktorij *var* općenito sadrži podatke koji se mijenjaju kad je operacijski sustav u normalnom radu pa tako programi tu putanju koriste za zapise, baze podataka, predmemoriju, privremene podatke i sl.

*home* sadrži direktorije korisnika, odnosno sve njihove privatne podatke i specifične postavke

<sup>2</sup>Za razloge vidi <https://www.freedesktop.org/wiki/Software/systemd/TheCaseForTheUsrMerge/>

programa koje koriste. Upravo je postavljanje *home* direktorija na zasebnu putanju primjer dobre prakse jer se za cijelu particiju može lako izraditi sigurnosna kopija te ju je moguće sačuvati kad se OS nanovo instalira. No kod osnovnog administratorskog (tzv. *root*) korisnika, *home* direktorij je zapravo na zasebnoj putanji */root* jer mora biti u funkciji i kad je *home* putanja nedostupna. Tom direktoriju može pristupiti samo *root* korisnik.

Direktorij *tmp* namijenjen je za privremene datoteke i sistemskih i korisničkih programa te se u pravilu briše nakon ponovnog pokretanja operacijskog sustava (za razliku od sličnog *tmp* direktorija pod *usr* gdje očekuje dulja trajnost privremenih podataka, pa i nakon restarta).

Putanje u *mnt* direktoriju služe za pristupne točke privremenih uređaja za pohranu podataka, tj. onih koji nisu uвijek dostupni i koje administrator sam priključuje (ili montira, od engl. *mount*) – sustav ili programi nikako ne bi trebali automatski priključivati niti očekivati ikakve pohrane na toj lokaciji. Istovremeno, *media* je namijenjen automatskom montiranju izmjenjivih medija (npr. CD/DVD-ROM diskova, USB prijenosnih memorija) od strane operacijskog sustava.

*dev* je posebna lokacija na kojoj se nalaze datoteke svih uređaja dostupnih operacijskom sustavu. Služe kao sučelje koje korisnik ili korisnički programi mogu iskoristiti da bi pristupili raznim uređajima, točnije njihovim upravljačkim programima.

Lokacija *proc* predstavlja pseudo ili virtualni datotečni sustav stvoren od samog sustava i postoji samo kad je sustav pokrenut. Preko putanja koje se nalaze u *proc* može se pristupiti internim funkcionalnostima jezgre sustava u formi datoteka. Veličina svih takvih datoteka, s obzirom na to da su virtualne, je nula.

### 1.2.2 Pretraživanje datoteka i direktorija

U kontekstu pretraživanja datotečnog sustava, prvo valja istaknuti naredbu **find** koja pretražuje cijelo datotečno drvo za danim upitom unutar radnog direktorija gdje je pozvana ili unutar putanje koja joj je dana kao prvi argumnet, na primjer:

```
[user@localhost ~]$ find /usr -name words
/usr/share/dict/words
/usr/share/doc/words
```

gdje je rezultat u prvom slučaju datoteka imena *words*, a u drugom direktorij *words*. Ako se žele ispisati samo datoteke, dodat će se opcija *-type f*, samo direktorije opcija *-type d* ili samo linkova *-type l*.

Naredba **find** može pretraživati po različitim kriterijima, pa će tako

```
[user@localhost ~]$ find -mtime 0
```

izlistati sve datoteke i direktorije koji su promijenjeni u posljednja 24 sata. Ovo je dano samo za ilustraciju bez puno objašnjenja jer će se kasnije ova naredba obraditi sistematičnije.

Puno bržu pretragu datotečnog sustava, ali skromniju mogućnostima i opcijama, daje

naredba **locate** koja pretražuje već izrađeni indeks datoteka, za razliku od **find** koja prolazi cijelu hijerarhiju. Indeks se može ažurirati naredbom **updatedb** koju je potrebno pokrenuti s administratorskim ovlastima jer u indeks ne može pisati bilo tko ili koju sustav automatski sam pokrene s vremena na vrijeme.. Naredba **locate** tražit će po nazivu datoteke ili putanje dane u argumentu, a -i (ili --ignore-case) može biti korisno kad se ne želi specificirati *case*, odnosno mora li se poštivati veliko i malo slovo prilikom pretrage.

### 1.3 Rad s datotekama i direktorijima

Dok su u prethodnom odjeljku obrađivane osnovne naredbe koje naprsto "opažaju" datoteke i direktorije, u ovom se predstavljaju naredbe koje ih analiziraju i mijenjaju.

Za početak, tu je naredba za (lokalno) kopiranje, **cp**, datoteka i direktorija te joj je potrebno priložiti i dva argumenta: ishodište (engl. *source*) i odredište (engl. *destination*), npr. redak:

```
[user@localhost ~]$ cp /usr/share/dict/words /home/user/words_copy
```

kopirat će datoteku words na drugu lokaciju i spremiti pod imenom words\_copy. Ako je druga putanja samo direktorij, onda će se datoteka prekopirati pod istim nazivom kao što glasi i na ishodištu. Ova naredba ne kopira direktorije sve dok joj se ne doda opcija -r, odnosno --recursive.

Isti način upotrebe stoji i za naredbu za premještanje **mv** (od engl. *move*) uz razliku da datoteka ili direktorij očito ne ostaje na polazištu. Opcija -r u ovom slučaju ne postoji. Ova se naredba koristi i u slučaju promjene imena datoteke jer je ta operacija u suštini ekvivalentna premještanju, samo unutar istog direktorija. Za sustavnije i fleksibilnije preimenovanje pa i više datoteka po nekom obrascu koristi se naredba **rename** s obavezna tri argumenta od kojih je prvi dio naziva ili obrazac u nazivu datoteke koji će se mijenjati, drugi u što će se mijenjati, a treći sam naziv datoteke ili datoteka koje se žele preimenovati. Tako je lako pogoditi kako će završiti sljedeće preimenovanje datoteke *baraka.txt*:

```
[user@localhost ~]$ rename rak nan baraka.txt
```

gdje je svaka asocijacija sa stvarnim mjestima i događajima slučajna. Ova naredba postaje puno korisnija kad se koristi s višezačnicima (engl. *wildcards*), a koji će biti obrađeni kasnije.

U određenim situacijama nije nužno imati nezavisnu kopiju datoteke ili direktorija, već samo nekakav "prečac" (engl. *shortcut*) na izvornik. Osim što takvo rješenje zauzima manje mesta na disku, ponekad je i neophodno ako je cilj istovremeno mijenjati sadržaj datoteka ili direktorija na dvije različite lokacije. Naredba **ln** radi upravo to – stvara prečac, odnosno vezu (engl. *link*) između izvornika i nove putanje. Prvi argument naredbe je putanja do izvornika, a drugi naziv, odnosno putanja veze. Tako će sljedeći unos:

```
[user@localhost ~]$ ln izvorna_datoteka.txt veza_na_izvornu.txt
```

stvoriti datoteku `veza_na_izvornu.txt` koja nije prava datoteka, već samo veza na original i mijenjanjem bilo koje od te dvije, promjena će se vidjeti u obje. Postoje *meke* (engl. *soft*) i *tvrde* (engl. *hard*) veze, gdje se *meka* verzija dobiva s opcijom `-s`, no objašnjenje razlike među njima ostavit će se za kasnije.

Naredba `rm` (od engl. *remove*) uklanja datoteke i to bez mogućnosti povratka te zato treba s tom naredbom uvijek pažljivo. Postoje dvije opcije koje se često koriste i to baš u paru: `-f` (--force) i `-r` (--recursive). Prva mijenja naredbu tako da se ignoriraju nepostojeće datoteke koje su možda dane u argumentu te da se ne postavljaju pitanja (npr. pitanje potvrde uklanjanja), a druga uklanja sve datoteke i direktorije rekursivno na putanji koja je dana kao argument. Ako je dana putanja do specifične datoteke, opcija `-r` ne čini razliku, već ima smisla samo ako je putanja dana do direktorija.

Da bi se napravio novi direktorij koristi se naredba `mkdir` (od engl. *make directory*) s jednim argumentom koji predstavlja ime novog direktorija. Ako je dana putanja i s poddirektorijima kao "prvi/drugi/treci" gdje ne postoji ni jedan od njih, tek uz opciju `-p` bit će napravljena sva tri. U suprotnom će naredba izbaciti grešku da ne može napraviti treci jer ne postoji prvi, ni drugi.

Suprotno od prethodne naredbe, `rmdir` uklanja prazan direktorij naziva istovjetnog argumentu naredbe, a da bi se uklonio direktorij koji sadrži neke datoteke ili direktorije, potrebno je vratiti se na naredbu `rm -rf`.

U ovom nizu naredbi koje brišu datoteke i direktorije, može se navesti i još jedna u istoj kategoriji, a to je naredba `shred` što bi u prijevodu s engleskog značilo *isjeckati*. Naziv je zapravo referenca na rezanje fizičkih papira i dokumenata u komadiće s ciljem da se oni kasnije vrlo teško mogu pročitati, čak i ako se izvuku iz smeća. Istu funkciju ima i `shred` koji datoteku ili uređaj ispunjava besmislenim sadržajem kako bi sadržaj datoteke bilo vrlo teško ili nemoguće kasnije rekonstruirati, a optionalno u zadnjem koraku datoteku može i ukloniti (opcija `-u`). Razlika u odnosu na naredbu `rm` je ta što klasično uklanjanje datoteka uklanja samo referencu na njih u indeksu datotečnog sustava i tako označava prostor na kojem se nalaze dostupnim za novo zapisivanje pa sve dok ne dođe do novog zapisivanja te datoteke i dalje postoje na disku. Za to postoje alati i tehnike da se tako uklonjenje datoteke ipak vrate i ponovno očitaju, no `shred` služi da njihov sadržaj i u tom slučaju ostane uistinu nečitljiv.

Korisno je znati koliko neka datoteka ili direktorij zauzima mesta na disku. Tomu služi naredba `du` (dolazi od *disk usage*, odnosno *upotreba diska*), a daje veličinu prostora koju datoteka ili datoteke specificirane kao njezini argumenti zauzima, odnosno zauzimaju na disku. Treba imati na umu da rezultat nije nužno veličina same datoteke već koliko prostora zauzima njezino držanje na disku, a to ovisi o svojstvima diska, datotečnog sustava i sl. S obzirom na to da bez dodatnih argumenata vraća veličinu u veličina diskovnog bloka (engl. *block size*), a što je najčešće u kilobajtovima, korisnije je dodati odmah opciju `-h` što naredbi govori da izrazi veličinu u čovjeku prihvatljivijem formatu (dolazi od engl. *human readable*) odnosno da automatski, s obzirom na veličinu datoteke, doda sufiks `k` i veličinu ispiše u kilobajtima, `M` u

megabajtima, G u gigabajtima itd. Ako je u argumentu dan direktorij, ona će rekurzivno ispisivati veličine svih datoteka koji se nalaze u tom direktoriju, pa se zato opcijom -s ponašanje mijenja tako da se ispiše sažetak (engl. *summary*), odnosno ukupna veličina svih datoteka u navedenom direktoriju. Na primjer:

```
[user@localhost ~]$ du -sh /usr  
13G   /usr
```

što znači da /usr direktorij ukupno zauzima 13 GB prostora na disku.

### 1.3.1 Provjera datoteka

Naredba **du** može se iskoristiti da se okvirno provjeri je li nešto npr. ispravno presnimpljeno ili preuzeto s Interneta. Tako bi recimo preuzeta video datoteka veličine 1 KB teško bila ispravna ili potpuna. No takvo što neće pomoći u slučaju da je datoteka uništена shredom jer će joj veličina ostati ista, a sadržaj će biti u potpunosti promijenjen. Da bi takva provjera bila pouzdanija, koriste se provjere kontrolnih suma datoteka. To je u pravilu broj ili niz znakova izračunat na temelju sadržaja datoteke, tako da će kontrolona suma dati isti rezultat za istu datoteku koja se nalazi na različitim lokacijama pod različitim nazivima.

Za tu svrhu koriste se tzv. *hash* funkcije koje preslikavaju na jedinstven način sav sadržaj datoteke u niz znakova određene duljine. Tako se provjerava konzistentnost datoteke – ako je *hash*, rezultat *hash* funkcije, identičan na dvije različite lokacije, onda je vrlo vjerojatno i datoteka identična na tim lokacijama, odnosno nije se promijenila pogrešnim kopiranjem ili nekim drugim uzrokom. Najpoznatija *hash* funkcija je MD5, a takav *hash* datoteke generira se naredbom **md5sum**:

```
[user@localhost ~]$ md5sum /usr/bin/bash  
4f7381988b667ac618fb25b5b87b184e /usr/bin/bash
```

što ispisuje heksadekatski broj duljine 32 znaka i predstavlja MD5 *hash* datoteke bash. Ako je broj isti i na drugom računalu, gotovo sigurno se radi o istim datotekama. Ovo je vrlo korisno kad se želi provjeriti je li neka datoteka ispravno preuzeta s Interneta, ne samo zato što je moglo doći do tehničke greške u prijenosu pa je preuzeta datoteka neispravna, već i kao sigurnosna mjera da se provjeri nije li neka treća strana promijenila datoteku u tijeku prijenosa i tako ubacila zlonamjeran kod. Većina GNU/Linux distribucija uz ISO datoteku nudi i *hash* te ISO datoteke da se datoteka nakon preuzimanja može provjeriti. Osim MD5 postoje i drugi *hash* algoritmi, a moderniji i sigurniji su primjeri **sha256sum** i **sha512sum**. Sigurniji u ovom smislu znači da je praktično nemoguće promijeniti datoteku tako da *hash* ostane isti. U slučaju da sigurnost pri ovakvoj provjeri nije bitna, koristi se puno brži algoritam, zvan CRC, i pripadajuća naredba **cksum**. Slučajna pogreška u prijenosu tako se vrlo lako i brzo detektira, ali treba imati na umu da ukoliko netko želi promijeniti datoteku tako da kontrolna suma ostane ista, to s CRC-om može daleko lakše napraviti nego s pravim *hash* funkcijama.

Poznavanje same veličine datoteke, njezine kontrolne sume ili *hasha* ne znači puno ako se želi saznati vrsta datoteke. Čest je običaj da se vrsta datoteke označi nastavkom u samom imenu datoteke. Tako će datoteka koja završava na `.txt` vjerojatno predstavljati tekstualnu datoteku, a ona koja završava na `.mp3` glazbenu datoteku kodiranu u MP3 formatu. Međutim, nastavak je u suštini arbitrarna oznaka vrste datoteke i može se lako izgubiti krivim preimenovanjem ili u nekim slučajevima često se i ne navodi. Recimo na Unixu u pravilu izvršne datoteke nemaju nastavak `.exe`, te najčešće one nemaju nikakav nastavak. Da bi se ipak odredila vrsta datoteke, koristi se naredba **file** koja različitim testovima pokušava klasificirati datoteku danu u argumentu naredbe. Kao rezultat, ako je klasifikacija uspješna, navodi se skup informacija koje opisuju svojstva datoteke: radi li se o tekstualnoj datoteci, izvršnoj ili podatkovnoj; koji ju je program vjerojatno stvorio, koje verzije i sl.

### 1.3.2 Arhiviranje i kompresija datoteka i direktorija

Datoteke se mogu spremiti u direktorije, kao što već slijedi iz činjenice o postojanju direktorija, no nekad je potrebno datoteke spremiti u druge datoteke. Takve datoteke obično se zovu kolekcije ili arhive. Da bi se jedna ili više datoteka spremila u datoteku `moja_arhiva` koristi se naredba **tar** i to na sljedeći način:

```
[user@localhost ~]$ tar -c -f moja_arhiva.tar prva.txt druga.mp3 treca.exe
```

gdje opcija `-c` (ili `--create`) znači stvori, a `-f` (ili `--file`) specificira naziv arhive, a svi argumenti u nastavku, nakon imena arhive, datoteke su koje se žele arhivirati. Češće se dvije opcije `-c` i `-f` pišu zajedno, `-cf`, a ne treba čuditi ako se u nekim primjerima i uputstvima ispusti crtica ispred opcija, što predstavlja tzv. tradicionalni stil opcija kod ove naredbe, makar se ovdje neće koristiti da se sačuva konzistentnost s ostalim naredbama.

Obrnuto postupak, izvlačenja datoteka iz arhive radi se s opcijom `-x` (ili `--extract`):

```
[user@localhost ~]$ tar -x -f moja_arhiva.tar
```

što će *otpakirati* arhivu i sve iz arhive smjestiti u direktorij gdje se arhiva nalazi.

Kao kuriozitet, može se navesti da naziv *tar* originalno dolazi od engl. *Tape ARchive* jer se takva vrsta arhiva koristila za spremanje podataka na magnetske trake.

Takva arhiva želi se u pravilu još i komprimirati da zauzima manje mesta na disku. Tome služi naredba **gzip** koja datoteku danu u argumentu zamijeni komprimiranom verzijom i doda joj nastavak `.gz`. Kompresiju u gzip može direktno izvršiti naredba **tar** tako da joj se specificira opcija `-z`. Sličnu funkciju naredbi **gzip**, vrste i naredbe **bzip2**, **zip**, **xz**, **7z** i dr., a razlikuje ih algoritam kompresije. Neki algoritmi su sporiji, ali rezultiraju većom kompresijom, neki su i brzi i daju veliku kompresiju ali nisu predinstalirani, a neki, makar nemaju dobre karakteristike kompresije kao konkurencija, naprsto su podržani na cijelom nizu platformi, kao recimo **zip**.

Za dekompresiju u pravilu se koristi naredba koja ima prefiks *un* u odnosu na naredbu

za kompresiju, kao **unzip**, **gunzip**, **unxz**, ... Argument tih naredbi mora biti komprimirana datoteka pripadajućeg kompresijskog algoritma, a rezultat je dekomprimirana datoteka.

## 1.4 Rad s tekstualnim datotekama

Naredbe koje će se obraditi: **less**, **more**, **cat**, **zcat**, **nano**, **vim**, **uniq**, **grep**, **wc**, **head**, **tail**, **tee**, **split**, **diff**

## 1.5 Naredbe za podešavanje i kontrolu naredbenog retka

Naredbe koje će se obraditi: **exit**, **reset**, **clear**, **alias**,

## 1.6 Sustav ovlasti i dozvola

Naredbe koje će se obraditi: **id**, **chmod**, **umask**, **su**, **chown**, **chgrp**, **passwd**, **chattr**, **chsh**, **useradd**, **userdel**, **groupadd**, **groupdel**, **groupmod**, **groups**

## 1.7 Rad s procesima

Naredbe koje će se obraditi: **ps**, **top**, **htop**, **jobs**, **bg**, **fg**, **kill**, **pkill**, **killall**, **at**, **crontab**, **nice**, **renice**, **nohup**, **time**

## 1.8 Rad s mrežom

Naredbe koje će se obraditi: **ping**, **traceroute**, **ip**, **wget**, **curl**, **ftp**, **sftp**, **ssh**, **host**, **w**, **who**, **dig**, **nslookup**, **telnet**, **route**

## 1.9 Rad s datotečnim sustavima

Naredbe koje će se obraditi: **mount**, **umount**, **fsck**, **fdisk**, **cfdisk**, **gparted**, **mkfs**, **dd**, **df**

## 1.10 Upravljanje sustavom

Naredbe koje će se obraditi: **shutdown**, **halt**, **reboot**, **uname**, **uptime**, **free**, **hostname**, **dpkg**, **rpm**, **mkswap**, **swapoff**, **swapon**, **modprobe**, **rmmmod**, **lsmod**

## 1.11 Ostalo

Naredbe koje će se obraditi: **whois, whatis, man, help, which, type, sort, echo, sed, tr, join, cmp, bc, xargs, touch, stat, nl, fold, fmt, pr, printf, expr, finger, cal, vmstat, last, mail, mesg, runlevel, wall, watch, write, we, rsync**

# A | Instalacija GNU/Linuxa

Ovaj tečaj prepostavlja da polaznici imaju pristup funkcionalnoj instalaciji GNU/Linuxa. Ako nemaju, ovaj dodatak bi im trebao pomoći da tu prepostavku ostvare na vlastitom računalu.

Postaviti, odnosno instalirati neku od modernih distribucija GNU/Linuxa<sup>1</sup> na osobno računalo nije zahtjevna zadaća. U glavnim se crtama svodi na pripremu USB prijenosne memorije (ili sad već iščezlih optičkih diskova) s nekom od distribucija, učitavanjem pripremljene USB memorije pri novom pokretanju računala (engl. *booting* ili *boot* procesa) i slijedenjem instalacijske procedure ne puno komplikiranije od instalacije uobičajenih programa pod Windowsima do ponovnog pokretanja računala gdje računalo starta novu i funkcionalnu instalaciju GNU/Linuxa.

Međutim, prethodno opisana procedura jednostavna je kada računalo ne sadrži već postojeći operacijski sustav (OS) ili ako se postojeći sustav može slobodno prebrisati. U slučaju da se GNU/Linux želi instalirati paralelno uz postojeći operacijski sustav koji treba ostati u funkciji (tzv. *dual-boot*), prvo je preporučljivo napraviti sigurnosnu kopiju (engl. *backup*) cijelog diska da se minimizira šansa nepovratnog gubitka podataka. Zatim je potrebno oslobođeniti dio diska za novi operacijski sustav. Ovaj je zahtjev već teže ostvariv jer malo tko ima particiju od barem 10-ak GB na disku "viška" koju može obrisati za oslobođenje potrebnog slobodnog prostora. Postoji mogućnost smanjivanja postojećih particija, bilo iz instalacije GNU/Linuxa ili iz samog MS Windowsa<sup>2</sup>, no to je opet dodatna operacija koja kod relativno popunjene particije može trajati podosta, prvenstveno zbog fragmentiranosti podataka. U konačnici, nova instalacija GNU/Linuxa, ako nije drugačije specificirano, modificirat će sam početak osnovnog diska, točnije njegov MBR (engl. *master boot record*), kako bi omogućila vlastito startanje pri pokretanju računala, a pri tom mora u isti izbornik dodati i postojeći operacijski sustav kako bi i on ostao dostupan. Iskustvo pokazuje da i u tom koraku može doći do komplikacija gdje postojeći operacijski sustav ostaje neregistriran u novom izborniku i time nedostupan ili postojeći operacijski sustav pri ponovnom pokretanju sam ukloni učinjene modifikacije pa GNU/Linux ostane nedostupan. U oba slučaja, moguće je popravak MBR-a, no postupak nije trivijalan.

---

<sup>1</sup>Za objašnjenje što su distribucije i kakvih sve ima vidi ??

<sup>2</sup><https://docs.microsoft.com/en-us/windows-server/storage/disk-management/shrink-a-basic-volume>

Srećom, osobna računala u zadnjih pet i više godine imaju dovoljne kapacitete i mogućnosti za pokretanje tzv. virtualnih mašina (VM), specijaliziranih programa koji emuliraju novo računalo kao zasebne okoline unutar već pokrenutog operacijskog sustava, i tako omogućuju instalaciju operacijskih sustava poput GNU/Linuxa unutar MS Windowsa (i obrnuto). Sustav "domaćin" (engl. *host*) tako "ugošćuje" gosta (engl. *guest*), a sama tehnologija zove se "virtualizacija". Hardverski zahtjevi za takav postav okvirno odgovaraju zbrojenim preporučenim memorijskim zahtjevima i jednog i drugog operacijskog sustava (ugrubo je to 8 GB radne memorije za gostovanje uobičajene distribucije GNU/Linuxa s domaćinom MS Windows 10). Uz to, u većini modernih tehnika virtualizacije i za određene gostujuće sustave, računalo treba imati i hardversku podršku ugrađenu u same procesore koja se zove VT-x kod Intelovih procesora, odnosno AMD-V kod AMD-ovih i često ju je nužno eksplicitno omogućiti u postavkama BIOS-a, odnosno UEFI-ja kako bi ona bila dostupna u samom programu za virtualizaciju. Još jedna postavka UEFI-ja može stvarati probleme prilikom pokretanja virtualnih mašina, a to je *Secure Boot*, mehanizam koji provjerava tzv. potpise ugrađenih programa (engl. *firmware*) i jezgri (engl. *kernel*) operacijskih sustava, pa ga je dobro preventivno isključiti kako ne bi stvarao probleme pri pokretanju virtualizacije, makar pri tomu djelomično umanjili sigurnost sustava. Unatoč tim detaljima, opcija instalacije GNU/Linuxa unutar virtualne maštine uvelike pojednostavljuje ostvarivanje pretpostavke prvog paragrafa.

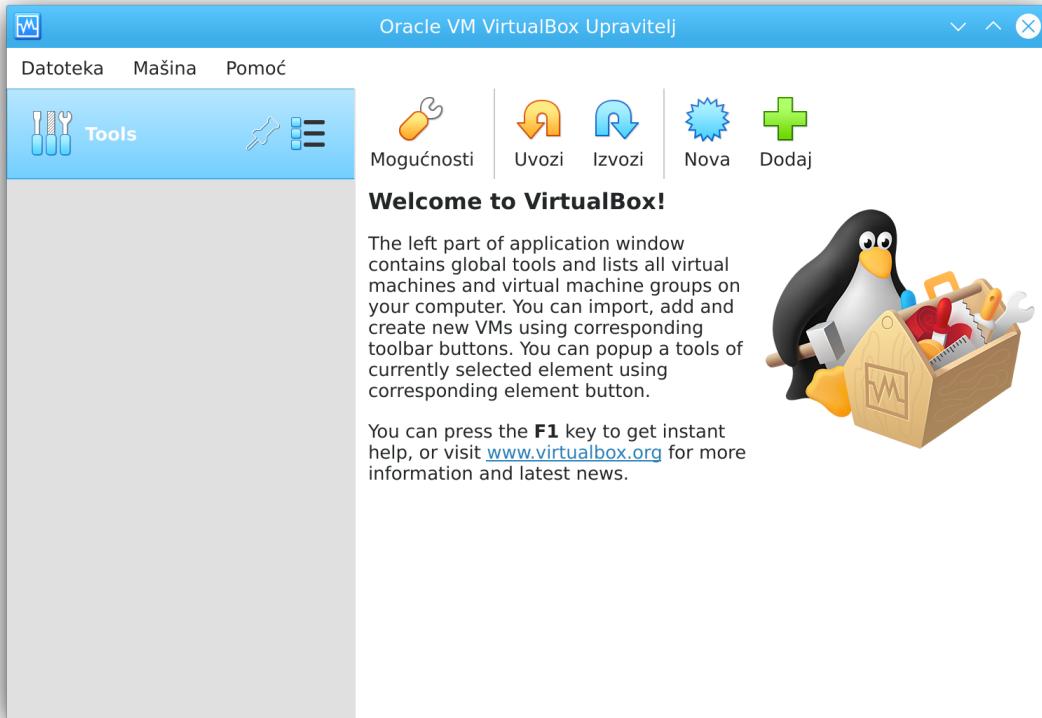
Kao poseban slučaj virtualizacije može se spomenuti još i tzv. WSL (*Windows Subsystem for Linux*). To je prilagođena Microsoftova tehnologija već ugrađena u recentna ažuriranja Windowsa 10 koja omogućuje jednostavnu instalaciju distribucija GNU/Linuxa direktno iz Microsoftovog Storea uz par dodatnih koraka<sup>3</sup>. Ova mogućnost, iako najjednostavnija za postojeće korisnike Windowsa, ne preporučuje se u ovom tečaju jer u nekim aspektima zahtijeva kompromise (npr. grafički sustav tako instaliranog GNU/Linuxa nije dostupan).

U nekim slučajevima spominje se i mogućnost pokretanja tzv. *live* (CD) distribucije GNU/Linuxa koja se izvršava direktno s USB memorije i ne koristi disk računala pa prema tome ni ne ostavlja nikakav trag na računalu nakon isključivanja. No to se u praktičnom smislu nikako ne preporučuje za korištenje već eventualno za kratko testiranje jer su performanse tako pokrenutog operacijskog sustava vrlo spore (u ovisnosti o brzini USB memorije) i generalno ne dopuštaju promjene koje opstaju nakon ponovnog pokretanja.

Od ovdje nabrojanih mogućnosti najveći je komfor samostalna instalacija ili *dual-boot* postav, ako mogućnosti dozvoljavaju. No kako u praksi ne dozvoljavaju na prvo mjesto zasjeda klasična virtualizacija. Ostale mogućnosti mogu koristiti u specifičnim kontekstima, ali treba istaknuti da mogu dovesti do nepredviđenih situacija i dodatnih komplikacija. Idući je odjeljak tako posvećen virtualizaciji.

---

<sup>3</sup><https://docs.microsoft.com/en-us/windows/wsl/install-win10>



Slika A.1: Osnovno sučelje VirtualBoxa.

## A.1 Instalacija na virtualno računalo (VirtualBox)

Iako ima više programa za virtualizaciju, iskustvo pokazuje da za korisnike osobnih računala Oracle VM VirtualBox<sup>4</sup> predstavlja najjednostavnije rješenje koje je pri tomu dijelom slobodan softver i nudi mnogo prigodnih dodatnih mogućnosti. VirtualBox može se instalirati na više različitih operacijskih sustava, uključujući MS Windows, Apple OS X i GNU/Linux. Preuzimanjem i instalacijeom izvršne datoteke za pripadajući sustav te u konačnici pokretanjem VirtualBoxa dobiva se sučelje otprilike kao na slici A.1.

Odabirom "Nova" i ispunjavanjem novootvorenog dijaloga osnovnim podacima: nazivom nove virtualne mašine (npr. Fedora-XFCE), putanjom gdje će se mašina smjestiti unutar datotečnog sustava te vrstom i verzijom gostujućeg operacijskog sustava (npr. Linux i Fedora (64-bit)), pa dalje količinom dostupne radne memorije za gostujući OS (preporuka 4096 MB u slučaju ukupno 8 GB dostupne radne memorije, odnosno 2048 MB ili 1024 MB za manje) dolazi se do koraka gdje je potrebno pripremiti novi virtualni disk za smještaj gostujućeg OS-a. To se ostvaruje tako da se odabere "Napravi...", zatim odabere VDI (*VirtualBox Disk Image*), zatim fiksna veličina i to barem 12 GB s obzirom na potrebe narednih tečajeva, pa do eventualno 20 GB.

<sup>4</sup><https://www.virtualbox.org/>

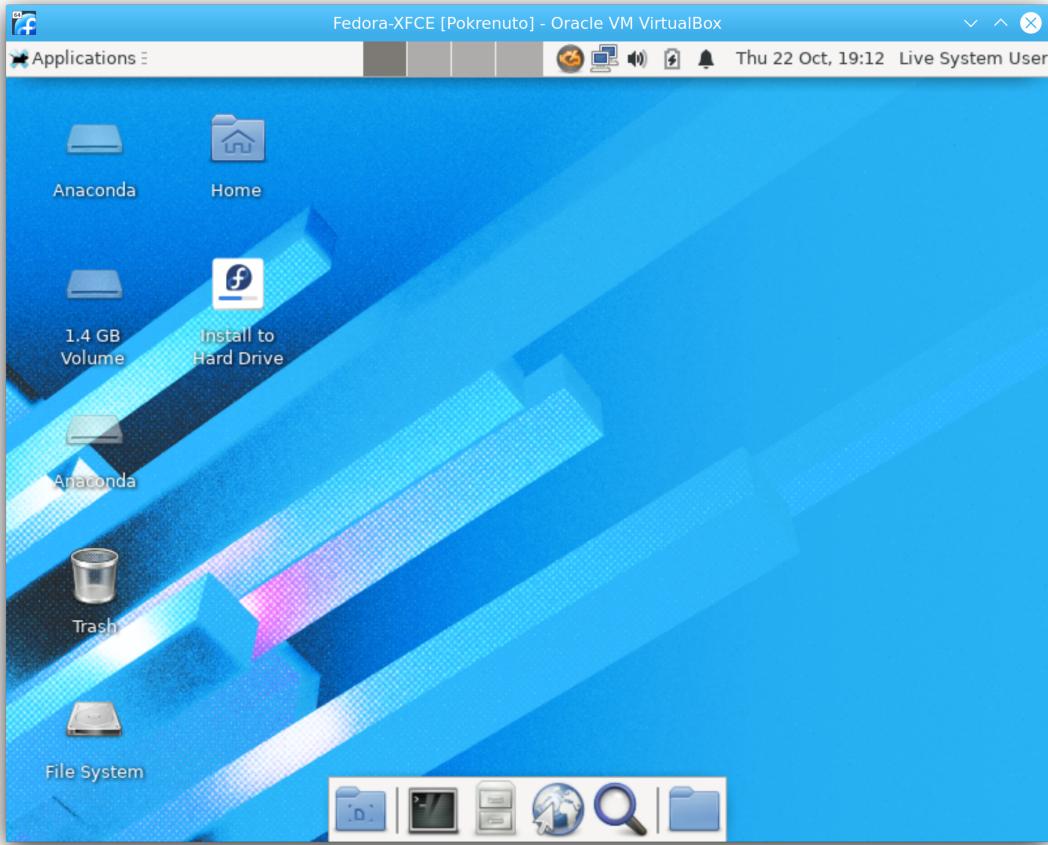
Konačno, kad je virtualna mašina pripremljena može se pokrenuti s "Pokreni" pri čemu se odmah pojavi dijalog za odabir "start-up diska" jer sama mašina još ne sadrži nikakav OS (slika A.2). Dodavanjem i odabirom odgovarajuće već ranije s interneta preuzete ISO preslike (npr. Fedora-Xfce-Live-x86\_64-32-1.6.iso) pokreće se instalacija.



Slika A.2: Odabir ISO preslike neke od ranije preuzetih distribucija GNU/Linuxa.

Kod većine distribucija, nakon učitavanja, prikazat će se *live* izvedba distribucije koja je već ranije opisana, a kroz koju se mogu isprobati osnovne mogućnosti OS-a. Da bi se istu i instaliralo treba odabrati "Install to Hard Drive" (ili slično nazvanu ikonicu) te slijediti proceduru.

Neka od uobičajenih pitanja tijekom instalacijske procedure su postavljanje vremenske zone (npr. *Europe/Zagreb*), osnovnog korisnika i/ili *root* (administratorske) zaporke, jezika tipkovnice i sl. Jedini korak koji inače zahtijeva detaljniji opis jer nije baš samorazumljiv particioniranje je diska. Međutim, s obzirom na to da je ionako cijeli disk zapravo vir-



Slika A.3: Odabir ISO preslike neke od ranije preuzetih distribucija GNU/Linuxa.

tualan, odnosno samo jedna datoteka unutar datotečnog sustava operacijskog sustava domaćina te nema mogućnosti gubitka podataka, nema nekih podreba niti prednosti modificirati već prepostavljenu (engl. *default*) konfiguraciju koju predloži sama instalacijska procedura. U posljednjem koraku započinje kopiranje datoteka i kad ono završi ostaje samo ponovno pokretanje virtualnog računala.

Da bi se izbjeglo ponovno pokretanje instalacije u sljedećem pokretanju VM-a, najjednostavnije je u postavkama VM-a "izbaciti" instalacijski disk (odnosno, u gore opisanom postupku, ISO presliku). Konkretno, u VirtualBoxu, to se može učiniti tek kad je VM isključena te u Postavkama (engl. *Settings*), u sklopu odjeljka Pohrana (engl. *Storage*) treba ukloniti ISO presliku iz virtualnog optičkog uređaja ili naprsto ukloniti cijeli Upravljač pohrane (engl. *Storage device*) kojem pripada virtualni optički uređaj. Pri sljedećem pokretanju VM-a, pokrenut će se instalirana distribucija i time je proces instalacije završen.

## A.2 Instalacija na fizičko računalo

Instalacija neke od distribucija GNU/Linuxa direktno na *fizičko*<sup>5</sup> računalo ne razlikuje se mnogo od gore opisane procedure instalacije na ono virtualno. Ipak, dva koraka u ovom slučaju zahtijevaju više pripreme i pažnje. Prvi je priprema optičkog diska ili USB prijenosne memorije s instalacijskim datotekama koji će se učitati (*bootati*) prilikom pokretanja računala i tako pokrenuti instalaciju, a drugi je partitioniranje diska, pogotovo ako na disku postoje particije koje se ne želi prebrisati jer već sadrže podatke i/ili operacijski sustav. Preporučuje se svakako stvaranje sigurnosne kopije diska prije ikakvog mijenjanja particija da bi se prevenirala neželjena situacija u kojoj dolazi do gubitka podataka. Softverske pogreške koje mogu dovesti do pogrešnog partitioniranja pa tako i gubitka postojećih podataka izuzetno su rijetke, već je najčešća ljudska pogreška, pa ako se ovaj korak radi bez prethodno napravljene sigurnosne kopije (što, ponavljamo, nije preporučeno!), trebalo bi svaki odabir i opciju višestruko provjeriti te izbjegavati u tom slučaju razno-razne "eksperimentalne" konfiguracije.

Za razliku od instalacije na virtualno računalo gdje je bilo dovoljno odabrati odgovarajuću ISO presliku instalacije da bi se ona pokrenula, za instalaciju na fizičko računalo potrebno je pripremiti vanjski podatkovni medij<sup>6</sup>, bilo da se radi o optičkom disku ili USB prijenosnoj memoriji, odnosno bilo kojem mediju s kojeg računalo može pokrenuti *boot* proceduru. Dok se procedura snimanja ISO preslike instalacije na optički medij svodi na učitavanje preslike u program za snimanje CD/DVD-ova te pokretanjem procedure snimanja na prazan medij, procedura pripremanja USB prijenosne memorije malo je komplikiranija. Osim prazne prijenosne memorije veličine barem 4 GB, potrebno je preuzeti i specijalizirani program<sup>7</sup> te ga pokrenuti, odabrati odgovarajuću ISO presliku, a koja je u pravilu ista kao i ona za optički medij, te pokrenuti izradu instalacijske USB memorije.

Što se tiče partitioniranja diska, prvi je problem što GNU/Linux ne raspoznae particije diska po nazivima po kojima ih raspoznae MS Windows ("C", "D", "E" itd.), već koristi druge konvencije tako da je možda najjednostavnije identificirati particije po njihovoj veličini jer bi ih svi operacijski sustavi trebali prikazati identično, pa ako znamo da je particija "C" koja recimo sadrži instalaciju MS Windowsa velika npr. 230 GB, onda takva mora postojati i u pregledu particija tijekom instalacije GNU/Linuxa te nju nikako nećemo modificirati niti formatirati. Ako postoji više particija identičnih veličina, onda preostaje ili utvrđivanje njihove hardverske adrese u oba operacijska sustava ili *montiranjem* particije unutar *live* izvedbe GNU/Linuxa (ako je takva dostupna u željenoj distribuciji) i provjerom sadržaja particija. Identificiranje particija po njihovom poretku nikako se ne preporuča. *Nastavit će se.*

---

<sup>5</sup>Fizičko se ovdje misli na računalo koje prikazuje svoj hardver operacijskom sustavu *neposredno*.

<sup>6</sup>Postoje i druge mogućnosti, ali one nikako nisu preporučene onima koji trebaju ovakav vodič za instalaciju.

<sup>7</sup>Npr. Rufus za MS Windows, <https://rufus.ie/>

# Bibliografija

- [1] L. Wirzenius, J. Oja, S. Stafford, and A. Weeks, "Linux system administrators' guide, version 0.9," 2004. [5](#)
- [2] D. Quinlan, P. R. Russell, and P. R. Russell, *Filesystem Hierarchy Standard*, v3.0 ed., 2015. [5](#)
- [3] A. Robbins, *Unix in a Nutshell*. O'Reilly Media, Inc., 4th ed., 2005.
- [4] T. Adelstein and B. Lubanovic, *Administriranje Linux sustava*. Dobar Plan, Zagreb, 2007.
- [5] N. Marsh, *Introduction to the Command Line: The Fat-Free Guide to Unix and Linux Commands*. 2nd ed., 2010.
- [6] W. Shotts, *The Linux Command Line: A Complete Introduction*. No Starch Press, 2nd ed., 2019.

# Popis naredbi

7z, 10	du, 8, 9	htop, 11
alias, 11	echo, 12	id, 11
at, 11	exit, 11	ip, 11
bash, 2	expr, 12	jobs, 11
bc, 12	fdisk, 11	join, 12
bg, 11	fg, 11	kill, 11
bzip2, 10	file, 10	killall, 11
cal, 12	find, 6, 7	last, 12
cat, 11	finger, 12	less, 11
cd, 4	fmt, 12	ln, 7
cfdisk, 11	fold, 12	locate, 7
chattr, 11	free, 11	ls, 4
chgrp, 11	fsck, 11	lsmod, 11
chmod, 11	ftp, 11	
chown, 11	gparted, 11	mail, 12
chsh, 11	grep, 11	man, 12
cksum, 9	groupadd, 11	md5sum, 9
clear, 11	groupdel, 11	mesg, 12
cmp, 12	groupmod, 11	mkdir, 8
cp, 7	groups, 11	mkfs, 11
crontab, 11	gunzip, 11	mkswap, 11
curl, 11	gzip, 10	modprobe, 11
date, 3	halt, 11	more, 11
dd, 11	head, 11	mount, 11
df, 11	help, 12	mv, 7
diff, 11	history, 3	nano, 11
dig, 11	host, 11	nice, 11
dpkg, 11	hostname, 11	

nl, 12  
nohup, 11  
nslookup, 11  
passwd, 11  
ping, 11  
pkill, 11  
pr, 12  
printf, 12  
ps, 11  
pwd, 4  
reboot, 11  
rename, 7  
renice, 11  
reset, 11  
rm, 8  
rmdir, 8  
rmmod, 11  
route, 11  
rpm, 11  
rsync, 12  
runlevel, 12  
sed, 12  
sftp, 11  
sha256sum, 9  
sha512sum, 9  
shred, 8  
shutdown, 11  
sort, 12  
split, 11  
ssh, 11  
stat, 12  
su, 11  
swapoff, 11  
swapon, 11  
tail, 11  
tar, 10  
tee, 11  
telnet, 11  
time, 11  
top, 11  
touch, 12  
tr, 12  
traceroute, 11  
tree, 4  
type, 12  
umask, 11  
umount, 11  
uname, 11  
uniq, 11  
unxz, 11  
unzip, 11  
updatedb, 7  
uptime, 11  
useradd, 11  
userdel, 11  
vim, 11  
vmstat, 12  
w, 11  
wall, 12  
watch, 12  
wc, 11  
we, 12  
wget, 11  
whatis, 12  
which, 12  
who, 11  
whois, 12  
write, 12  
xargs, 12  
xz, 10  
zcat, 11  
zip, 10