

Radno okruženje



# Što ćemo naučiti?

- Vrste operacijskih sustava
- Virtualni strojevi
- Upravljanje datotekama I direktorijima
- Koristiti terminal I naredbe u terminalu
- Uređivače teksta “VI” I “NANO”
- Upravljanje izvornim kodom GIT
- Web servere
- Razvojne alate

# Operacijski sustavi (OS)

Operacijski sustav je skup osnovnih sustavnih programa koji upravljaju sklopovljem računala (eng. Hardware) radi ostvarivanja osnovnih funkcija računala: ulaz, memoriranje, obrada i izlaz podataka.

Operacijski sustav omogućuje vezu između sklopovlja i korisničkih programa. Mnogi korisnički programi u svom izvođenju pozivaju funkcije koje su sadržane u operacijskom sustavu kroz tzv. API (eng. application program interface)

# Vrste OS-a

Operacijske sustave dijelimo još na sustave za:

- središnja računala
- poslužitelje
- osobna računala
- minijaturene uređaje
- pametne kartice

# Osobna računala

Najpoznatiji operacijski sustavi:

- Linux
- Microsoft Windows
- Mac OS X

# Poslužitelji

Najpoznatiji operacijski sustavi:

- Linux
- Microsoft Windows

# Virtualni strojevi

Virtualni stroj je emulirana računalna sredina koja radi unutar operacijskog sustava i koja se ponaša kao operacijski sustav i može da izvršava računalne programe koji su kompajlirani za tu virtualnu sredinu.

Neki od virtualnih stojeva su:

- Javin Virtualni Stroj
- Macromedia Flash Player
- Virtual PC
- VMware
- VirtualBox

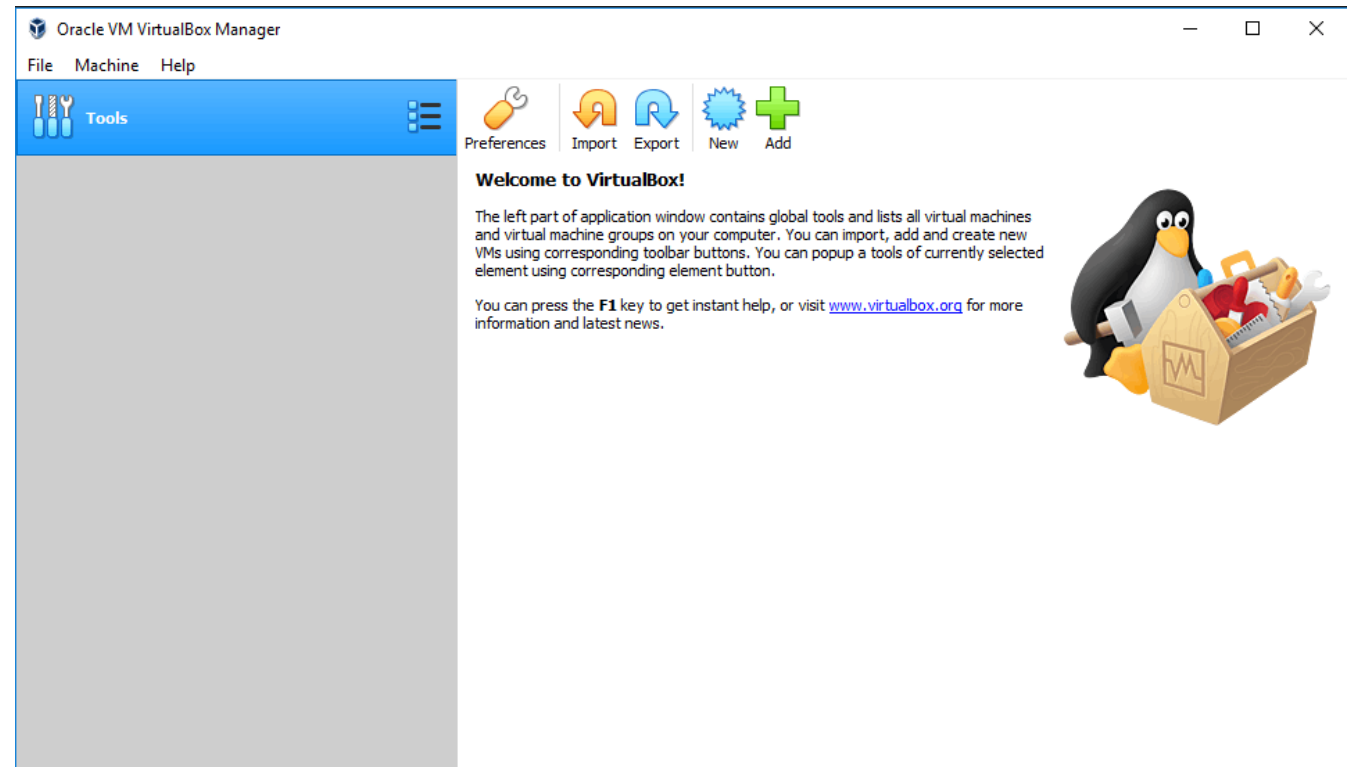
# VirtualBox

VirtualBox može biti instaliran na:

- Linux
- Microsoft Windows
- Mac OS

Link za preuzimanje:

<https://www.virtualbox.org/wiki/Downloads>





# VirtualBox – kreiranje virtualnog stroja

Klikom na ikonu “New” otvara se prozor s desne strane.

- 1) Definiramo ime virtualnog stroja
- 2) Odaberemo tip i verziju operativnog sustava
- 3) Odaberemo količinu memorije
- 4) Odaberemo disk za pohranu podataka
- 5) Kliknemo na gumb “Create”

← Create Virtual Machine

Name and operating system

Name: Backend Developer

Machine Folder: C:\Users\Tomislav\VirtualBox VMs

Type: Linux

Version: Ubuntu (64-bit)

Memory size

4 MB 8192 MB

4096 MB

Hard disk

☐ Do not add a virtual hard disk

☒ Create a virtual hard disk now

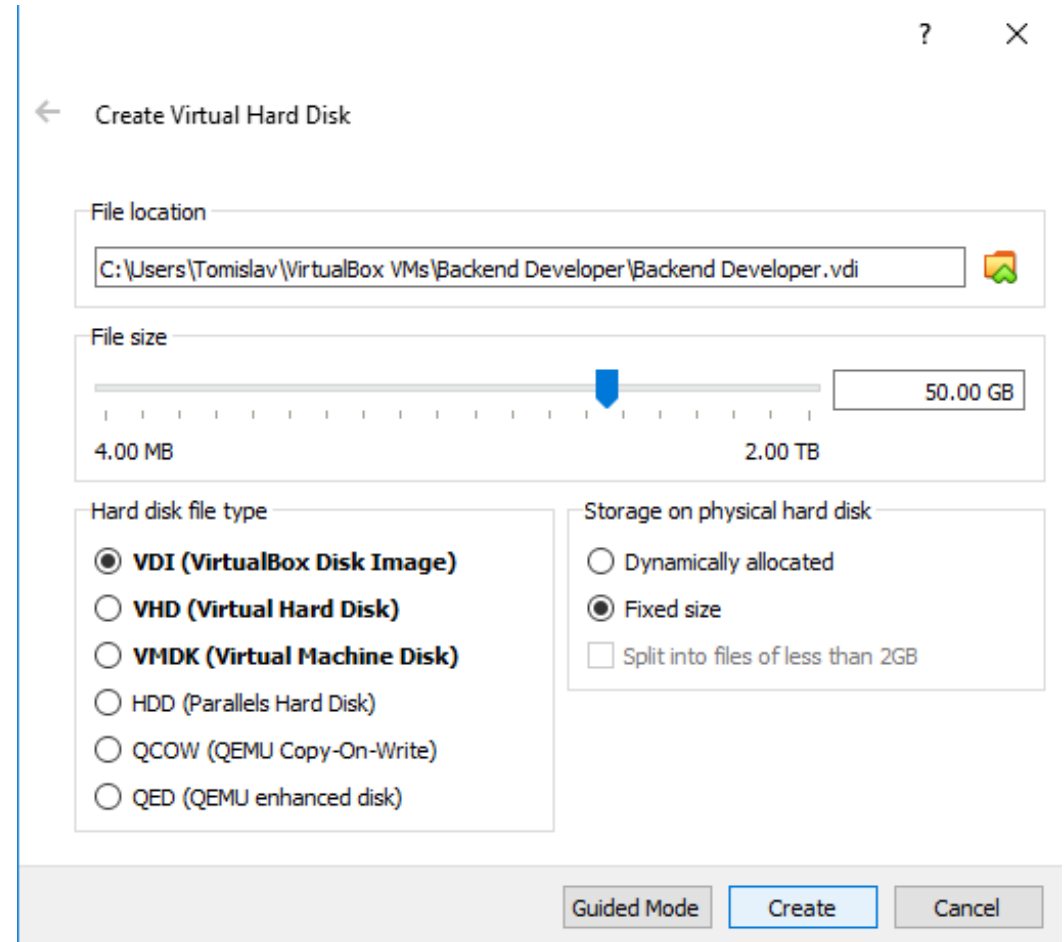
☐ Use an existing virtual hard disk file

Empty

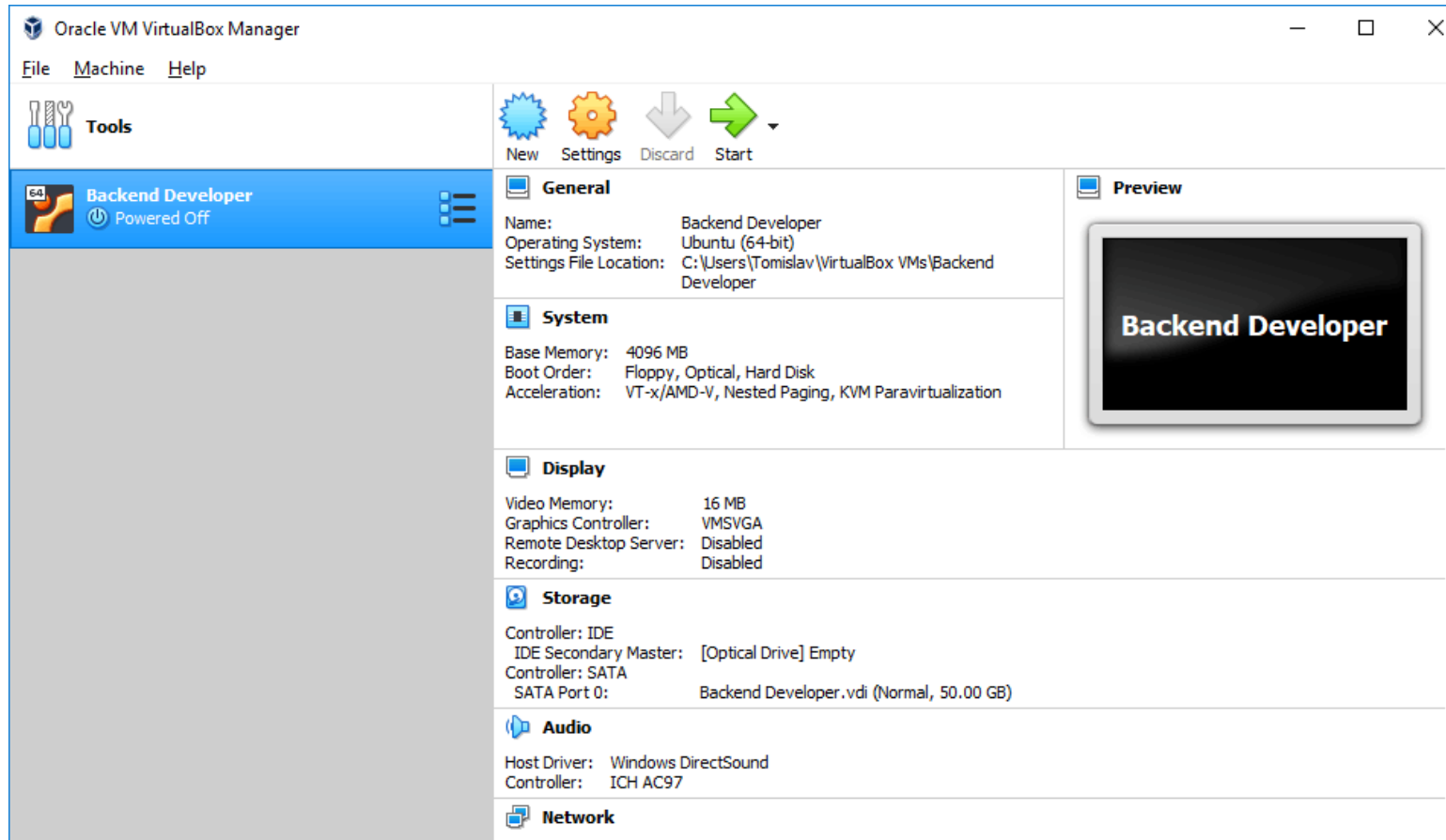
Guided Mode Create Cancel

# VirtualBox – kreiranje virtualnog stroja

- 1) Odaberemo količinu prostora (ovisi o OS-u koji će te instalirati)
- 2) Odaberete vrstu datoteke tvrdog diska
- 3) Odaberete pohranu na fizički tvrdi disk
- 4) Pritisnete gumb “Create”



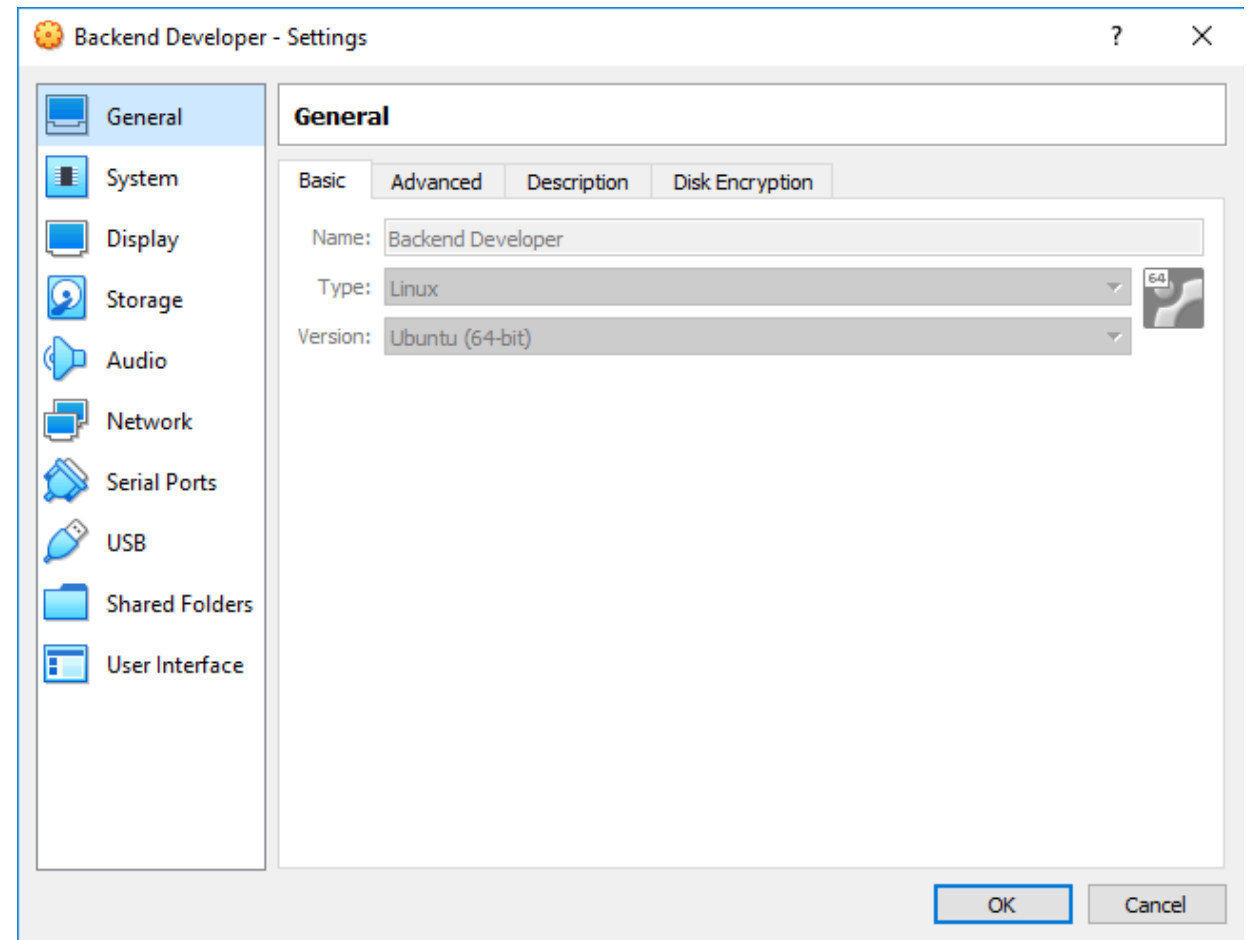
# VirtualBox – kreiranje virtualnog stroja



# VirtualBox – postavke

Nakon uspješnog kreiranja virtualnog stroja, isti možete podesiti pritiskom na gumb “Settings”

Ovdje možete fino podesiti virtualni stroj. Od promjene memorije do promjene broja procesora itd.



# Virtualni strojevi - vježba

- Samostalno kreirati virtualni stroj i pripremiti ga za Linux Ubuntu OS.
- Pokrenuti virtualni stroj i instalirati odabrani OS.

# Datotečni sustavi

- Datotečni sustavi jedna su od stvari s kojom se novi korisnik Linuxa mora upoznati.
- Svaki korisnik bi trebao razumjeti što je datotečni sustav i kako utječe na računalo.
- U srži računala sve je 1 i 0, ali organizacija tih podataka i nije tako jednostavna.
- Bit je 1 ili 0, bajt je sastavljen od 8 bita, kilobajt je 1024 (tj.  $2^{10}$ ) bajta, megabajt je 1024 kilobajta i tako dalje.
- Kada spremite datoteku na tvrdi disk se zapisuje na tisuće 1 i 0 koja se kasnije mogu pročitati.
- Toliko je podataka na tvrdom disku da mora postojati način organiziranja, poput knjižnica koje su sve knjige indeksirale, bez tog indeksa izgubili bi se.
- Knjižnice uglavnom koriste sustav “Dewey Decimal” za organiziranje svojih knjiga. Datotečni sustavi koriste isti način indeksiranja.
- Sustavi kojih su većina korisnika svjesni su oni koje koristi Windows, FAT ili NTFS sustavi.

# Datoteční systavi

File System	Max File Size	Max Partition Size	Journaling	Notes
Fat16	2 GiB	2 GiB	No	Legacy
Fat32	4 GiB	8 TiB	No	Legacy
NTFS	2 TiB	256 TiB	Yes	(For Windows Compatibility) NTFS-3g is installed by default in Ubuntu, allowing Read/Write support
ext2	2 TiB	32 TiB	No	Legacy
ext3	2 TiB	32 TiB	Yes	Standard linux filesystem for many years. Best choice for super-standard installation.
ext4	16 TiB	1 EiB	Yes	Modern iteration of ext3. Best choice for new installations where super-standard isn't necessary.
reiserFS	8 TiB	16 TiB	Yes	No longer well-maintained.
JFS	4 PiB	32 PiB	Yes (metadata)	Created by IBM - Not well maintained.
XFS	8 EiB	8 EiB	Yes (metadata)	Created by SGI. Best choice for a mix of stability and advanced journaling.
GiB = Gibibyte (1024 MiB) :: TiB = Tebibyte (1024 GiB) :: PiB = Pebibyte (1024 TiB) :: EiB = Exbibyte (1024 PiB)				

# Upravljanje datotekama i direktorijima

Ubuntu se pridržava “Filesystem Hierarchy Standard” za imenovanje datoteka i mapa. Ovaj standard omogućava korisnicima i softverskim programima da predvide mjesto datoteka i mapa. Mapa razine korijena predstavljen je jednostavno kosom linijom /. Na razini korijena svi Ubuntu sustavi uključuju ove mape:

Directory	Content
<b>bin</b>	Essential command binaries
<b>boot</b>	Static files of the boot loader
<b>dev</b>	Device files
<b>etc</b>	Host-specific system configuration
<b>home</b>	User home directories
<b>lib</b>	Essential shared libraries and kernel modules
<b>media</b>	Contains mount points for replaceable media
<b>mnt</b>	Mount point for mounting a file system temporarily
<b>proc</b>	Virtual directory for system information
<b>root</b>	Home directory for the root user
<b>run</b>	Run-time variable data
<b>sbin</b>	Essential system binaries
<b>sys</b>	Virtual directory for system information
<b>tmp</b>	Temporary files
<b>usr</b>	Secondary hierarchy
<b>var</b>	Variable data
<b>srv</b>	Data for services provided by the system
<b>opt</b>	Add-on application software packages



# Upravljanje datotekama i direktorijima

**cd** - je naredba koja se koristi za ulazak u postojeći direktorij.

**mkdir** - je naredba koja se koristi za kreiranje novog direktorija.

**touch** - je naredba koja se koristi za kreiranje datoteka sa zadanim dozvolama.

**mv** - je naredba koja se koristi za promjenu imena i premještanje datoteka i direktorija.

**rm** - je naredba koja se koristi za brisanje datoteka.

**rm -r** - je naredba koja se koristi za brisanje direktorija i njezinog sadržaja.

```
root@backenddeveloper: /var/www/html/NoviProjekt
File Edit View Search Terminal Help
root@backenddeveloper:~# cd /var/www/html/
root@backenddeveloper:/var/www/html# mkdir NoviProjekt
root@backenddeveloper:/var/www/html# cd NoviProjekt/
root@backenddeveloper:/var/www/html/NoviProjekt# mkdir photos
root@backenddeveloper:/var/www/html/NoviProjekt# mkdir icons
root@backenddeveloper:/var/www/html/NoviProjekt# touch composer.json
root@backenddeveloper:/var/www/html/NoviProjekt# mv photos images
root@backenddeveloper:/var/www/html/NoviProjekt# mv composer.json packa
ges.jsonroot@backenddeveloper:/var/www/html/NoviProjekt# ll
total 16
drwxr-xr-x 4 root    root    4096 Sep 20 15:27 ./
drwxr-xr-x 3 www-data www-data 4096 Sep 20 15:24 ../
drwxr-xr-x 2 root    root    4096 Sep 20 15:24 icons/
drwxr-xr-x 2 root    root    4096 Sep 20 15:24 images/
-rw-r--r-- 1 root    root      0 Sep 20 15:25 packages.json
root@backenddeveloper:/var/www/html/NoviProjekt# rm packages.json
root@backenddeveloper:/var/www/html/NoviProjekt# rm -r icons/
root@backenddeveloper:/var/www/html/NoviProjekt# ll
total 12
drwxr-xr-x 3 root    root    4096 Sep 20 15:28 ./
drwxr-xr-x 3 www-data www-data 4096 Sep 20 15:24 ../
drwxr-xr-x 2 root    root    4096 Sep 20 15:24 images/
root@backenddeveloper:/var/www/html/NoviProjekt#
```

# Prava pristupa datotekama i direktorijima

- U Linuxu i Unixu sve je datoteka. Direktoriji su datoteke, datoteke su datoteke i uređaji su datoteke.
- Uređaji se obično nazivaju čvor; međutim, to su još uvijek datoteke.
- Sve datoteke u sustavu imaju dozvole koje dopuštaju ili sprječavaju druge da ih pregledavaju, mijenjaju ili izvršavaju.
- Super korisnik “root” ima mogućnost pristupa bilo kojoj datoteci u sustavu.
- Svaka datoteka ima ograničenje s dozvolama kao i ograničenje korisnika s “owner/group” udruživanjem.

Permission	Action	chmod option
read	(view)	r or 4
write	(edit)	w or 2
execute	(execute)	x or 1

User	ls output
owner	-rwx-----
group	----rwx---
other	-----rwx

# Chmod naredba

Naredba koja se koristi prilikom izmjene dozvola je **chmod**. Postoje dva načina za izmjenu dopuštenja, brojevima ili slovima. Upotreba slova je većini ljudi lakša za razumijevanje. Prilikom izmjene dopuštenja budite oprezni da ne biste stvorili sigurnosne probleme.

```
user@host:/home/user# ls -l /etc/shadow
-rw-r----- 1 root shadow 869 2005-11-08 13:16 /etc/shadow
user@host:/home/user#

Permissions:
owner = Read & Write (rw-)
group = Read (r--)
other = None (---)

Ownership:
owner = root
group = shadow
```

# Chmod naredba - primjeri

Evo nekoliko primjera chmod upotrebe slovima.

Prvo stvorite nekoliko praznih datoteka:

```
user@host:/home/user$ touch file1 file2 file3 file4
user@host:/home/user$ ls -l
total 0
-rw-r--r--  1 user user 0 Nov 19 20:13 file1
-rw-r--r--  1 user user 0 Nov 19 20:13 file2
-rw-r--r--  1 user user 0 Nov 19 20:13 file3
-rw-r--r--  1 user user 0 Nov 19 20:13 file4
```

# Chmod naredba - primjeri

Datoteci **file1** dodjeliti ćemo korisniku “**owner**”, dozvolu “**execute**”.

```
user@host:/home/user$ chmod u+x file1
user@host:/home/user$ ls -l file1
-rwxr--r--  1 user user 0 Nov 19 20:13 file1
```

Datoteci **file2** dodjeliti ćemo korisniku “**other**”, dozvolu “**write & execute**”.

```
user@host:/home/user$ chmod o+wx file2
user@host:/home/user$ ls -l file2
-rw-r--rwx  1 user user 0 Nov 19 20:13 file2
```

# Chmod naredba - primjeri

Datoteci **file3** dodjeliti ćemo korisniku “**group**”, dozvolu “**read**”.

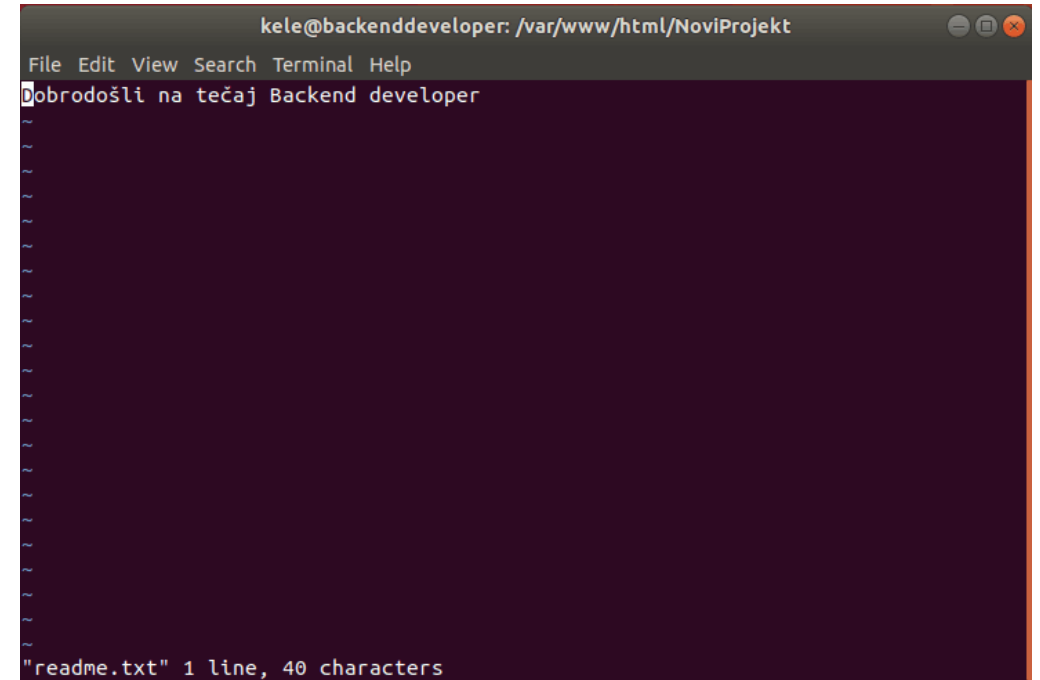
```
user@host:/home/user$ chmod g-r file3
user@host:/home/user$ ls -l file3
-rw----r--  1 user user 0 Nov 19 20:13 file3
```

Datoteci **file4** dodjeliti ćemo svim korisnicima, dozvolu “**read, write & execute**”.

```
user@host:/home/user$ chmod ugo+rwx file4
user@host:/home/user$ ls -l file4
-rwxrwxrwx  1 user user 0 Nov 19 20:13 file4
user@host:/home/user$
```

# Uređivač teksta “VI”

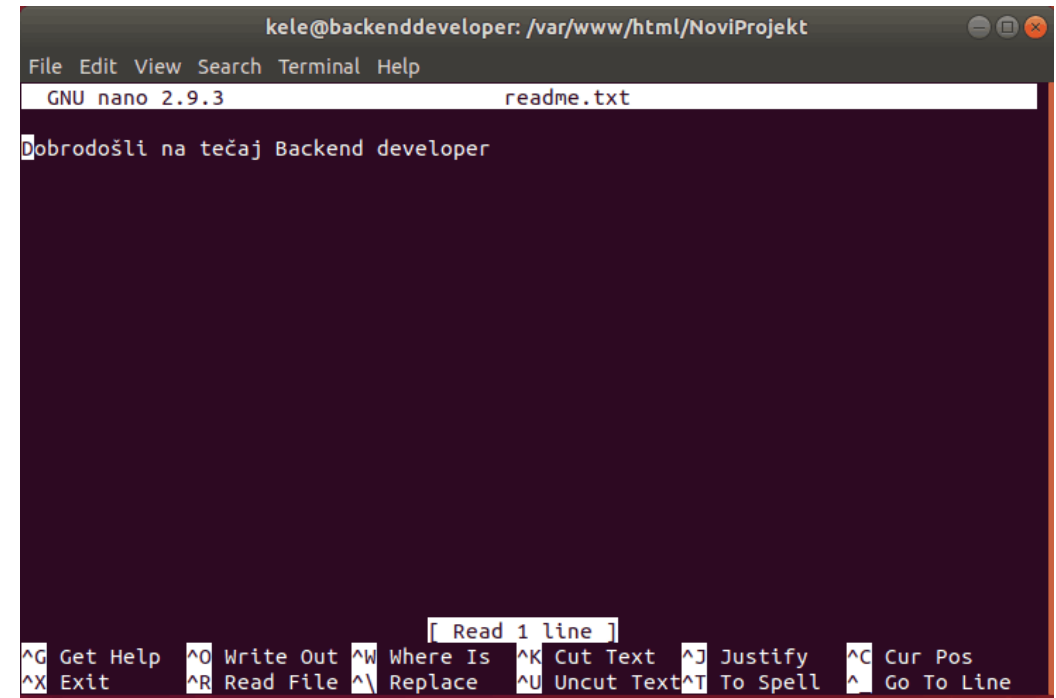
- “VI” je zaslonski uređivač teksta koji je izvorno kreiran za Unix operativni sustav.
- “VI” je modalni uređivač: djeluje ili u načinu pisanja (gdje tipkani tekst postaje dio dokumenta) ili naredbenom načinu (gdje se pritisci tipki tumače kao naredbe koje upravljaju sesijom uređivanja).
- Primjerice, tipkanjem slova i dok je u naredbenom načinu prebacuje uređivač na način za pisanje, ali ponovno tipkanje slova i umetne znak "i" u dokument. Iz načina pisanja, pritiskom na tipku ESC, urednik se vraća u naredbeni način.
- Za uređivanje dokumenta u terminalu je potrebno upisati: **vi 'naziv\_dokumenta'**



The screenshot shows a terminal window with the title bar 'kele@backenddeveloper: /var/www/html/NoviProjekt'. The terminal displays the VI text editor interface. The menu bar at the top includes 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The main editing area has a dark purple background with white text. The first line of the document contains the text 'Dobrodošli na tečaj Backend developer'. Below this line, there are several lines of tilde (~) characters, indicating empty lines in the document. At the bottom of the terminal, a status line reads '"readme.txt" 1 line, 40 characters'.

# Uređivač teksta “NANO”

- “**NANO**” je uređivač teksta za Unix računalne sustave ili radna okruženja koristeći sučelje naredbenog retka.
- “**NANO**” je orijentiran na tipkovnicu, kontrolira se upravljačkim tipkama. Na primjer, **Ctrl + O** sprema trenutnu datoteku; **Ctrl + W** ide u izbornik za pretraživanje.
- “**NANO**” stavlja dvorednu "traku prečaca" na dno zaslona, nabrajajući mnoge naredbe dostupne u trenutnom kontekstu.
- Za uređivanje dokumenta u terminalu je potrebno upisati: **nano 'naziv\_dokumenta'**



```
kele@backenddeveloper: /var/www/html/NoviProjekt
File Edit View Search Terminal Help
GNU nano 2.9.3                               readme.txt
Dobrodošli na tečaj Backend developer

[ Read 1 line ]
^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text ^T To Spell  ^_ Go To Line
```



# Datoteke i direktoriji - vježba

- Kreirati mapu BackendDeveloper.
- U mapi BackendDeveloper kreirajte novu mapu Documentation.
- U mapi Documentation kreirajte datoteku readme.txt
- Datoteku readme.txt otvorite u NANO uređivači te upišite svoje ime i prezime i spremite izmjene.
- Izadite iz uređivača
- Datoteci readme.txt dodjelite sva prava za korisnika “**owner**”, a za korisnike “**group**” i “**other**” dodjelite “**read**” i “**execute**” prava.

# Uvod u GIT

- Git je distribuirani sustav za praćenje promjena izvornog koda tijekom razvoja softvera.
- Dizajniran je za koordiniranje rada među programerima, ali se može koristiti za praćenje promjena u bilo kojem skupu datoteka.
- Njegove prednosti uključuju brzinu, integritet podataka i podršku za distribuirane, nelinearne tijekove rada.
- Git je kreirao Linus Torvalds 2005. godine za razvoj Linux kernela, a drugi su programeri kernela pridonijeli njegovom početnom razvoju.
- Neki od poznatijih servisa koji nude besplatna i plaćena GIT spremišta su: **GitHub**, **GitLab**, **Bitbucket**, itd.

# GIT naredbe

- Općenito, git naredbe uvijek imaju sljedeći format:

```
git <naredba> <opcija1> <opcija2> ...
```

- Za svaku git naredbu možemo dobiti pomoć s:

```
git help <naredba>
```

- Nekoliko postavki je poželjno konfigurirati da bismo nastavili normalan rad. Sva git konfiguracija se postavlja pomoću naredbe `git config`. Postavke mogu biti lokalne (odnosno vezane uz jedan jedini projekt) ili globalne (vezane uz korisnika na računalu).

- Globalne postavke se postavljaju s:

```
git config --global <naziv> <vrijednost>
```

- Lokalne postavke se postavljaju s:

```
git config <naziv> <vrijednost>
```

# GIT naredbe

- Stvaranje novog lokalnog repozitorija:

```
git init
```

- Provjera stanja repozitorija:

```
git status
```

- Dodavanje promjena u indeks:

```
git add
```

- Spremanje novog stanja projekta u lokalni repozitorij:

```
git commit
```

# GIT naredbe

- Stvaranje novog lokalnog repozitorija:

```
git init
```

- Provjera stanja repozitorija:

```
git status
```

- Dodavanje promjena u indeks:

```
git add <datoteka>
```

- Spremanje novog stanja projekta u lokalni repozitorij:

```
git commit -m <message>
```

# GIT repozitoriji

- U GIT-u se repozitoriji dijele na lokalne i udaljene.
- Apsolutno svaki direktorij može postati git repozitorij.
- Ne mora uopće postojati udaljeni server i neki centralni repozitorij kojeg koriste i ostali koji rade na projektu.
- U prošlom poglavlju naučili smo GIT naredbu `git init` koja kreira lokalni repozitorij. Međutim što ako taj lokalni repozitorij želimo pohraniti na udaljeni repozitorij?

# Git repozitoriji

Prvu stvar koju ćemo obraditi je kreiranje udaljenog repozitorija.

Svaki udaljeni repozitorij s kojime će git "komunicirati" mora imati svoju adresu.

Povezivanje lokalnog s udaljenim repozitorijem možemo učiniti s:

```
git remote add origin <adresa_udaljenog_repozitorija>
```

Naredba za "prebacivanje" izmjena iz lokalnog u udaljeni repozitorij:

```
git push
```

Naredba za dohvat promjena iz udaljenog repozitorija ali ne i sjedinjavanje s lokalnim repozitorijem:

```
git fetch
```

Naredba za dohvat i sjedinjavanje promjena s udaljenog repozitorija:

```
git pull
```

# Git repozitoriji - kloniranje

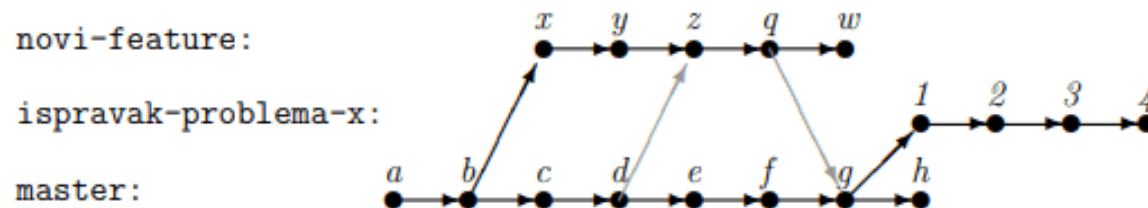
- Kloniranje je postupak kojim kopiramo cijeli repozitorij s udaljene lokacije na naše lokalno računalo. S tako kloniranim repozitorijem možemo nastaviti rad kao s repozitorijem kojeg smo inicirali lokalno.
- Kloniranje udaljenog repozitorija možemo učiniti s:  

```
git clone <adresa_udaljenog_repozitorija>
```
- Kloniranje je kopiranje udaljenog repozitorija, ali tako da novi (lokalni) repozitorij ostaje "svjestan" da je on kopija nekog udaljenog repozitorija.
- Klonirani repozitorij čuva informaciju o repozitoriju iz kojeg je kloniran.



# Git grananje

- Grananje je umnožavanje objekta (poput datoteke s izvornim kodom ili stabla direktorija), tako da se izmjene mogu odvijati paralelno duž više grana.
- Grane omogućuju paralelno razvijanje dijelova softvera.
- Grane omogućuju suradnicima da izoliraju promjene bez destabiliziranja baze koda, na primjer, ispravke pogrešaka, dodavanje novih značajki i integraciju novih verzija. Te se promjene kasnije mogu spojiti (ponovo sinkronizirati) nakon testiranja.
- Primjer grananja:



# Git grananje

- Popis grana projekta:

```
git branch
```

- Stvaranje nove grane:

```
git branch <naziv_nove_grane>
```

- Prebacivanje s grane na granu:

```
git checkout <naziv_grane>
```

- Brisanje grane:

```
git branch -D <naziv_grane>
```

Preuzimanje datoteke iz druge grane:

```
git checkout <naziv_grane> -- <datoteka1> <datoteka2> ...
```

Preuzimanje izmjena iz jedne grane u drugu:

```
git merge <naziv_grane>
```

# GIT - vježba

- U mapi BackendDeveloper inicijalizirajte GIT.
- Kreirajte udaljeni repozitorij na odabranom servisu(GitLab, GitHub, itd).
- Povežite lokalni i udaljeni repozitorij.
- Napravite inicijalni “commit” i promjene pospremte na udaljeni repozitorij.
- Kreirajte novu granu development te se prebacite na istu.
- Datoteku readme.txt otvorite u NANO uređivači te poslije imena i prezimena dodajte tekst po želji i spremite izmjene.
- Napravite “commit” I izmjene spojite s “master” granom.
- Promjene pošaljite na udaljeni repozitorij.

# Apache server

- Apache HTTP Server (Apache HTTP poslužitelj) je besplatni/otvoreni web poslužitelj koji može raditi na raznim operacijskim sustavima kao što su Linux i Windows.
- Apache podržava razne funkcionalnosti, mnoge implementirane kao sastavljene module koji proširuju glavnu funkcionalnost.
- Apache podržava “server-side” programske jezike kao što su Perl, Python i PHP.
- Popularne metode kompresije na Apacheu uključuju vanjski modul proširenja, **mod\_gzip**, implementiran da pomogne u smanjenju veličine (težine) web stranica koje se poslužuju preko HTTP-a.
- Virtualni hosting omogućuje jednu Apache instalaciju za posluživanje različitih web stranica. Na primjer, jedno računalo s jednom Apache instalacijom moglo bi istovremeno poslužiti test.com, test.org, test47.test-server.example.edu, itd.

# Apache server - instalacija

- Instalirajte Apache pomoću upravitelja paketa Ubuntu(**apt**):

```
sudo apt update
```

```
sudo apt install apache2
```

- Cijeli postupak možete pogledati na:

<https://www.digitalocean.com/community/tutorials/how-to-install-linux-apache-mysql-php-lamp-stack-ubuntu-18-04>

- Na Windows operativnom sustavu najbolje je instalirati paket XAMPP koji ima preinstalirani apache server.

- XAMPP možete preuzeti na:

<https://www.apachefriends.org/index.html>

# Apache server – virtualni hosting

- Virtualni hosting jedna je od najčešće korištenih tehnika za hostiranje više domena na jednom poslužitelju kako bi se smanjili potrebni resursi.
- Postoje dvije vrste virtualnog hostinga:
  - **“Name-based Virtual Hosting”**
  - **“IP-based virtual hosting”**
- Razlika između virtualnog hostinga temeljenog na nazivu i IP adrese je korištenje IP-a.
- **“Name-based Virtual Hosting”** koristi jedinstvenu IP adresu za sve domene na poslužitelju, a **“IP-based virtual hosting”** koristi različite IP adrese za svaku domenu.
- Internet stranice gdje možete pogledati postupak postavljanja oba tipa virtualnog hostinga:

<https://www.ostechnix.com/configure-apache-virtual-hosts-ubuntu-part-1/>

<https://poweruphosting.com/blog/ip-virtual-hosting-apache-ubuntu/>

# Apache server – .htaccess

- **.htaccess (Hypertext Access)** je zadani naziv Apacheove konfiguracijske datoteke na razini direktorija.
- Ona pruža mogućnost prilagodbe konfiguracijskih direktiva definiranih u glavnoj konfiguracijskoj datoteci.
- Konfiguracijske smjernice moraju biti u **.htaccess** kontekstu i korisniku su potrebna odgovarajuća dopuštenja.
- Internet stranice gdje možete pogledati sve o **.htaccess** datoteci:

<https://www.askapache.com/htaccess/htaccess-wikipedia/>

# Apache server - vježba

Instalirati Apache server na instancu Linuxa kreiranog u prvoj vježbi.

Postaviti **/var/www/html/BackendDeveloper** mapu kao "root" mapu servera.



# MySQL server

- MySQL je besplatan, open source sustav za upravljanje bazom podataka. Uz PostgreSQL MySQL je čest izbor baze za projekte otvorenog koda, te se distribuira kao sastavni dio serverskih Linux distribucija, no također postoje inačice i za ostale operacijske sustave poput Mac OS-a, Windowse itd.
- MySQL baza je slobodna za većinu uporaba. Ranije u svom razvoju, MySQL baza podataka suočila se s raznim protivnicima MySQL sustava organiziranja podataka jer su joj nedostajale neke osnovne funkcije definirane SQL standardom. Naime, MySQL baza je optimizirana kako bi bila brza nauštrb funkcionalnosti. Nasuprot tome, vrlo je stabilna i ima dobro dokumentirane module i ekstenzije te podršku od brojnih programskih jezika: PHP, Java , Perl, Python...
- MySQL baze su relacijskog tipa, koji se pokazao kao najbolji način skladištenja i pretraživanja velikih količina podataka i u suštini predstavljaju osnovu svakog informacijskog sustava, tj. temelj svakog poslovnog subjekta koji svoje poslovanje bazira na dostupnosti kvalitetnih i brzih informacija.
- MySQL i PHP su osvojili veliki dio tržišta jer su open source i besplatni za koristiti.

# MySQL server - instalacija

- Instalirajte MySQL pomoću upravitelja paketa Ubuntu(**apt**):

```
sudo apt update
```

```
sudo apt install mysql-server
```

- Cijeli postupak možete pogledati na:

<https://www.digitalocean.com/community/tutorials/how-to-install-linux-apache-mysql-php-lamp-stack-ubuntu-18-04>

- Na Windows operativnom sustavu najbolje je instalirati paket XAMPP koji ima preinstalirani apache server.

- XAMPP možete preuzeti na:

<https://www.apachefriends.org/index.html>

# MySQL server - vježba

- Instalirati MySQL server na instancu Linuxa kreiranog u prvoj vježbi.
- Podesiti “root” korisnika da se može prijaviti pomoću lozinke.

# PHP

PHP: Hypertext Preprocessor (ili jednostavno PHP) je programski jezik opće namjene koji je prvobitno dizajniran za razvoj internet aplikacija. Izvorno ga je stvorio Rasmus Lerdorf 1994. godine.

PHP kôd može se izvršiti pomoću sučelja naredbenog retka (CLI), ugrađenog u HTML kôd ili koristiti u kombinaciji s različitim sustavima internet predložaka, sustavima za upravljanje internet sadržajem i radnim okvirima.

PHP kôd obično obrađuje PHP interpreter implementiran kao modul na internet poslužitelju ili kao izvršni Common Gateway Interface (CGI).

Internet poslužitelj izvodi rezultate interpretiranog i izvršenog PHP koda, koji mogu biti bilo koje vrste podataka, poput generiranog HTML koda ili podataka binarne slike.

PHP se može koristiti za mnoge programske zadatke izvan internet konteksta, kao što su samostalne grafičke aplikacije i robotska kontrola bespilotnih letjelica.

# PHP - instalacija

- Instalirajte PHP interpretor pomoću upravitelja paketa Ubuntu(**apt**):

```
sudo apt update
```

```
sudo apt install php libapache2-mod-php php-mysql
```

- Osim što smo instalirali PHP interpretor, instalirali smo i pomoćne pakete kako bi se PHP kôd mogao izvoditi na Apache poslužitelju i komunicirao s MySQL bazom podataka.
- Cijeli postupak možete pogledati na:

<https://www.digitalocean.com/community/tutorials/how-to-install-linux-apache-mysql-php-lamp-stack-ubuntu-18-04>

- Na Windows operativnom sustavu najbolje je instalirati paket XAMPP koji ima preinstalirani PHP interpretor.
- XAMPP možete preuzeti na:

<https://www.apachefriends.org/index.html>

# PHP – php.ini datoteka

- PHP konfiguracijska datoteka, **php.ini**, je konačni i neposredni način utjecanja na PHP funkcionalnost. **php.ini** datoteka se čita svaki put kada se PHP inicijalizira.
- Konfiguracijska datoteka dobro je komentirana i temeljita.
- Ključevi razlikuju velika i mala slova.
- Vrijednosti ključnih riječi ne mogu biti: razmaci i redovi koji počinju sa zarezom.
- Logičke vrijednosti mogu biti predstavljeni sa **1** ili **0**, **Yes** ili **No**, **On** ili **Off** te **True** ili **False**.
- Internet stranice gdje možete pogledati sve o **php.ini** datoteci:

<https://www.php.net/manual/en/configuration.file.php>

[https://www.tutorialspoint.com/php/php\\_ini\\_configuration.htm](https://www.tutorialspoint.com/php/php_ini_configuration.htm)

# PHP - vježba

- Instalirati PHP na instancu Linuxa kreiranog u prvoj vježbi.
- U konfiguracijskoj datoteci podesite **upload\_max\_filesize** na 5M.

# Composer

- Composer je upravitelj paketa na razini aplikacije za programski jezik PHP koji pruža standardni format za upravljanje potrebnim paketima.
- Composer se pokreće iz naredbenog retka i instalira pakete za aplikaciju koju razvijamo.
- Također korisnicima omogućuje instalaciju PHP aplikacija koje su dostupne na "**Packagist**", što je njegovo glavno spremište koje sadrži dostupne pakete. Omogućuje automatsko učitavanje paketa kako bi se olakšala upotreba koda treće strane.
- Postupak instalacije composera možete pogledati na:

<https://www.digitalocean.com/community/tutorials/how-to-install-and-use-composer-on-ubuntu-18-04>



# Composer - vježba

- Instalirati Composer na instancu Linuxa kreiranog u prvoj vježbi.
- U mapi /var/www/html/BackendDeveloper kreirati mapu ComposerExercise i instalirajte paket po želji s **“Packagist”**

# Razvojni alati IDE

- Integrirano razvojno okruženje (IDE) softverska je aplikacija koja računalnim programerima pruža sveobuhvatne pogodnosti za razvoj softvera.
- IDE se obično sastoji od uređivača izvornog koda, alata za automatizaciju prikaza i uklanjanja pogrešaka, itd. Neki IDE-ovi, kao što su **NetBeans** i **Eclipse**, sadrže potrebni kompajler, interpreter ili oba.
- IDE dizajnirana su kako bi se povećala produktivnost programera pružajući usko povezane komponente sa sličnim korisničkim sučeljima. IDE predstavlja jedinstveni program u kojem se vrši sav razvoj. Ovaj program obično nudi mnogo mogućnosti za pisanje, mijenjanje, sastavljanje, raspoređivanje i uklanjanje pogrešaka softvera.

# Parcijalni ispit

- Samostalno kreirati virtualni stroj i pripremiti ga za Linux Ubuntu OS.
- Pokrenuti virtualni stroj i instalirati odabrani OS.
- Instalirati Apache server, MySQL server i PHP interpretor I Composer.
- Postaviti **/var/www/html/Parcijalniispit** mapu kao “root” mapu servera.
- U mapi **Parcijalniispit** inicijalizirajte GIT.
- Kreirajte udaljeni repozitorij na odabranom servisu(GitLab, GitHub, itd) I povežite lokalni i udaljeni repozitorij.
- U mapi **/var/www/html/Parcijalniispit** instalirajte paket po želji s “**Packagist**”
- Kreirajte .gitignore datoteku I dodajte mapu **/vendor**
- Napravite inicijalni “commit” i promjene pospremite na udaljeni repozitorij.

**Hvala na pažnji!**

