

# Vežba 3 i 4

## Raspberry Pi OS (Linux)

<b>1.Uvod</b>	<b>1</b>
<b>2.Shell (bash)</b>	<b>2</b>
3.Konzola ili Terminal	2
4.Osnovne komande u Bash-u	3
5.Zadaci Bash	3
<b>6.Login i sistem korisničkih naloga</b>	<b>4</b>
7.Promena korisničke šifre	4
8.Kreiranje korisnickog naloga	4
<b>9.Tekstualni editori</b>	<b>5</b>
10.Nano - tekstualni editor iz konzole	5
11.Zadaci Nano	5
12.Thonny - tekstualni editor za Python programe	6
13.Geany - tekstualni editor za C i Python programe	6
<b>14.Linux sistem fajlova (filesystem)</b>	<b>6</b>
<b>15.Pristupanje RPi uređaju bežicno - preko SSH</b>	<b>6</b>
16.Pristup preko Putty programa sa Windows-a	6
17.Pristup fajlovima sa RPi preko WinSCP programa	6
18.Kako dozvoliti SSH preko PC-a (upis na MicroSD karticu)	7
19.Kako podesiti Wifi preko upisa na SD karticu	7

## 20.Uvod

Raspberry Pi računari najčešće koriste Raspberry Pi OS distribuciju Linux operativnog sistema. Nekadašnji naziv je Raspbian.

Na ovim vežbama proći ćemo osnove rada na RPi OS-u.

Moguće je i druge distribucije Linux OS-a pokrenuti na Rpi, neki od razloga mogu da budu manje zauzeće memorije, a samim tim i potreba za manjom SD karticom. Ono što je prednost Rpi OS-a je što je unapred prilagođen potrebama i postoji velika zajednica koja kreira i održava biblioteke.

Instalacija RPi OS-a na MicroSD karticu je vrlo jednostavna i uputstvo se nalazi na sledećem linku: <https://www.raspberrypi.org/documentation/installation/installing-images/>

## 21.Shell (bash)

Šel u Linux operativnim sistemima je najvažniji način komunikacije korisnika sa operativnim sistemom. Potpuno je zasnovan na tekstu (text-based user interface). Jednostavno rečeno **shell interpretira korisnički unos i izvršava zadate komande**. *Shell* omogućava kompletnu i preciznu kontrolu nad operativnim sistemom.

Bash (*bourne again shell*) predstavlja najkorišćeniji *shell* u Linux operativnim sistemima i njega ćemo koristiti i na Raspberry Pi OS-u.

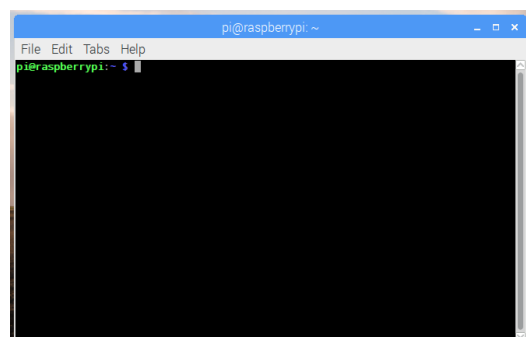
U našem slučaju, ako se RPi uređaju pristupa bežično, korišćenje *shell*-a je daleko jednostavnije i bolje od pristupa grafičkom okruženju. Glavni razlog je brzina prenosa podataka, odnosno tekst se mnogo brže šalje preko mreže nego grafički podaci, ali je potreban i dodatni posao kako bi se omogućilo bežično slanje grafičkih podataka.

Jedna negativna strana korišćenja *shell*-a je ta što je potrebno upamtiti određen broj komandi, kako se bi koristio na efikasan način. Ali nakon nekog perioda korišćenja, korisnik se „navikne“ na korišćenje i često bude mnogo efikasnije od korišćenja miša i grafičkog interfejsa.

### 21.1.1.Konzola ili Terminal

Konzola ili terminal predstavljaju ulaznu tačku za unos komandi u *shell*. U prošlosti su to bili fizički uređaji, koji su omogućavali korisniku da unosi komande, a na ekranu se ispisivao rezultat. U današnje vreme se koriste virtuelni terminali, posebni programi koji otvore prozor i u okviru tog prozora se unosi u *shell* i ispisuju se izlazi. Dakle, jedini posao terminala je da prima unos sa tastature, prosledi taj unos *shell* programu, shell izvrši komande i vrati sadržaj koji se ispiše na ekranu.

Teminal koji se podrazumevano nalazi na Raspberry Pi OS-u je LXTerminal.



## 21.1.2.Osnovne komande u Bash-u

- **ls** – pomoću komande „ls“ se izlistava sadržaj direktorijuma koji je trenutno otvoren
  - komanda: **ls -lha** ispisuje sadržaj direktorijuma mnogo detaljnije
- **pwd** – ova komanda daje informaciju o tome gde se trenutno nalazi terminal, odnosno u kom je trenutno direktorijumu
- **cd** - funkcija „cd“ omogućava kretanje kroz direktorijume
  - **cd <ime\_direktorijuma>** - na ovaj način se ulazi u direktorijum koji se nalazi u aktivnom direktorijumu
  - **cd ..** - prelaz u prethodni direktorijum
- **mkdir <ime direktorijuma>** – funkcija za kreiranje direktorijuma
- **touch <ime fajla>** - funkcija koja osvežava vreme kad je fajl menjan, ili **kreira fajl** ako trenutno ne postoji
- **cat <ime fajla>** - čita sadržaj nekog tekstualnog fajla
- **echo <string koji je potrebno ispisati>** - ispisuje string na terminal
- **„neki sadržaj“ > ime\_fajla.txt** – pomoću znaka „>“ moguće je preusmeriti standardni izlaz u tekstualnu datoteku
- **rm <ime fajla> ; rmdir <ime direktorijuma>** - ovako se brisu fajlovi ili direktorijumi, ali je moguće obrisati samo direktorijum koji je prazan
- **clear** - briše tekst sa terminala, prečica je CTRL + L

## 21.1.3.Zadaci Bash

Za svaki od narednih zadataka ćemo koristiti konzolu na Rpi uređajima.

- 1) Pomoću „pwd“ komande proveriti gde se konzola trenutno nalazi
- 2) Pomoću „ls“ komande videti trenutni sadržaj direktorijuma
- 3) Pomoću komande „touch prvi\_fajl.txt“ napraviti fajl, a potom pomoću „ls -lha“ proveriti vreme kreiranja fajla, veličinu fajla i dozvole koje postoje nad fajlom

- 4) Iskoristiti „echo „neki string“ > prvi\_fajl.txt “ komandu i proveriti da li se promenila veličina fajla
- 5) Pomoću „cat“ ispisati sadržaj prethodno kreiranog fajla
- 6) Pomoću „cd /“ preći u *root* direktorijum Linux OS-a i proveriti sadržaj
- 7) Pomoću komande „cd“ ili „cd ~“ vratiti se u *home* direktorijum korisnika
- 8) Sa „pwd“ proveriti da li je to zaista */home/<username>* direktorijum
- 9) Kreirati direktorijum „pes\_vezbe“ a potom preći pomoću „cd“ u taj direktorijum
- 10) Iskoristiti kmandu „touch fajl{1..10}.txt“ i proveriti sta se desilo, zatim obrisati fajlove od 5 do 10 na slican nacin i proveriti ucinak
- 11) Ostatak fajlova je moguće obrisati pozivanjem komande „rm fajl\*“ gde zvezdica menja bilo koji niz kadaktera. Ako zelimo obrisati sve .txt fajlove komanda bi izgledala ovako:  
„rm \*.txt“

Napomena: biti izuzetno oprezan sa funkcijom „rm“ jer se taj fajl u potpunosti brise i nije ga moguće povratiti

## 22.Login i sistem korisničkih naloga

Linux operativni sistem koristi korisničke naloge i pri pokretanju operativnog sistema vrši se „log in“ u sistem uvek preko nekog korisničkog naloga.

Na RPi OS-u je to „pi“ korisnički nalog podrazumevano.

Svaki korisnički nalog ima svoj */home/<username>* direktorijum i to je direktorijum gde se smeštaju podaci koje samo taj korisnik može da vidi i pristupa im.

Postoji uvek i *root* korisnik koji može da vidi sve i to je uglavnom administratorski nalog. Da bi obični korisnik dobio *root* privilegije koristi se alat *sudo*. Alat *sudo* omogućava korisniku da pristupa celom filesistemu i da instalira programe, ali je potrebno da ukuca svoju šifru i da se taj nalog nalazi u *sudoers* grupi. Treba biti pažljiv pri korišćenju tog alata jer važi izreka „*with great power comes great responsibility*“ i moguće je pokvariti sistem. U zavisnosti od sistema, nekada nije potrebno ni ukucati šifru kod poziva *sudo-a* ali svakako treba biti pažljiv.

### Promena korisničke šifre

Na Raspberry Pi OS-u se uvek koristi ista početna šifra (raspberry), tako da je korisno promeniti je ako je privatnost podataka bitna.

Da bi se promenila šifra trenutno ulogovanog korisnika potrebno je otvoriti terminal i ukucati komandu: „passwd“

A da bi se promenila šifra bilo kog korisnika ukucati: “sudo passwd <username>”

## Kreiranje korisnickog naloga

Kreiranje korisničkog naloga se preporučuje kada će više korisnika da koristi isti računar.

U narednih nekoliko koraka ćemo kreirati korisnički nalog, a onda se i ulogovati kao taj korisnik na Raspberry Pi.

Komanda kojom se kreiraju korisnici je “*adduser*”.

Ukucati komandu “sudo adduser <željeno kor. ime>” i sistem će pitati koju šifru želite dodeliti tom korisniku kao i kreirati /home/<željeno kor. ime> direktorijum.

Podrazumevani nalog *pi* je član *sudo* grupe i može pomoću te komande da izvršava naredbe kao *root* ali novokreirani nalog nema tu opciju.

Ako je potrebno dodeliti te opcije novom korisniku moguće je koristeći komandu:  
“sudo adduser <kor. ime> sudo”

Moguće je obrisati korisnika pomoću komande:  
“sudo userdel -r <kor. ime>”

Podrazumevano je logovanje korisnika “pi” ali je to moguće isključiti u Raspberry Pi Configuration alatu, i onda će pri podizanju sistema moći da se bira koji nalog će se logovati.

Za ove vežbe, radi jednostavnosti, najbolje je da svi koristimo isti korisnički nalog “pi”, i da ostavimo da se automatski uloguje na njega. Po želji promeniti eventualno lozinku.

## 23. Tekstualni editori

### Nano - tekstualni editor iz konzole

Nano je izuzetno koristan alat jer je moguće menjati fajlove iz konzole.

Fajl se otvara tako što se u konzoli ukuca: “**nano <ime fajla>**” i otvara se relativno skroman editor koji je najbolje koristiti za male izmene.

Nakon menjanja fajla iz programa se izlazi pomoću **CTRL + X** i program će pitati da izmene da budu sacuvane. Nakon pritiskanja karaktera **Y** program pita pod kojim imenom da se sacuva i nakon pritiska Enter-a tekst biva sacuvan.

Napredniji program za editovanje iz konzole je **Vim** ali zbog relativno komplikovanog koriscenja neće biti objasnjavan. To je jedan editor iz konzole koji nakon podesavanja može da parira i velikim programima za editovanje bilo kakve vrste koda.

## Zadaci Nano

- 1) Napisati kratak "Hello world" program u C programskom jeziku, a potom ga u konzoli kompajlirati pomocu komande: `gcc -o <ime_izvrsnog_fajla> <ulazni .c fajl>` i isprobati ga pomocu `./<ime_izvrsnog_fajla>`
- 2) Napisati kratak Python program koji 10 puta ispisuje tekst koji je unet preko terminala, a potom ga isprobati pozivajuci: `python <ime .py fajla>`

## Thonny - tekstualni editor za Python programe

Uz Raspberry Pi OS dolazi preinstaliran Thonny editor koji je sasvim zadovoljavajuci za nase potrebe.

Takodje je i Python vec instaliran, tako da je moguće odmah krenuti sa radom.

Isprobati Thonny editor ponavljajuci zadatak iz prethodne glave.

## Geany - tekstualni editor za C i Python programe

RPi OS takodje dolazi sa preinstaliranim Geany editorom, koji je predvidjen za rad sa C fajlovima. Ovaj editor se moze koristiti i za Python programe i radi izuzetno lepo.

Isprobati Geany editor da se napisu C i Python programe iz prethodnih glava.

## 24.Linux sistem fajlova (filesystem)

## 25.Pristupanje RPi uređaju bežicno - preko SSH

SSH ili Secured Shell je mrežni protokol koji omogućava pristup drugom računaru preko računarske mreže. Karakteristika ovog protokola je velika sigurnost prenosa podataka. Moguće je pristupiti *shell*-u drugog računara, kopirati podatke, kao i generički prenos podataka.

Kako bi se pristupilo nekom Linux računaru preko SSH potrebno je poznavati IP adresu uređaja kao i korisničko ime i lozinku naloga na tom računaru.

Primer pristupa RPi-ju sa nekog drugog Linux računara bi bio: `ssh pi@192.168.0.29` i potom ukucati šifru za korisnika "pi".

## Pristup preko Putty programa sa Windows-a

Sa Windows računara je potrebno korišćenje dodatnog softvera. Program Putty omogućava pristup preko SSH.

Potrebno je ukucati IP adresu RPi-ja, i onda ukucati podatke o nalogu.

## Pristup fajlovima sa RPi preko WinSCP programa

WinSCP je vrlo zgodan program koji omogućava pristup celokupnom Filesystem-u RPi-ja. Takođe je potrebno znati IP adresu, korisničko ime i lozinku.

## Kako dozvoliti SSH preko PC-a (upis na MicroSD karticu)

Moguće je dozvoliti privremeni pristup preko SSH samo preko menjanja sadržaja MicroSD kartice. Ovo je izuzetno korisno prvi put kad se pristupa RPi jer je podrazumevano SSH zabranjen.

Potrebno je dodati prazan fajl bez ekstenzije pod imenom: "ssh" na karticu na kojoj se nalazi sistem. Nakon toga vratiti karticu u RPi i nakon podizanja sistema moguće će biti pristupiti RPi preko SSH.

## Kako podesiti Wifi preko upisa na SD karticu

- Put the card in a drive and mount it.
- Go to the partition called boot.

Create an empty file called `ssh` just using:

```
touch ssh # Works in Linux/macOS
```

```
type NUL >> ssh # Works in Windows
```

Create a file called `wpa_supplicant.conf` with your WiFi settings - be very sure that Windows hasn't added any `.txt` extension:

```
country=US
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
network={ ssid="NETWORK-NAME"
          psk="NETWORK-PASSWORD" }
```

Once it works, and you can `ssh` successfully into the Raspberry Pi, be sure to run:

```
sudo raspi-config
```

and permanently enable `ssh` for subsequent reboots.