Sri Lanka Institute of Information Technology

Linux Administration Case Study

IE2012

Systems and Network Programming

D.D.K. ELESINGHE

Table of Contents

1. Basics of Linux Environments

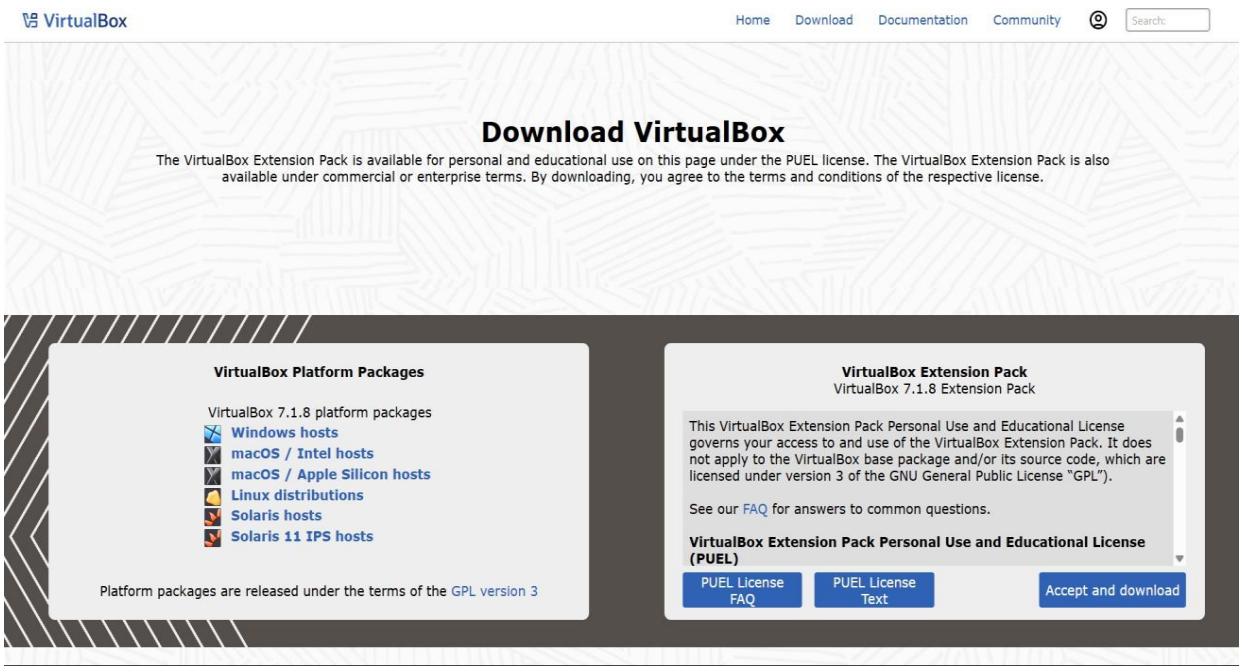
This section explains how to set up and install a virtual machine, and describes basic navigation and file manipulation commands, as well as how to manage system information and user accounts using Linux commands.

1.1 Virtual Machine setup

1.1.1 Virtual Machine setup

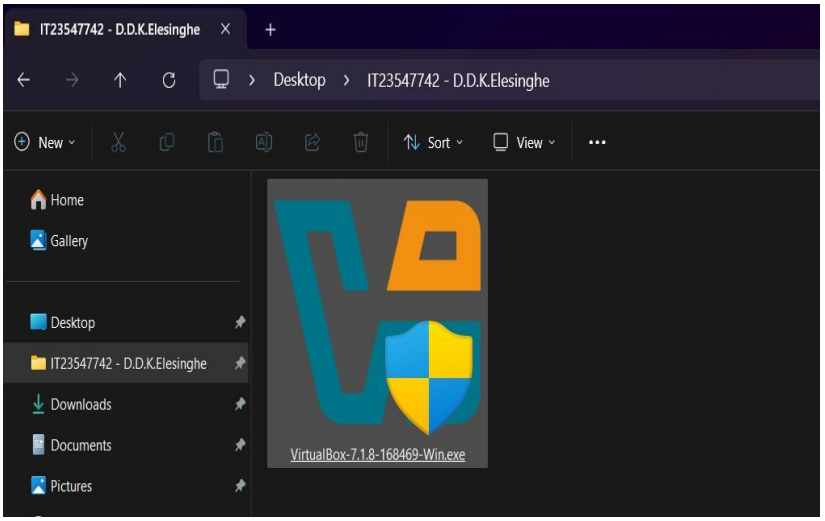
This section details the essential steps for installing and configuring a virtual machine, using Oracle VirtualBox as the reference.

i. **Visit to https://www.virtualbox.org/wiki/Downloads**



ii. **Select the platform package that suits your host operating system. I have chosen windows hosts as my host operating system.**

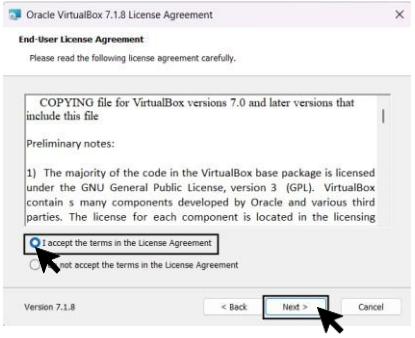
iii. **Save the file to your preferred location.**

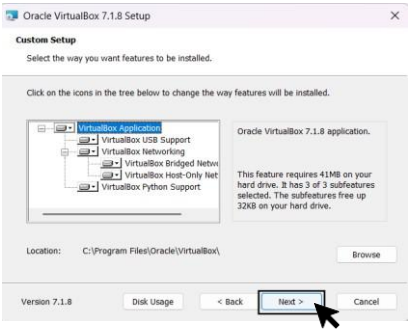


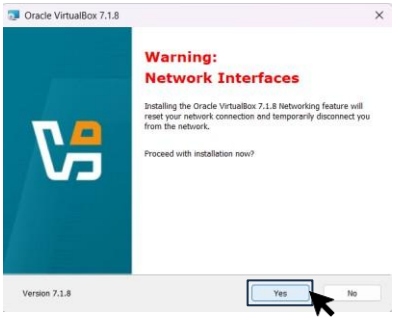
iv. **Double click on the downloaded setup file and click next on the setup wizard to begin installation**.

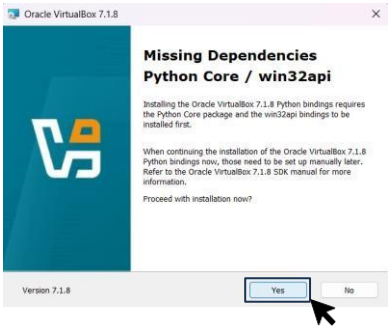
v. Installation

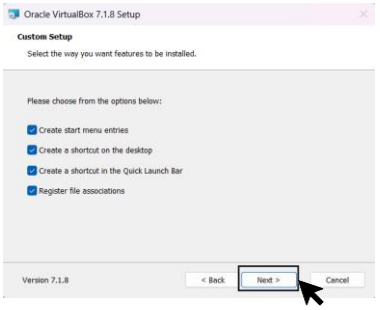
- Accept the terms and the license agreement and keep all other default options selected.
- Proceed by clicking next.
- Click Install when prompted.
- Wait until the completion of installation.
- Check “Start Oracle VirtualBox 7.1.8 after installation.”
- Click Finish to complete installation.

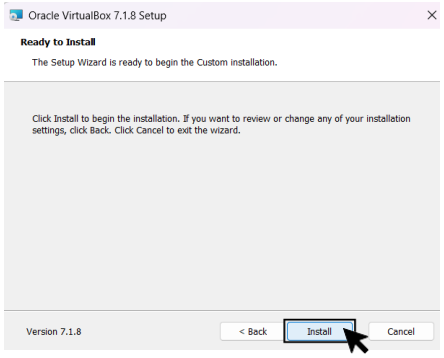


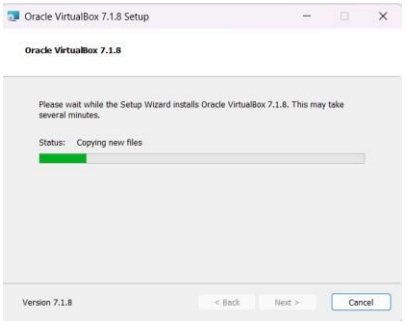


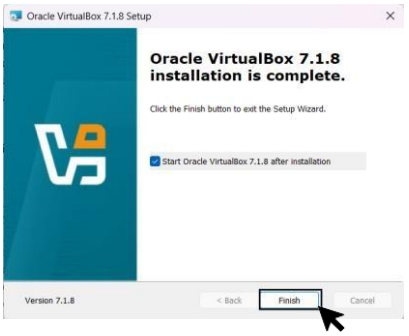


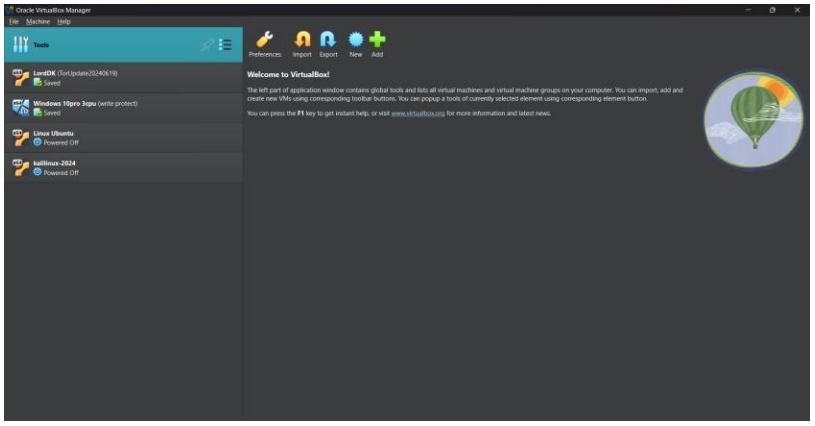








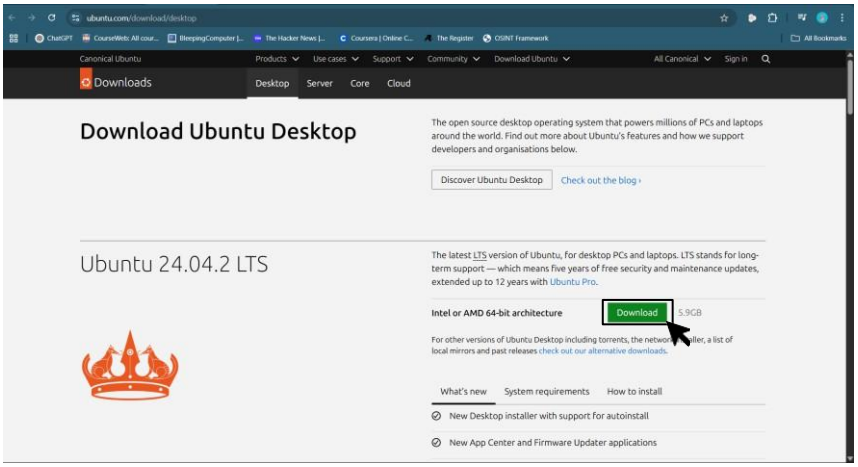




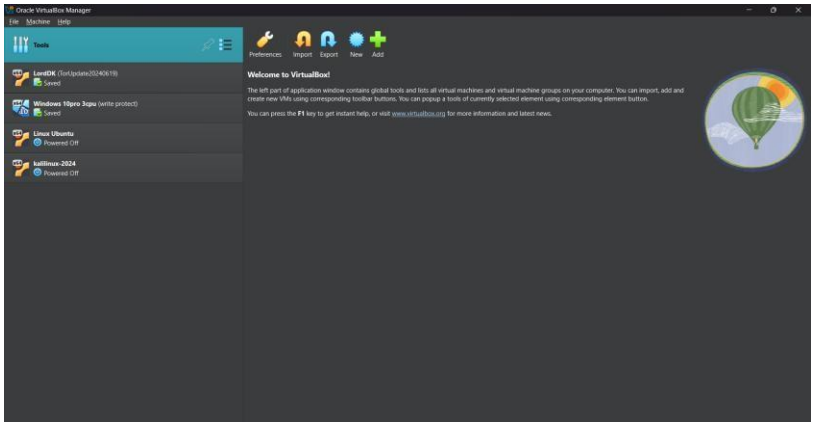
1.1.2 Download of a Linux distribution (Ubuntu)

This section describes the main steps needed to install and set up Ubuntu using Oracle VirtualBox.

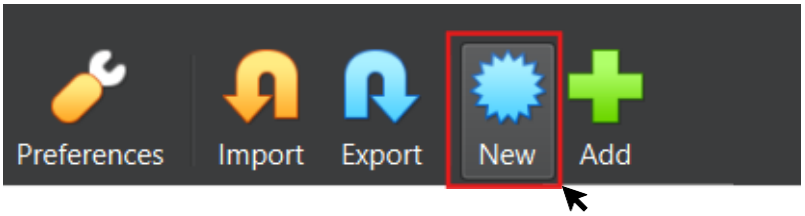
i. Navigate to https://ubuntu.com/download/desktop and download the latest version of Ubuntu.



ii. Open VirtualBox

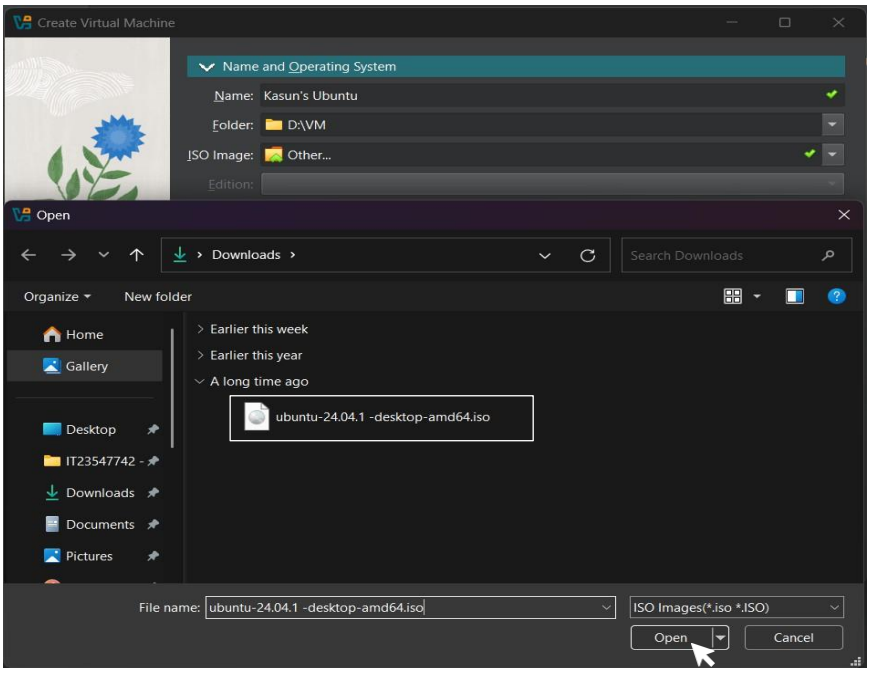


iii. Click the "New" Icon.



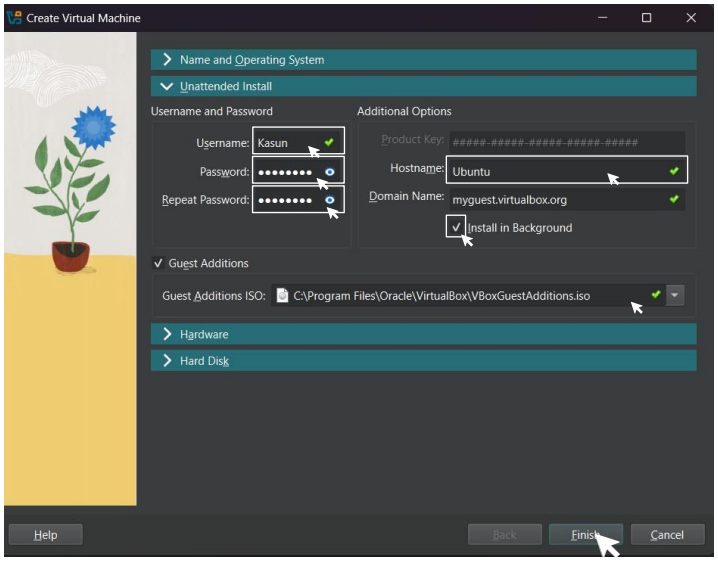
iv. Fill the details in "Name and Operating System" section,

- Enter a name for the machine that you want to create (Ex:-Kasun's Ubuntu).
- In the Folder field choose a folder that you want to save the virtual machine.
- In the "ISO image" field, open and select an ISO image you have previously downloaded.



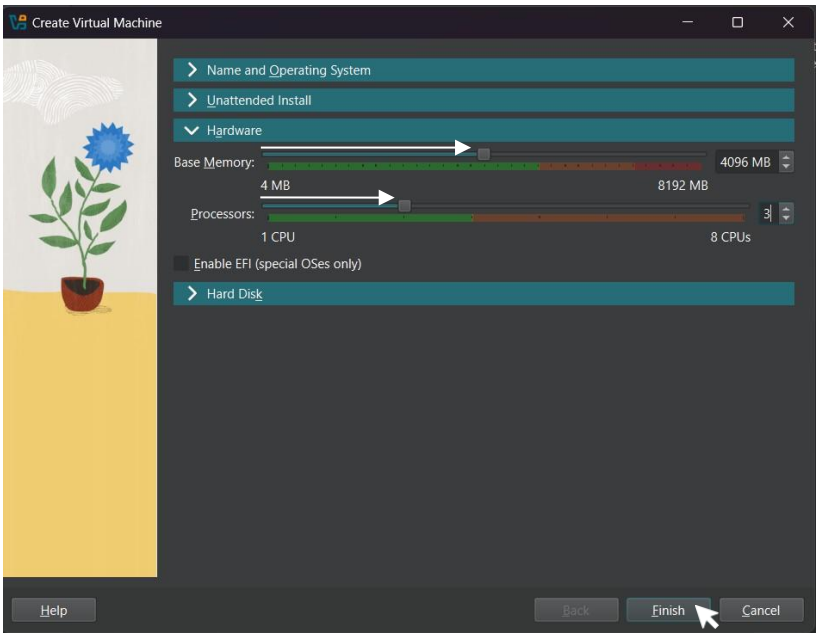
v. In the "Unattended Install" section,

- Set a username and a password in the required fields.
- Check "Install in background"
- Check "Guest Additions"
- If the path is not selected already the default path is **`C:\Program Files\Oracle\VirtualBox\VBoxGuestAdditions.iso`**



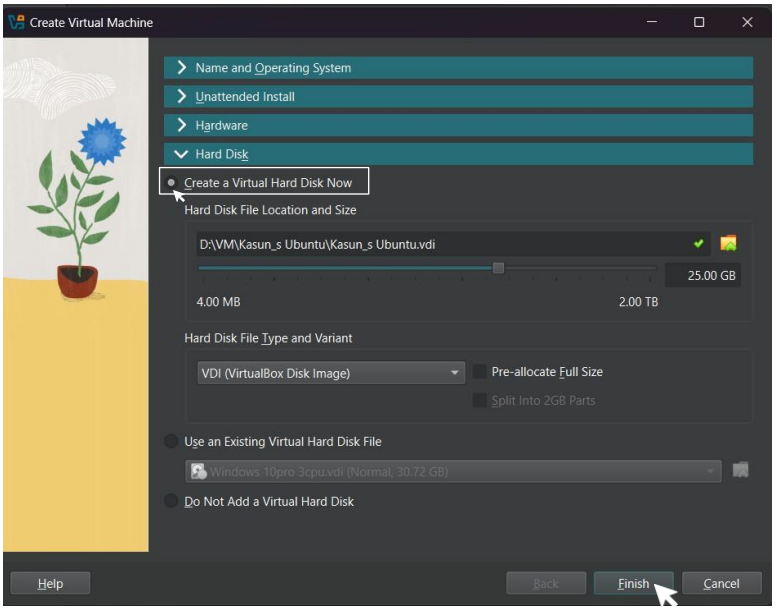
vi. In the “Hardware” section,

- Allocate “Base Memory” (RAM) while keeping the slider within the green zone. A minimum of 2GB is required, but it is recommended to allocate at least 4GB for smoother performance, especially when running GUI-based applications.
- Set the number of “Processors” within the green zone a minimum of 1 CPU core is necessary, though assigning 2 or more cores is a best practice to ensure better responsiveness and multitasking capability during system use.

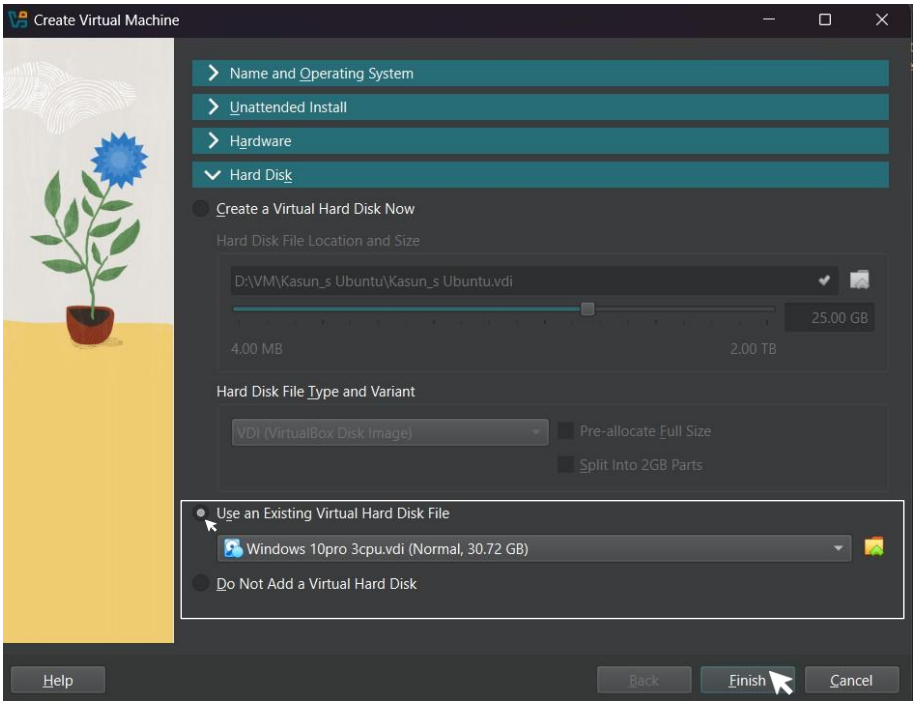


vii. In the “Hard Disk” section,

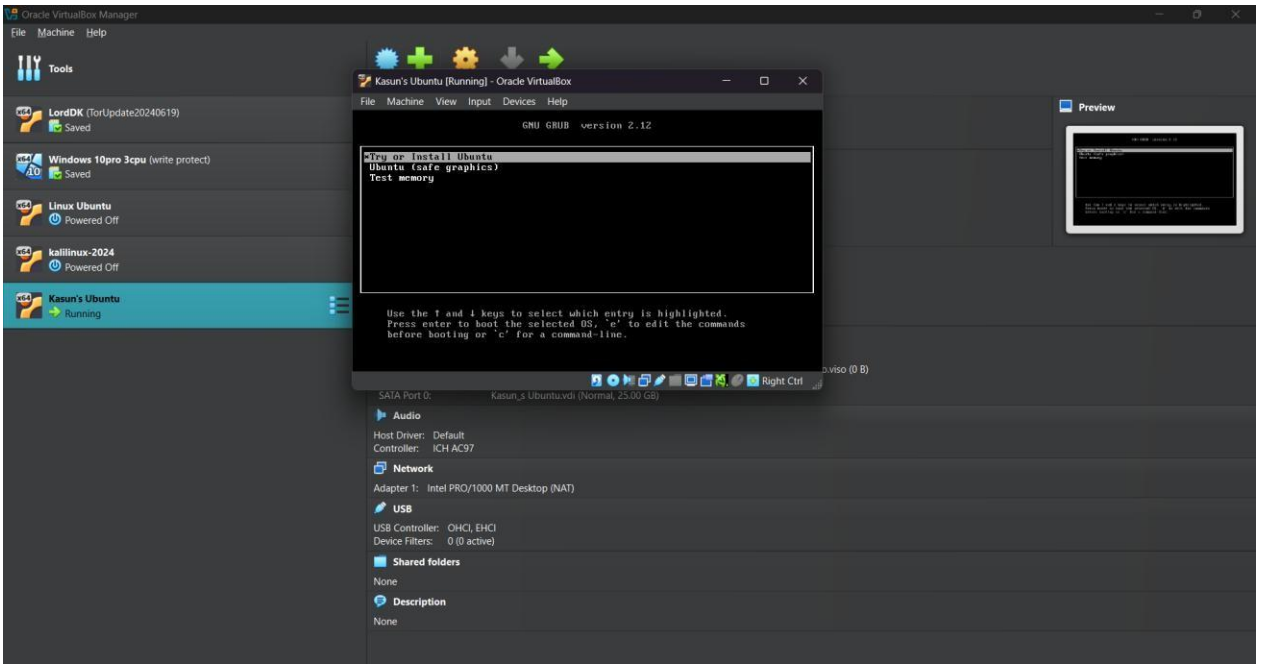
- Select "Create a virtual hard disk."
- Set at least 25GB (Minimum).
- Check “Pre-allocate Full Size” to reserve space now, or leave it unchecked to let it grow as needed (dynamically).



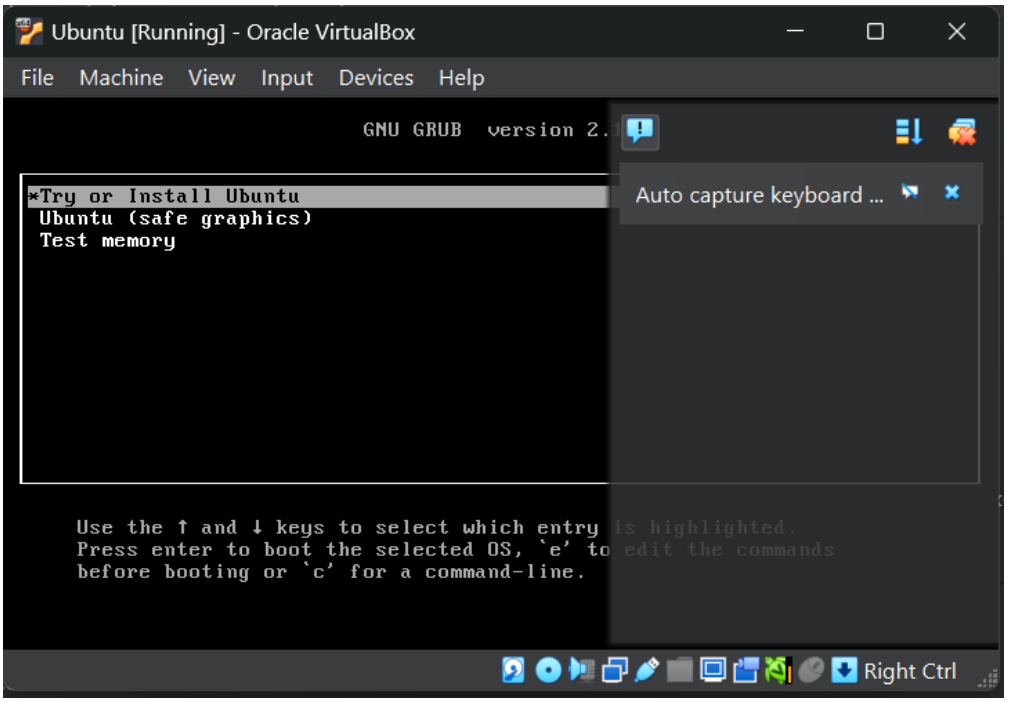
- Also, you can use an existing Virtual Hard Disk ***if you have one***.



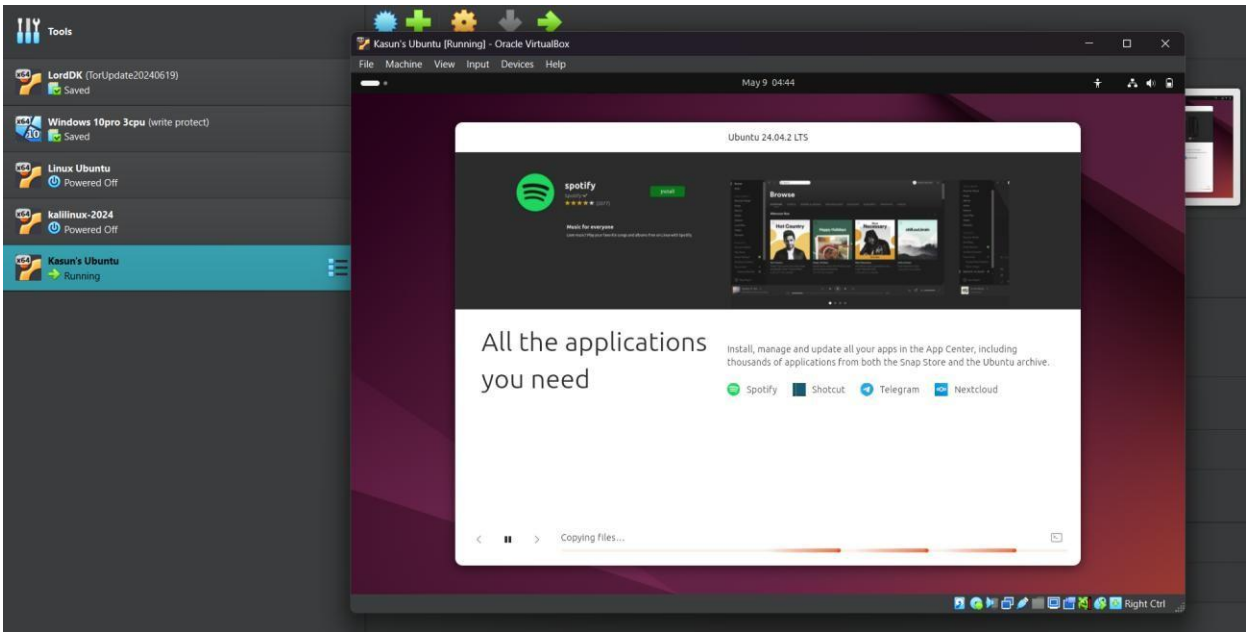
viii. Click the finish button and the VM will start automatically.



ix. **“Try or Install Ubuntu” option is selected by default. Hit “Enter” key to execute.**

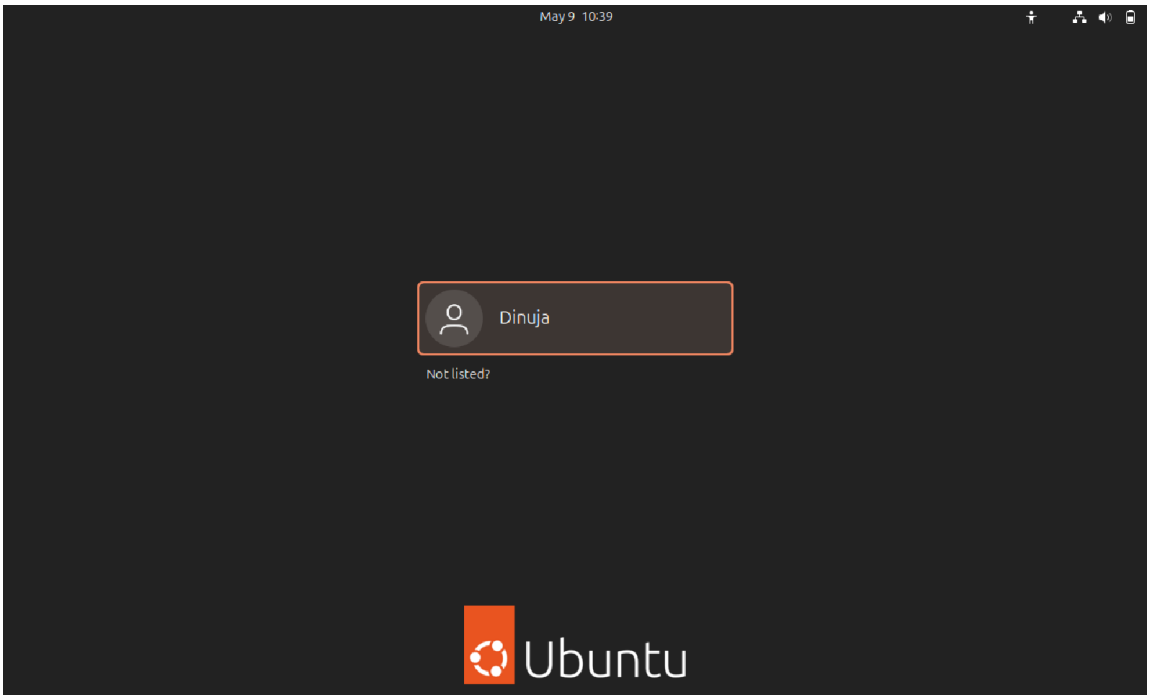


x. **Ubuntu will start the installation automatically**

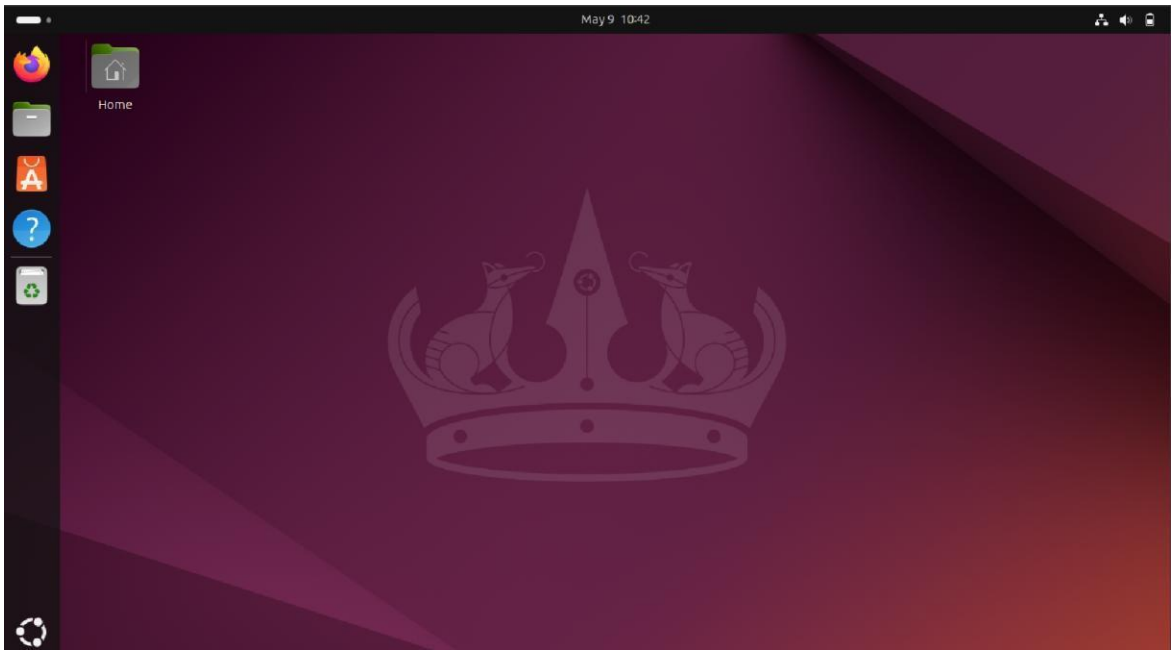


Note: I'll continue using my previous Ubuntu setup instead of the new installation mentioned above.

xi. After the installation is completed the VM will restart and will show the login page



xii. Enter your chosen username and password in step *vi* to log in.



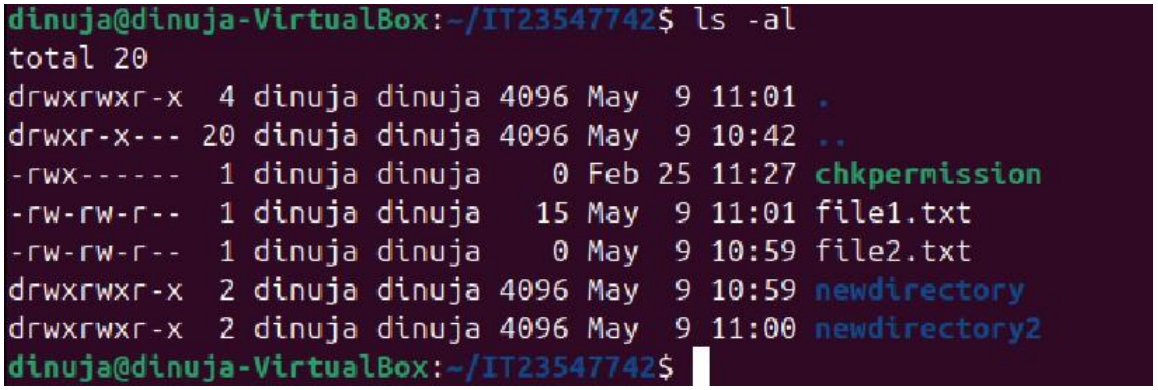
1.2 Command Line Introduction

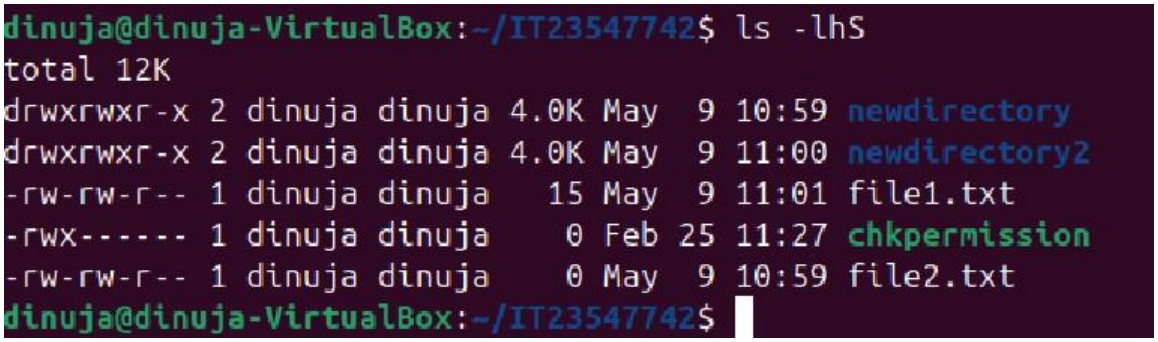
1.2.1 Navigation Commands

➢ `ls`

- `ls` is used to list files

- flags can be used to enhance the result
  - `-l`: Lists files in long format (permissions, owner, size, etc.).
  - `-a`: Includes all files, including hidden ones (those starting with `.`).
  - `-h`: Displays file sizes in human-readable format (e.g., KB, MB).
  - `-t`: Sorts files by modification time, newest first.
  - `-r`: Reverses the sorting order.
  - `-S`: Sorts files by size, largest first.
  - `-R`: Recursively lists all directories and their contents.





- cd
  - **cd** is used to change directories.

```
dinuja@dinuja-VirtualBox:~$ cd IT23547742/newdirectory
dinuja@dinuja-VirtualBox:~/IT23547742/newdirectory$
```

  - **cd ..** is used to go to the parent directory

```
dinuja@dinuja-VirtualBox:~/IT23547742/newdirectory$ cd ..
dinuja@dinuja-VirtualBox:~/IT23547742$
```

  - **cd** can be used to go to the root directory

```
dinuja@dinuja-VirtualBox:~/IT23547742/newdirectory$ cd
dinuja@dinuja-VirtualBox:~$
```

- **pwd**
  - **pwd** is used to print current working directory

```
dinuja@dinuja-VirtualBox:~/IT23547742/newdirectory$ pwd
/home/dinuja/IT23547742/newdirectory
```

1.2.2 File manipulation commands

- touch
  - **touch filename** creates an empty file or updates the timestamp if it exists.

```
dinuja@dinuja-VirtualBox:~/IT23547742/SNP$ touch file1
dinuja@dinuja-VirtualBox:~/IT23547742/SNP$ ls
file1
dinuja@dinuja-VirtualBox:~/IT23547742/SNP$
```

- cat
  - **cat >** is used to create files with an input.

```
dinuja@dinuja-VirtualBox:~/IT23547742$ cat > inputfile.c
Hello Everyone!!
My name is Dinuja.
My IT number is IT23547742
dinuja@dinuja-VirtualBox:~/IT23547742$ cat inputfile.c
Hello Everyone!!
My name is Dinuja.
My IT number is IT23547742
dinuja@dinuja-VirtualBox:~/IT23547742$
```

  - **cat >>** can be used to append to a file.

```
dinuja@dinuja-VirtualBox:~/IT23547742$ cat inputfile.c
Hello Everyone!!
My name is Dinuja.
My IT number is IT23547742
dinuja@dinuja-VirtualBox:~/IT23547742$ cat >> inputfile.c
This is an append to previous file
dinuja@dinuja-VirtualBox:~/IT23547742$ cat inputfile.c
Hello Everyone!!
My name is Dinuja.
My IT number is IT23547742
This is an append to previous file
dinuja@dinuja-VirtualBox:~/IT23547742$
```

- cp
  - **cp** is used to copy files from the current directory to another directory.

```
dinuja@dinuja-VirtualBox:~/IT23547742/SNP$ ls
target_destination  target_file
dinuja@dinuja-VirtualBox:~/IT23547742/SNP$ cp target_file ./target_destination
dinuja@dinuja-VirtualBox:~/IT23547742/SNP$ ls
target_destination  target_file
dinuja@dinuja-VirtualBox:~/IT23547742/SNP$ cd target_destination
dinuja@dinuja-VirtualBox:~/IT23547742/SNP/target_destination$ ls
target_file
dinuja@dinuja-VirtualBox:~/IT23547742/SNP/target_destination$
```

- **mv**
  - mv is used to move or rename files.
    - Moving a file

```
dinuja@dinuja-VirtualBox:~/IT23547742/SNP$ ls
target_destination  target_file
dinuja@dinuja-VirtualBox:~/IT23547742/SNP$ mv target_file ./target_destination
dinuja@dinuja-VirtualBox:~/IT23547742/SNP$ ls
target_destination
dinuja@dinuja-VirtualBox:~/IT23547742/SNP$ cd target_destination
dinuja@dinuja-VirtualBox:~/IT23547742/SNP/target_destination$ ls
target_file
dinuja@dinuja-VirtualBox:~/IT23547742/SNP/target_destination$
```

    - Renaming a file

```
dinuja@dinuja-VirtualBox:~/IT23547742/SNP$ ls
file_1  target_destination
dinuja@dinuja-VirtualBox:~/IT23547742/SNP$ mv file_1 Renamed_file
dinuja@dinuja-VirtualBox:~/IT23547742/SNP$ ls
Renamed_file  target_destination
dinuja@dinuja-VirtualBox:~/IT23547742/SNP$
```

- **`mkdir`**
  - `mkdir` command stands for make new directories.
  - Used to create new directories.

```
dinuja@dinuja-VirtualBox:~/IT23547742/SNP$ ls
target_destination
dinuja@dinuja-VirtualBox:~/IT23547742/SNP$ mkdir newdir
dinuja@dinuja-VirtualBox:~/IT23547742/SNP$ ls
newdir  target_destination
dinuja@dinuja-VirtualBox:~/IT23547742/SNP$
```

- **`rmdir`**
  - `rmdir` means remove directory.
  - Used to delete directories.

```
dinuja@dinuja-VirtualBox:~/IT23547742/SNP$ ls
newdir  target_destination
dinuja@dinuja-VirtualBox:~/IT23547742/SNP$ rmdir newdir
dinuja@dinuja-VirtualBox:~/IT23547742/SNP$ ls
target_destination
dinuja@dinuja-VirtualBox:~/IT23547742/SNP$
```

- **`rm`**
  - **`rm`** is used to remove files.

```
dinuja@dinuja-VirtualBox:~/IT23547742/SNP$ ls
file
dinuja@dinuja-VirtualBox:~/IT23547742/SNP$ rm file
dinuja@dinuja-VirtualBox:~/IT23547742/SNP$ ls
dinuja@dinuja-VirtualBox:~/IT23547742/SNP$
```

  - **`rm`** with flags **`-rf`** can also be used to delete a directory recursively (Directory and all the subdirectories and files in it).

```
dinuja@dinuja-VirtualBox:~/IT23547742/SNP$ ls
file  new_dir  target_destination  target_file
dinuja@dinuja-VirtualBox:~/IT23547742/SNP$ cd new_dir
dinuja@dinuja-VirtualBox:~/IT23547742/SNP/new_dir$ ls
file1  file3
dinuja@dinuja-VirtualBox:~/IT23547742/SNP/new_dir$ cd ..
dinuja@dinuja-VirtualBox:~/IT23547742/SNP$ rmdir new_dir
rmdir: failed to remove 'new_dir': Directory not empty
dinuja@dinuja-VirtualBox:~/IT23547742/SNP$ rm -rf ./new_dir
dinuja@dinuja-VirtualBox:~/IT23547742/SNP$ ls
file  target_destination  target_file
dinuja@dinuja-VirtualBox:~/IT23547742/SNP$
```

1.2.3 System information commands

- **`uname`**
  - Displays system and kernel information.

```
dinuja@dinuja-VirtualBox:~$ uname -a
Linux dinuja-VirtualBox 6.8.0-52-generic #53-Ubuntu SMP PREEMPT_DYNAMIC Sat Jan
11 00:06:25 UTC 2025 x86_64 x86_64 x86_64 GNU/Linux
dinuja@dinuja-VirtualBox:~$
```

- **`hostname`**
  - Displays hostname.

```
dinuja@dinuja-VirtualBox:~$ hostname
dinuja-VirtualBox
dinuja@dinuja-VirtualBox:~$
```

- **`lscpu`**
  - Displays CPU architecture information.

```
dinuja@dinuja-VirtualBox:~$ lscpu
Architecture:             x86_64
  CPU op-mode(s):         32-bit, 64-bit
  Address sizes:          39 bits physical, 48 bits virtual
  Byte Order:             Little Endian
CPU(s):                   3
  On-line CPU(s) list:    0-2
Vendor ID:                GenuineIntel
```

- **`free`**
  - Displays memory usage (RAM).

```
dinuja@dinuja-VirtualBox:~$ free -h
               total        used        free      shared  buff/cache   available
Mem:           2.9Gi       1.2Gi       141Mi        32Mi       1.8Gi       1.7Gi
Swap:          2.9Gi       1.0Mi       2.9Gi
```

- **df**
  - Displays the disk space usage.

```
dinuja@dinuja-VirtualBox:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
tmpfs           298M  1.6M  296M   1% /run
/dev/sda2        25G   12G   13G  49% /
tmpfs           1.5G     0  1.5G   0% /dev/shm
tmpfs           5.0M  8.0K  5.0M   1% /run/lock
tmpfs           298M  140K  298M   1% /run/user/1000
```

- **uptime**
  - Displays the system uptime.

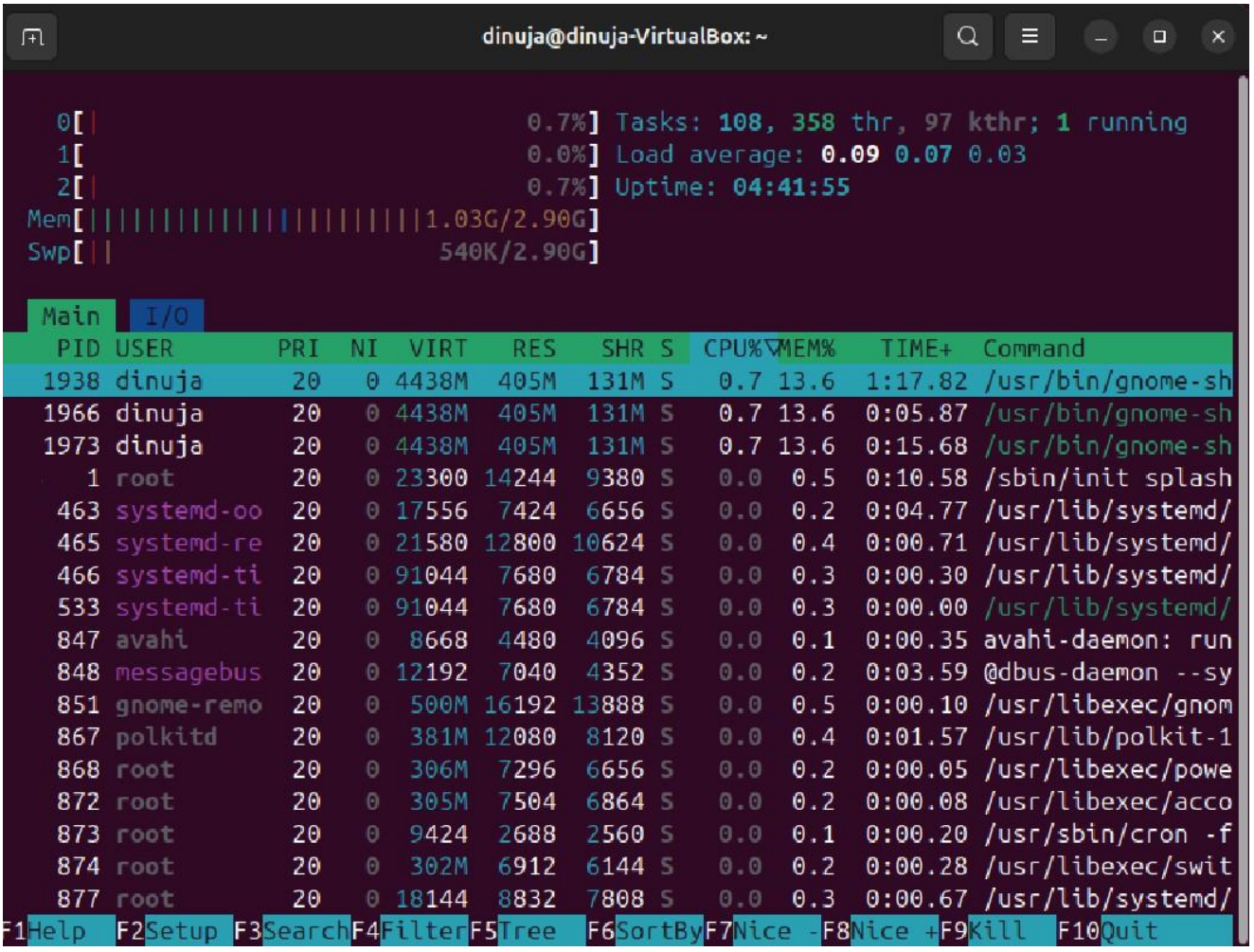
```
dinuja@dinuja-VirtualBox:~$ uptime
 15:08:00 up  4:29,  1 user,  load average: 0.00, 0.00, 0.00
```

- **top/htop**
  - Displays system monitoring information.
  - `top` is preinstalled and `htop` should be manually installed
  - `htop` is more user friendly and interactive than `top`
- **top**


```
top - 15:15:21 up  4:37,  1 user,  load average: 0.18, 0.14, 0.05
Tasks: 206 total,   2 running, 204 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.1 us,  0.1 sy,  0.0 ni, 99.8 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :   2972.5 total,    237.1 free,   1234.0 used,   1716.4 buff/cache
MiB Swap:   2972.0 total,   2970.7 free,      1.3 used.   1738.5 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
   1938 dinuja    20   0 4544728 415252 134676 S   0.6  13.6   2:26.70 gnome-shell
   2031 dinuja    20   0  388840  11936   6784 S   0.3   0.4   0:08.90 ibus-daemon
   4334 dinuja    20   0  626508  56700  44296 R   0.3   1.9   0:15.55 gnome-termi+
   7101 root      20   0       0      0      0 I   0.3   0.0   0:00.52 kworker/u8:+
      1 root      20   0   23300  14244   9380 S   0.0   0.5   0:10.53 systemd
      2 root      20   0       0      0      0 S   0.0   0.0   0:00.13 kthreadd
      3 root      20   0       0      0      0 S   0.0   0.0   0:00.00 pool_workqu+
      4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/R-r+
      5 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/R-r+
      6 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/R-s+
      7 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/R-n+
     11 root      20   0       0      0      0 I   0.0   0.0   0:00.00 kworker/u6:+
     12 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/R-m+
     13 root      20   0       0      0      0 I   0.0   0.0   0:00.00 rcu_tasks_k+
     14 root      20   0       0      0      0 I   0.0   0.0   0:00.00 rcu_tasks_r+
     15 root      20   0       0      0      0 I   0.0   0.0   0:00.00 rcu_tasks_t+
     16 root      20   0       0      0      0 S   0.0   0.0   0:00.95 ksoftirqd/0
     17 root      20   0       0      0      0 I   0.0   0.0   0:03.07 rcu_preempt
     18 root      rt   0       0      0      0 S   0.0   0.0   0:00.18 migration/0
     19 root     -51   0       0      0      0 S   0.0   0.0   0:00.00 idle_inject+
```

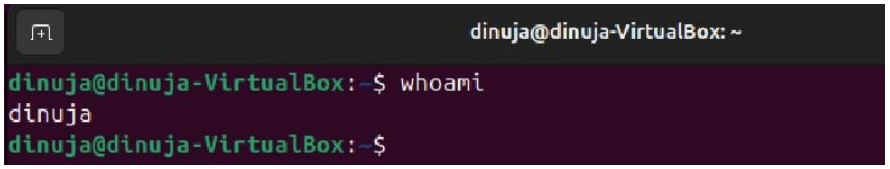
htop

```
dinuja@dinuja-VirtualBox:~$ htop
```

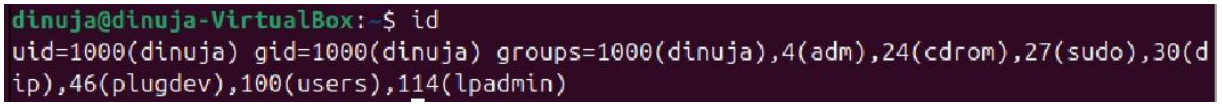


1.2.4 User management commands

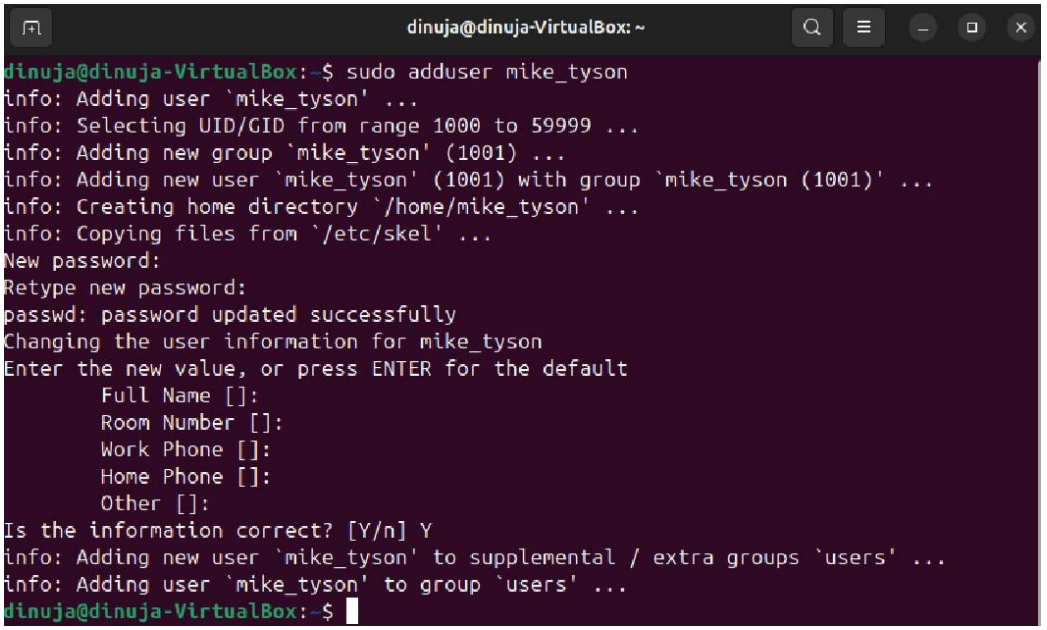
- **`whoami`**
  - Displays current user information.



- **`id`**
  - Displays user and group information.



- **`sudo adduser`**
  - Adds a new user.



- **`sudo deluser`**
  - Deletes a user.

```
dinuja@dinuja-VirtualBox:~$ sudo deluser mike_tyson
info: Removing crontab ...
info: Removing user `mike_tyson' ...
dinuja@dinuja-VirtualBox:~$
```

- **`passwd`**
  - Changes the password of a user.

```
dinuja@dinuja-VirtualBox:~$ passwd dinuja
Changing password for dinuja.
Current password:
New password:
BAD PASSWORD: The password is too similar to the old one
New password:
Retype new password:
passwd: password updated successfully
```

- **`groups`**
  - Lists user groups.

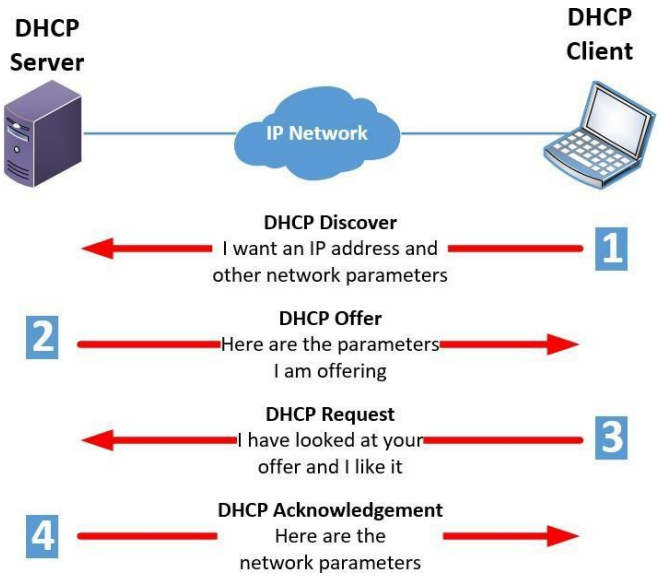
```
dinuja@dinuja-VirtualBox:~$ groups
dinuja adm cdrom sudo dip plugdev users lpadmin
dinuja@dinuja-VirtualBox:~$
```

- **`last`**
  - Displays login history.

```
dinuja@dinuja-VirtualBox:~$ last
dinuja   tty2         tty2             Fri May  9 10:41   still logged in
dinuja   seat0        login screen     Fri May  9 10:41   still logged in
reboot   system boot  6.8.0-52-generic Fri May  9 10:38   still running
reboot   system boot  6.8.0-52-generic Fri May  9 10:08   still running
dinuja   tty2         tty2             Mon Feb 24 23:12 - crash (73+10:55)
dinuja   seat0        login screen     Mon Feb 24 23:12 - crash (73+10:55)
reboot   system boot  6.8.0-52-generic Mon Feb 24 23:11   still running
dinuja   tty2         tty2             Sun Feb 23 20:40 - crash (1+02:30)
dinuja   seat0        login screen     Sun Feb 23 20:40 - crash (1+02:30)
reboot   system boot  6.8.0-52-generic Sun Feb 23 20:39   still running
dinuja   tty2         tty2             Sun Feb 23 20:33 - crash  (00:05)
dinuja   seat0        login screen     Sun Feb 23 20:33 - crash  (00:05)
reboot   system boot  6.8.0-52-generic Sun Feb 23 20:32   still running
dinuja   tty2         tty2             Mon Feb 17 13:47 - crash (6+06:44)
dinuja   seat0        login screen     Mon Feb 17 13:47 - crash (6+06:44)
reboot   system boot  6.8.0-51-generic Mon Feb 17 13:46   still running
dinuja   tty2         tty2             Thu Feb 13 10:17 - crash (4+03:28)
dinuja   seat0        login screen     Thu Feb 13 10:17 - crash (4+03:28)
reboot   system boot  6.8.0-51-generic Thu Feb 13 10:16   still running
dinuja   tty2         tty2             Thu Feb 13 09:56 - crash  (00:20)
dinuja   seat0        login screen     Thu Feb 13 09:56 - crash  (00:20)
reboot   system boot  6.8.0-51-generic Thu Feb 13 09:54   still running
dinuja   tty2         tty2             Tue Jan  7 23:45 - crash (36+10:09)
dinuja   seat0        login screen     Tue Jan  7 23:45 - crash (36+10:09)
reboot   system boot  6.8.0-51-generic Tue Jan  7 23:42   still running

wtmp begins Tue Jan  7 23:42:57 2025
dinuja@dinuja-VirtualBox:~$
```

2. DHCP, DNS and NTP services

2.1 DHCP (Dynamic Host Configuration Protocol)



2.1.1 What is DHCP

- **DHCP** is a network management protocol that automates the assignment of IP addresses, subnet masks, default gateways, and DNS server information.
- It helps eliminate IP conflicts by dynamically leasing IP addresses to devices on the network.
- This ensures that each connected device receives a unique IP address and prevents manual configuration errors.
- DHCP also optimizes IP usage by reassigning addresses once leases expire and devices are no longer connected.
- In this implementation, the DHCP server I configured manages a limited address pool (Scope: **192.168.100.20–30**), dynamically allocating IP addresses to devices within the same subnet.

2.1.2 Configuring DHCP server

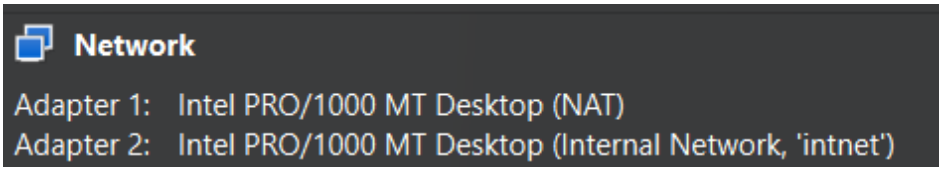
i. Update the package list

```
dinuja@dinuja-VirtualBox:~$ sudo apt update
[sudo] password for dinuja:
Hit:1 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:2 http://lk.archive.ubuntu.com/ubuntu noble InRelease
Get:3 http://lk.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Hit:4 http://lk.archive.ubuntu.com/ubuntu noble-backports InRelease
Fetched 126 kB in 2s (78.7 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
327 packages can be upgraded. Run 'apt list --upgradable' to see them.
dinuja@dinuja-VirtualBox:~$
```

ii. Identify network settings

```
dinuja@dinuja-VirtualBox:~$ ip -br a
lo               UNKNOWN        127.0.0.1/8 ::1/128
enp0s3           UP             10.0.2.15/24 fd17:625c:f037:2:5664:1e5
4:6460:a544/64 fd17:625c:f037:2:a00:27ff:fe99:ade2/64 fe80::a00:27ff:f
e99:ade2/64
enp0s8           UP
```

- Before configuring the DHCP server I used `ip -br a` command to identify the network settings.
- You can identify your IP address, Subnet mask, and network ID from this step.
- Here you can see I am using dual network adaptelsrm -rs `enps03` and `enpso8`.
- This occurred while attempting to simulate a network environment using another virtual machine as a client, during which I encountered an issue.
- The issue was that I was unable to establish an internet connection using any network adapter configuration other than the NAT adapter.
- So I used NAT for internet and internal network adaptor for DHCP simulation.



Key Information:

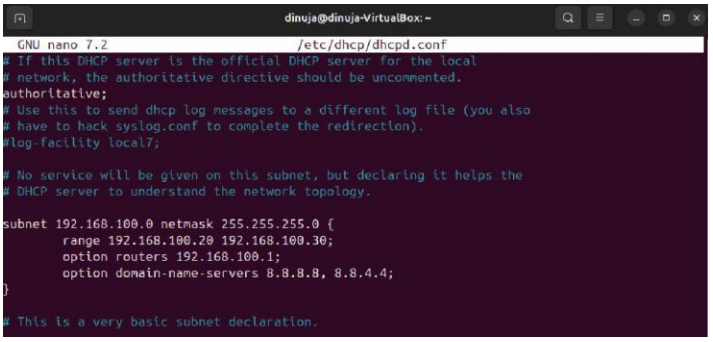
- **`enp0s3` (NAT)**:
  - IP: 10.0.2.15
  - Subnet Mask: /24 (Class C)
  - Network ID: 10.0.2.0
- **`enp0s8` (Internal Network)**:
  - No IPv4 assigned (only IPv6 link-local)
  - We will need to assign an IP address to this for our simulation to work.

iii. Install the DHCP server (isc-dhcp-server)

```
dinuja@dinuja-VirtualBox:~$ sudo apt install isc-dhcp-server -y
[sudo] password for dinuja:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

iv. Configure the DHCP server

```
dinuja@dinuja-VirtualBox:~$ sudo cp /etc/dhcp/dhcpd.conf /etc/dhcp/dhcpd.conf.bak
[sudo] password for dinuja:
dinuja@dinuja-VirtualBox:~$ sudo nano /etc/dhcp/dhcpd.conf
dinuja@dinuja-VirtualBox:~$
```



- I have copied the configuration file and created a backup file prior to configuration because of the possibility of having a misconfiguration we can use the original file from backup.
- Here I used the IP range `192.168.100.20 - 192.168.100.30` to avoid conflicts with Virtualbox's default DHCP configurations.
- Because the recommended range for local networks by **RFC 1918** is `192.168.0.0/16` block.

v. Specify the network interface

```
dinuja@dinuja-VirtualBox:~$ sudo nano /etc/default/isc-dhcp-server
```

```
# On what interfaces should the DHCP server (dhcpd) serve DHCP requests?
#       Separate multiple interfaces with spaces, e.g. "eth0 eth1".
INTERFACESv4="enp0s8"
INTERFACESv6=""
```

- Here I used `enp0s8` instead of `enp0s3` because we are using only `enp0s8` for DHCP simulation.

vi. Assign a permanent static IP address to `enp0s8`.

```
dinuja@dinuja-VirtualBox:~$ sudo nano /etc/netplan/00-installer-config.yaml
[sudo] password for dinuja:
```

```
network:
  version: 2
  ethernets:
    enp0s3:  # NAT adapter
      dhcp4: true
      optional: true
    enp0s8:  # Internal network
      addresses: [192.168.100.1/24]
      dhcp4: no
      optional: true
```

- Here the enps03 is unchanged which is used for external connectivity (internet)
- Enps08 is used for DHCP emulation.

```
dinuja@dinuja-VirtualBox:~$ sudo netplan apply
```

- Now apply the changes.

```
dinuja@dinuja-VirtualBox:~$ ip -br a
lo               UNKNOWN        127.0.0.1/8 ::1/128
enp0s3           UP             10.0.2.15/24 fd17:625c:f037:2:305f:85da:def2:7323/64 fd17:625
c:f037:2:a00:27ff:fe99:ade2/64 fe80::a00:27ff:fe99:ade2/64
enp0s8           UP             192.168.100.1/24 fe80::a00:27ff:fedc:9080/64
dinuja@dinuja-VirtualBox:~$
```

- If you ran `ip -br a` again you can see now that `enps08` adapter also has an IPv4 address which we assigned manually.
- Because we defined that in `/etc/netplan/00-installer-config.yaml` file it will not get reset when the system is rebooted.

vii. Restart the server and check the status.

```
dinuja@dinuja-VirtualBox:~$ sudo systemctl restart isc-dhcp-server
dinuja@dinuja-VirtualBox:~$ sudo systemctl status isc-dhcp-server
● isc-dhcp-server.service - ISC DHCP IPv4 server
     Loaded: loaded (/usr/lib/systemd/system/isc-dhcp-server.service; enabled; preset: enabl>
     Active: active (running) since Fri 2025-05-09 22:05:35 +0530; 19s ago
```

2.1.3 *Setting up the client machine*

i. Checking network settings of the client machine

```
┌──(dinuja㉿kali)-[~]
└─$ ip -br a
lo               UNKNOWN        127.0.0.1/8 ::1/128
eth0             UP
eth1             UP             10.0.2.15/24 fd17:625c:f037:2:649a:f2b0:1a20:e188/64 fd1
7:625c:f037:2:a00:27ff:fe0c:ad0/64 fe80::a00:27ff:fe0c:ad0/64
```

- Two network adapters are used here are same just like in the server machine.
- I am using `eth1` for connectivity and `eth0` for DHCP simulation.
- The above output shows that for `eth0` there is no IP address because unlike NAT adapter, Internal Network adapter does not assign an IP by default. (Same as in the server machine).
- we can use the DHCP server we created in the Ubuntu machine to acquire an IP address to `eth0`.

ii. Acquiring a DHCP lease

```
┌──(dinuja㉿kali)-[~]
└─$ sudo dhclient -r eth0 && sudo dhclient -v eth0
[sudo] password for dinuja:
Internet Systems Consortium DHCP Client 4.4.3-P1
Copyright 2004-2022 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

Listening on LPF/eth0/08:00:27:12:65:df
Sending on   LPF/eth0/08:00:27:12:65:df
Sending on   Socket/fallback
DHCPDISCOVER on eth0 to 255.255.255.255 port 67 interval 7
DHCPOFFER of 192.168.100.21 from 192.168.100.1
DHCPREQUEST for 192.168.100.21 on eth0 to 255.255.255.255 port 67
DHCPACK of 192.168.100.21 from 192.168.100.1
bound to 192.168.100.21 -- renewal in 228 seconds.

┌──(dinuja㉿kali)-[~]
└─$
```

- The output shows that the client machine IP address is `192.168.100.21` (Which is the second IP address of the range we specified in the server machine)
- We can confirm this by running `ip -br a` command.

```
┌──(dinuja㉿kali)-[~]
└─$ ip -br a
lo               UNKNOWN        127.0.0.1/8 ::1/128
eth0             UP             192.168.100.21/24
eth1             UP             10.0.2.15/24 fd17:625c:f037:2:649a:f2b0:1a20:e188/64 fd1
7:625c:f037:2:a00:27ff:fe0c:ad0/64 fe80::a00:27ff:fe0c:ad0/64
```

- You can configure networking on Kali Linux using the traditional method (/etc/network/interfaces), or using Netplan.

*Note: For this I tried to install netplan but ended up with some errors. So I manually configure networking on **Kali Linux** using the traditional method (/etc/network/interfaces), which **does not require additional resources like Netplan**.*

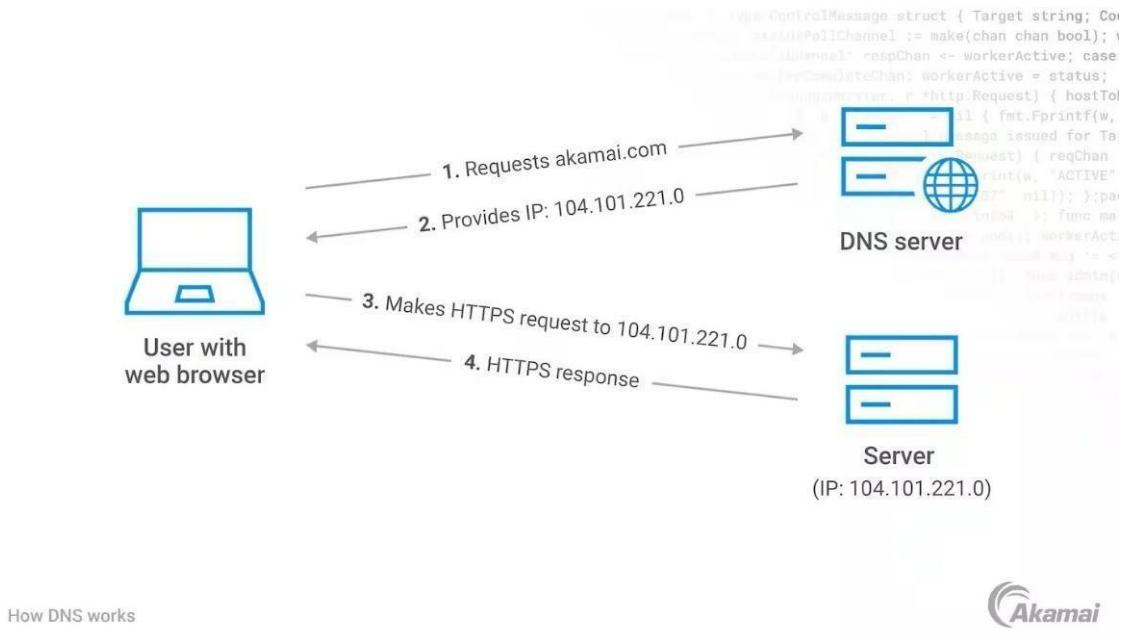
```
┌──(dinuja㉿kali)-[~]
└─$ sudo nano /etc/network/interfaces
[sudo] password for dinuja:
Sorry, try again.
[sudo] password for dinuja:

┌──(dinuja㉿kali)-[~]
└─$
```

```
dinuja@kali: ~
File  Actions  Edit  View  Help
  GNU nano 8.2                    /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# eth0 - Internal network (DHCP)
auto eth0
iface eth0 inet dhcp

# eth1 - NAT (VirtualBox default, DHCP)
auto eth1
iface eth1 inet dhcp
```

- You can also check DHCP leases on the server machine to see the lease details.

```
dinuja@dinuja-VirtualBox:~$ cat /var/lib/dhcp/dhcpd.leases
# The format of this file is documented in the dhcpd.leases(5) manual page.
# This lease file was written by isc-dhcp-4.4.3-P1

# authoring-byte-order entry is generated, DO NOT DELETE
authoring-byte-order little-endian;

lease 192.168.100.20 {
  starts 5 2025/05/09 18:45:22;
  ends 5 2025/05/09 18:55:22;
  tstp 5 2025/05/09 18:55:22;
  cltt 5 2025/05/09 18:45:22;
  binding state free;
  hardware ethernet 08:00:27:12:65:df;
  uid "\001\010\000'\022e\337";
}
lease 192.168.100.21 {
  starts 5 2025/05/09 20:59:50;
  ends 5 2025/05/09 21:09:50;
  cltt 5 2025/05/09 20:59:50;
  binding state active;
  next binding state free;
  rewind binding state free;
  hardware ethernet 08:00:27:12:65:df;
  uid "\377'\022e\337\000\001\000\001/\261%\342\010\000'\022e\337";
  client-hostname "kali";
}
```

2.2 DNS (Domain Name System)



2.2.1 What is DNS

- DNS (Domain Name System) is a hierarchical naming system that translates easy-to-remember domain names like **lab.local** into IP addresses like **192.168.100.20**, which computers use to communicate with each other.
- It saves us from having to remember complex numerical IP addresses every time we want to access a website or service.
- DNS also improves network performance by temporarily storing (caching) recent domain name resolutions, making future lookups faster.
- For my implementation, I set up a **BIND9 DNS server** with the following features:
    - It authoritatively resolves local domain names like **www.lab.local**, mapping them to IP addresses such as **192.168.100.20**.
    - It forwards any external DNS queries (outside the local network) to Google's DNS server at **8.8.8.8**.
    - I also reconfigured the previously created DHCP server to work seamlessly with this DNS setup, ensuring proper name resolution across the network.

2.2.2 *Configuring DNS (Local) using BIND9*

i. Installing BIND9

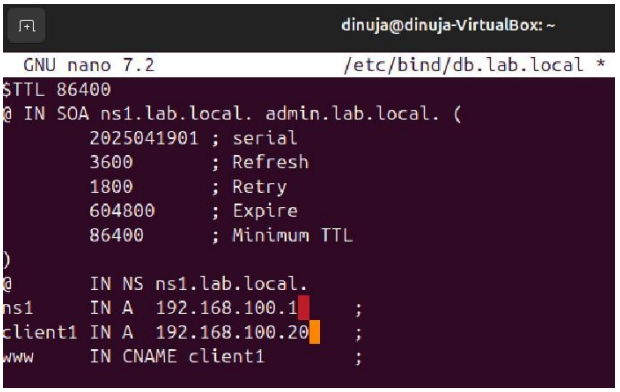
```
dinuja@dinuja-VirtualBox:~$ sudo apt install bind9 dnsutils -y
```

ii. Configuring BIND9

```
dinuja@dinuja-VirtualBox:~$ sudo nano /etc/bind/named.conf.options
```

iii. Configuring zone file

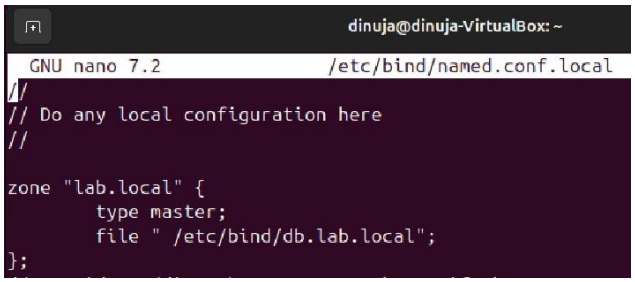
```
dinuja@dinuja-VirtualBox:~$ sudo nano /etc/bind/db.lab.local
```



- This configuration takes into account my previously created DHCP server too.

iv. Linking the zone file

```
dinuja@dinuja-VirtualBox:~$ sudo nano /etc/bind/named.conf.local
```



v. Validating BIND9

```
dinuja@dinuja-VirtualBox:~$ sudo named-checkzone lab.local /etc/bind/db.lab.local
zone lab.local/IN: loaded serial 2025041901
OK
```

vi. Restarting and Checking Status

```
dinuja@dinuja-VirtualBox:~$ sudo systemctl restart bind9
dinuja@dinuja-VirtualBox:~$ sudo systemctl status bind9
● named.service - BIND Domain Name Server
     Loaded: loaded (/usr/lib/systemd/system/named.service; enabled; preset: en>
     Active: active (running) since Sat 2025-05-10 12:33:54 +0530; 1min 3s ago
       Docs: man:named(8)
   Main PID: 4106 (named)
     Status: "running"
```

- I also reconfigured my DHCP server to use the DNS server we just created.

```
dinuja@dinuja-VirtualBox:~$ sudo nano /etc/dhcp/dhcpd.conf
```

```
subnet 192.168.100.0 netmask 255.255.255.0 {
        range 192.168.100.20 192.168.100.30;
        option routers 192.168.100.1;
        option domain-name "lab.local';
        option domain-name-servers 192.168.100.1;
}
```

- Since I added Google's Public DNS (8.8.8.8) as a forwarder when setting up BIND9, external domain queries will still work just like before. If my local DNS server can't resolve something or runs into issues, the system will automatically fall back to Google DNS to keep things running smoothly.

vii. Verifying DNS resolution on the server

```
dinuja@dinuja-VirtualBox:~$ dig client1.lab.local @192.168.100.1 +short
192.168.100.20
dinuja@dinuja-VirtualBox:~$ dig www.lab.local @192.168.100.1 +short
client1.lab.local.
192.168.100.20
```

viii. Verifying DNS resolution on the clientcat

```
┌──(dinuja㉿kali)-[~]
└─$ ip -br a show eth0
eth0             UP             192.168.100.21/24 fe80::a00:27ff:fe12:65df/64
```

```
┌──(dinuja㉿kali)-[~]
└─$ cat /etc/resolv.conf
domain lab.local
search lab.local
nameserver 192.168.100.1
```

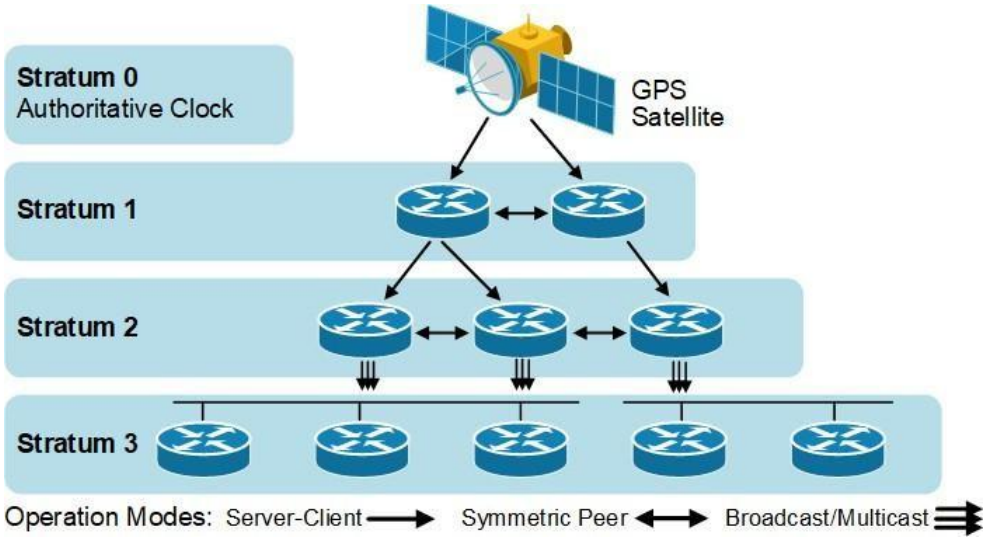
- The output confirms that the DNS is correctly resolved in the client machine.

```
┌──(dinuja㉿kali)-[~]
└─$ nslookup www.lab.local
Server:         192.168.100.1
Address:        192.168.100.1#53

www.lab.local   canonical name = client1.lab.local.
Name:   client1.lab.local
Address: 192.168.100.20
```

```
┌──(dinuja㉿kali)-[~]
└─$ dig client1.lab.local @192.168.100.1

; <<>> DiG 9.20.2-1-Debian <<>> client1.lab.local @192.168.100.1
;; global options: +cmd
;; Got answer:
;; WARNING: .local is reserved for Multicast DNS
;; You are currently testing what happens when an mDNS query is leaked to DN
S
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 29319
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 7d7e8e741e331d4601000000681f79b0dfaf1c5fb2dd0c2d (good)
;; QUESTION SECTION:
;client1.lab.local.             IN      A

;; ANSWER SECTION:
client1.lab.local.      86400   IN      A       192.168.100.20

;; Query time: 15 msec
;; SERVER: 192.168.100.1#53(192.168.100.1) (UDP)
;; WHEN: Sat May 10 12:07:06 EDT 2025
;; MSG SIZE  rcvd: 90
```

```
┌──(dinuja㉿kali)-[~]
└─$ dig www.lab.local @192.168.100.1

; <<>> DiG 9.20.2-1-Debian <<>> www.lab.local @192.168.100.1
;; global options: +cmd
;; Got answer:
;; WARNING: .local is reserved for Multicast DNS
;; You are currently testing what happens when an mDNS query is leaked to DN
S
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 7075
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: d3767e79d0194a5e01000000681f7a3196dcd9bebcd92dfd (good)
;; QUESTION SECTION:
;www.lab.local.                 IN      A

;; ANSWER SECTION:
www.lab.local.          86400   IN      CNAME   client1.lab.local.
client1.lab.local.      86400   IN      A       192.168.100.20

;; Query time: 31 msec
;; SERVER: 192.168.100.1#53(192.168.100.1) (UDP)
;; WHEN: Sat May 10 12:09:15 EDT 2025
;; MSG SIZE  rcvd: 108
```

2.3 NTP (Network Time Protocol)



2.3.1 What is NTP

- NTP (Network Time Protocol) is used to keep the clocks of different computer systems in sync.

- It helps make sure that all systems, no matter where they are in the world or what time zone they're in, have the correct and consistent time.

- Having synchronized time is important for many network services, especially for logging, security, and smooth communication between devices.

    - In my setup,
    - The NTPsec service was configured to,
    - Synchronize with Google's public NTP servers (*time1.google.com*)
    - Use pool.ntp.org servers as fallback.

2.3.2 Configuring NTP server

i. Installing NTP

```
dinuja@dinuja-VirtualBox:~$ sudo apt update
[sudo] password for dinuja:
Hit:1 http://security.ubuntu.com/ubuntu noble-security InReleas
Hit:2 http://lk.archive.ubuntu.com/ubuntu noble InRelease
Get:3 http://lk.archive.ubuntu.com/ubuntu noble-updates InRelea
[126 kB]
```

```
dinuja@dinuja-VirtualBox:~$ sudo apt install ntp -y
```

ii. Configuring NTP

- First I ran `sudo nano /etc/ntp.conf command`, but it only opened an empty file.
- So I tried again by finding the correct file by running

  `sudo find /etc -name "ntp.conf*".`
- Then I found out the correct file was `/etc/ntpsec/ntp.conf` file.

```
# Google's NTP servers
server time1.google.com iburst
server time2.google.com iburst
server time3.google.com iburst
server time4.google.com iburst

# Fallback servers
server 0.pool.ntp.org iburst
server 1.pool.ntp.org iburst
```

- I used Google's Public servers for Primary servers and pool.ntp.org project's public NTP servers for fallback.

iii. Restarting and Checking status

```
dinuja@dinuja-VirtualBox:~$ sudo systemctl status ntp
[sudo] password for dinuja:
● ntpsec.service - Network Time Service
     Loaded: loaded (/usr/lib/systemd/system/ntpsec.service; enabled; preset: enabled)
     Active: active (running) since Sun 2025-05-11 09:30:54 +0530; 3min 8s ago
       Docs: man:ntpd(8)
    Process: 10043 ExecStart=/usr/libexec/ntpsec/ntp-systemd-wrapper (code=exited, status=0/SUCCESS)
```

iv. Verifying synchronization

```
dinuja@dinuja-VirtualBox:~$ ntpq -p
     remote
==========================================
+time1.google.com
+time2.google.com
+time3.google.com
*time4.google.com
+time.cloudflare.com
-ntp.sltidc.lk
```

- The output confirms that NTP service is working as intended.

```
dinuja@dinuja-VirtualBox:~$ timedatectl
               Local time: Sun 2025-05-11 11:56:19 +0530
           Universal time: Sun 2025-05-11 06:26:19 UTC
                 RTC time: Sun 2025-05-11 06:26:17
                Time zone: Asia/Colombo (+0530, +0530)
System clock synchronized: yes
              NTP service: active
          RTC in local TZ: no
```

3. Security and Other servers

3.1 Shell Scripting

3.1.1 Shell script to automate log cleanup and archival

i. Create a directory for backups

```
dinuja@dinuja-VirtualBox:~/SNP3/snp$ mkdir log_backups
dinuja@dinuja-VirtualBox:~/SNP3/snp$ ls
log_backups
```

ii. Create the shell script

```
dinuja@dinuja-VirtualBox:~/SNP3/snp$ code log_cleanup.sh
```

- I am using VS Code instead of nano because it is easier to use.
- To use VS code I have to install it into ubuntu because it don't come as default.
- I used ***sudo snap install code –classic*** command to install VS code into ubuntu
- Code is on the next page.
- I used *"codesnap"* extension to take the screenshot.

```bash
#! /bin/bash

#Log Cleanup and Archival Script
# Location: /home/dinuja/snp3/SNP/log_cleanup.sh

LOG_DIR="/var/log/"
BACKUP_DIR="/home/dinuja/SNP3/snp/log_backups"
BACKUP_FILE="logs_backup_$(date +'%Y-%m-%d').tar.gz"
DAYS_TO_KEEP=7

#counters
DELETED_COUNT=0
ARCHIVED_COUNT=0

#Create the directories if they do not exist
mkdir -p "$LOG_DIR"
mkdir -p "$BACKUP_DIR"

echo "[$(date)] Starting log cleanup and archival process..."

#Deleting logs older than 7d
echo "Finding and deleting logs older than $DAYS_TO_KEEP days..."

#creating a variable to store the result of the find
DELETED_FILES=$(find "$LOG_DIR" -name "*.log" -type f -mtime +$DAYS_TO_KEEP)

if [ -n "$DELETED_FILES" ]; then
    echo "The following files will be deleted:"
    echo "$DELETED_FILES"
    DELETED_COUNT=$(echo "$DELETED_FILES" | wc -l) #counts the lines using wc -l
    find "$LOG_DIR" -name "*.log" -type f -mtime +$DAYS_TO_KEEP -delete #Deleting files older than 7d

else
    echo "Could't find files older than $DAYS_TO_KEEP."
fi

#archiving the remaining files

ARCHIVED_FILES=$(find "$LOG_DIR" -name "*.log" -type f) #find all the files that didnt got deleted

if [ -n "$ARCHIVED_FILES" ]; then
    echo "The following files will be archived:"
    echo "$ARCHIVED_FILES"
    ARCHIVED_COUNT=$(echo "$ARCHIVED_FILES" | wc -l)
    tar -czf "$BACKUP_DIR/$BACKUP_FILE" -c "$LOG_DIR" --ignore-failed-read $(find "$LOG_DIR" -name "*.log" -type f -printf "%f\n")

    find "$LOG_DIR" -name "*.log" -type f -delete; #deleted files after archiving

else
    echo "Couldn't find files to archive."
fi

#printing summary
echo "Cleanup and archival completed:"
echo "Deleted $DELETED_COUNT log file(s)"
echo "Archived $ARCHIVED_COUNT log file(s) to $BACKUP_DIR/$BACKUP_FILE"

exit 0

```

```
#! /bin/bash

#Log Cleanup and Archival Script
# Location: /home/dinuja/snp3/SNP/log_cleanup.sh

LOG_DIR="/var/log/"
BACKUP_DIR="/home/dinuja/SNP3/snp/log_backups"
BACKUP_FILE="logs_backup_$(date +'%Y-%m-%d').tar.gz"
DAYS_TO_KEEP=7

#counters
DELETED_COUNT=0
ARCHIVED_COUNT=0

#Create the directories if they do not exist
mkdir -p "$LOG_DIR"
mkdir -p "$BACKUP_DIR"

echo "[$(date)] Starting log cleanup and archival process..."

#Deleting logs older than 7d
echo "Finding and deleting logs older than $DAYS_TO_KEEP days..."

#creating a variable to store the result of the find
DELETED_FILES=$(find "$LOG_DIR" -name "*.log" -type f -mtime +$DAYS_TO_KEEP)

if [ -n "$DELETED_FILES" ]; then
    echo "The following files will be deleted:"
    echo "$DELETED_FILES"
    DELETED_COUNT=$(echo "$DELETED_FILES" | wc -l) #counts the lines using wc -l
    find "$LOG_DIR" -name "*.log" -type f -mtime +$DAYS_TO_KEEP -delete #Deleting files older than 7d

else
    echo "Could't find files older than $DAYS_TO_KEEP."
fi

#archiving the remaining files

ARCHIVED_FILES=$(find "$LOG_DIR" -name "*.log" -type f) #find all the files that didnt got deleted

if [ -n "$ARCHIVED_FILES" ]; then
    echo "The following files will be archived:"
    echo "$ARCHIVED_FILES"
    ARCHIVED_COUNT=$(echo "$ARCHIVED_FILES" | wc -l)
    tar -czf "$BACKUP_DIR/$BACKUP_FILE" -c "$LOG_DIR" --ignore-failed-read $(find "$LOG_DIR" -name "*.log" -type f
-printf "%f\n")

    find "$LOG_DIR" -name "*.log" -type f -delete; #deleted files after archiving

else
    echo "Couldn't find files to archive."
fi

#printing summary
echo "Cleanup and archival completed:"
echo "Deleted $DELETED_COUNT log file(s)"
echo "Archived $ARCHIVED_COUNT log file(s) to $BACKUP_DIR/$BACKUP_FILE"

exit 0
```

iii. Make the script executable

```
dinuja@dinuja-VirtualBox:~/SNP3/snp$ ls
log_backups  log_cleanup.sh
dinuja@dinuja-VirtualBox:~/SNP3/snp$ chmod +x log_cleanup.sh
dinuja@dinuja-VirtualBox:~/SNP3/snp$ ls
log_backups  log_cleanup.sh
```

iv. Testing the script

```
dinuja@dinuja-VirtualBox:~/SNP3/snp$ sudo ./log_cleanup.sh
[sudo] password for dinuja:
[Sun May 11 05:43:27 PM +0530 2025] Starting log cleanup and archival process...
Finding and deleting logs older than 7 days...
Could't find files older than 7.
The following files will be archived:
/var/log/apport.log
tar: Removing leading `/' from member names
tar: Removing leading `/' from hard link targets
tar: apport.log: Warning: Cannot stat: No such file or directory
Cleanup and archival completed:
Deleted 0 log file(s)
Archived 1 log file(s) to /home/dinuja/SNP3/snp/log_backups/logs_backup_2025-05-11.tar.gz
```

- The output shows that the script works as intended and we can find the compressed file in the appropriate location.

```
dinuja@dinuja-VirtualBox:~/SNP3/snp$ cd log_backups
dinuja@dinuja-VirtualBox:~/SNP3/snp/log_backups$ ls
logs_backup_2025-05-11.tar.gz
dinuja@dinuja-VirtualBox:~/SNP3/snp/log_backups$
```

3.1.2 Scheduling the script to run automatically

i. Editing crontab

```
dinuja@dinuja-VirtualBox:~/SNP3/snp$ crontab -e
no crontab for dinuja - using an empty one

Select an editor.  To change later, run 'select-editor'.
  1. /bin/nano        <---- easiest
  2. /usr/bin/vim.tiny
  3. /bin/ed

Choose 1-3 [1]: 1
crontab: installing new crontab
```

```
0 0 * * 0 /home/dinuja/SNP3/snp/log_cleanup.sh
```

- The script is scheduled to execute at **00:00 (midnight) every Sunday**, as specified by the cron timing format: minute 0, hour 0, and day of the week 0.
- This configuration ensures that the script runs once a week at the beginning of each Sunday.
- As per the assignment instructions, it was assumed that the necessary permissions were already in place; therefore, the script was not executed with root privileges during testing.

3.2 SSH (Secure Shell)

i. Installing SSH

```
dinuja@dinuja-VirtualBox:~$ sudo apt install openssh-server
[sudo] password for dinuja:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

ii. Configuring SSH

```
dinuja@dinuja-VirtualBox:~$ code /etc/ssh/sshd_config
```

```
dinuja@dinuja-VirtualBox:~$ grep -v '^#' /etc/ssh/sshd_config | grep -v '^$'
Include /etc/ssh/sshd_config.d/*.conf
Port 2222
AddressFamily inet
ListenAddress 192.168.100.1
PermitRootLogin no
PubkeyAuthentication yes
PasswordAuthentication no
KbdInteractiveAuthentication no
UsePAM yes
X11Forwarding no
PrintMotd no
ClientAliveInterval 300
ClientAliveCountMax 2
AcceptEnv LANG LC_*
Subsystem       sftp    /usr/lib/openssh/sftp-server
dinuja@dinuja-VirtualBox:~$
```

- These are the settings I changed in the `/etc/ssh/sshd_config` file
- Some of them are default settings while I changed
    - `Port` to 2222
    - `AddressFamily inet` to force IPv4 only.
    - `ListenAddress` to the DHCP server IP (So that the devices outside of that network cannot connect even if they knew the credentials)
    - `ClientAliveInterval` to 300 so the server checks for Idle devices every 5 minutes.

iii. validating SSH

```
dinuja@dinuja-VirtualBox:~$ sudo sshd -t
dinuja@dinuja-VirtualBox:~$
```

iv. Checking status

```
dinuja@dinuja-VirtualBox:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/usr/lib/systemd/system/ssh.service; enabled; preset: enabled)
     Active: active (running) since Sun 2025-05-11 18:39:54 +0530; 1s ago
TriggeredBy: ● ssh.socket
       Docs: man:sshd(8)
             man:sshd_config(5)
    Process: 37953 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Main PID: 37956 (sshd)
      Tasks: 1 (limit: 3477)
     Memory: 1.5M (peak: 1.8M)
        CPU: 33ms
     CGroup: /system.slice/ssh.service
             └─37956 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

May 11 18:39:54 dinuja-VirtualBox systemd[1]: ssh.service: Deactivated successfully.
May 11 18:39:54 dinuja-VirtualBox systemd[1]: Stopped ssh.service - OpenBSD Secure Shell server.
May 11 18:39:54 dinuja-VirtualBox systemd[1]: Starting ssh.service - OpenBSD Secure Shell server...
May 11 18:39:54 dinuja-VirtualBox sshd[37956]: Server listening on 192.168.100.1 port 2222.
May 11 18:39:54 dinuja-VirtualBox systemd[1]: Started ssh.service - OpenBSD Secure Shell server.
```

v. Connecting through a client

```
┌──(dinuja㉿kali)-[~]
└─$ ssh -p 2222 dinuja@192.168.100.1
The authenticity of host '[192.168.100.1]:2222 ([192.168.100.1]:2222)' can't be established.
ED25519 key fingerprint is SHA256:nuyZc2Ty6jp6RAJP9o122dAyh9bWlWjWWfDWORuCldc.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])?
```

- First connection attempt. In the second connection attempt I typed yes.

```
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[192.168.100.1]:2222' (ED25519) to the list of known hosts.
Connection closed by 192.168.100.1 port 2222
```

vi. Authentication security

```
┌──(dinuja㉿kali)-[~]
└─$ ssh -p 2222 dinuja@192.168.100.1
dinuja@192.168.100.1: Permission denied (publickey).
```

- This demonstrates that the server is properly configured for key based authentication only.

vii. Key generation

```
┌──(dinuja㉿kali)-[~]
└─$ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/dinuja/.ssh/id_ed25519):
Enter passphrase for "/home/dinuja/.ssh/id_ed25519" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/dinuja/.ssh/id_ed25519
Your public key has been saved in /home/dinuja/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:dyghD9T1Vs1NFR00Gh9PuzFsYqs3ep7gwhAbywg8/GA dinuja@kali
The key's randomart image is:
+--[ED25519 256]--+
|       .. ..   .oXX|
|      .  .  . .=.X|
|  o   o .   o+ B.|
|   E   * . o. + +|
|  . = o S o .. . |
|     o = o ..    |
|        o  o o   |
|         o. +.o  |
|          .ooo   |
+----[SHA256]-----+
```

- I generated an ***Ed25519*** key pair and transferred the public key securely to the server's `authorized_keys` file.

viii. Reconnecting to the server

```
┌──(dinuja㉿kali)-[~]
└─$ ssh -p 2222 dinuja@192.168.100.1
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.11.0-25-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

239 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

10 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

Last login: Mon May 12 00:12:52 2025 from 192.168.100.21
dinuja@dinuja-VirtualBox:~$ ls
check1       Documents  ex2    ex44   IT23547742      Music     SNP1     Templates   Videos
check123     Downloads  ex3    ex4.c  lab2_ex2.c      Pictures  snp3     testfolder
check1.save  ex1        ex3.c  ex5    log_backups     Public    SNP3     time
Desktop      ex1.c      ex4    ex5.c  log_cleanup.sh  snap      student  time.c
dinuja@dinuja-VirtualBox:~$ cd SNP3
dinuja@dinuja-VirtualBox:~/SNP3$ cd snp
dinuja@dinuja-VirtualBox:~/SNP3/snp$ touch created_by_kali_using_ssh.kali
dinuja@dinuja-VirtualBox:~/SNP3/snp$ ls
created_by_kali_using_ssh.kali  log_backups  log_cleanup.sh
dinuja@dinuja-VirtualBox:~/SNP3/snp$
```

ix. Logging out

```
dinuja@dinuja-VirtualBox:~/SNP3/snp$ exit
logout
Connection to 192.168.100.1 closed.
```

3.3 Firewall rules (iptables)

i. Clearing existing tables

```
dinuja@dinuja-VirtualBox:~$ sudo iptables -F
```

ii. Blocking social media (Facebook, Instagram, Twitter)

```
dinuja@dinuja-VirtualBox:~$ sudo iptables -A OUTPUT -p tcp -d facebook.com -j DROP
dinuja@dinuja-VirtualBox:~$ sudo iptables -A OUTPUT -p tcp -d instagram.com -j DROP
dinuja@dinuja-VirtualBox:~$ sudo iptables -A OUTPUT -p tcp -d twitter.com -j DROP
```

- **`-A OUTPUT`**: Applies to outgoing traffic
- **`-p tcp`**: Blocks TCP connections
- **`-d facebook.com`**: Target domain
- **`-j DROP`**: Discards packets

iii. Allow HTTPS but block HTTP

```
dinuja@dinuja-VirtualBox:~$ sudo iptables -A OUTPUT -p tcp --dport 80 -j DROP
dinuja@dinuja-VirtualBox:~$ sudo iptables -A OUTPUT -p tcp --dport 443 -j ACCEPT
```

- **`-dport 80`**: HTTP port
- **`-dport 443`**: HTTPS port
- **`-j ACCEPT`**: Accepts packets

iv. Verifying rules

```
dinuja@dinuja-VirtualBox:~$ sudo iptables -L -v
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 175 packets, 26182 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 DROP       tcp  --  any    any     anywhere             edge-star-mini-shv-01-sin11.facebook.com
    0     0 DROP       tcp  --  any    any     anywhere             instagram-p42-shv-03-sin6.fbcdn.net
    0     0 DROP       tcp  --  any    any     anywhere             172.66.0.227
    0     0 DROP       tcp  --  any    any     anywhere             anywhere             tcp dpt:http
   67 11966 ACCEPT     tcp  --  any    any     anywhere             anywhere             tcp dpt:https
    0     0 DROP       tcp  --  any    any     anywhere             anywhere             tcp dpt:http
    0     0 ACCEPT     tcp  --  any    any     anywhere             anywhere             tcp dpt:https
dinuja@dinuja-VirtualBox:~$
```

3.4 Web Server (Apache)

i. Installing Apache

```
dinuja@dinuja-VirtualBox:~$ sudo apt install apache2 -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

ii. Checking the web root directory

```
dinuja@dinuja-VirtualBox:~$ ls /var/www/html
index.html
```

- Since Apache provides a default index.html file, I will create a separate HTML file for this implementation to avoid overwriting the default page.

iii. Creating a separate personal file

```
dinuja@dinuja-VirtualBox:~$ code /var/www/html/dinuja-index.html
dinuja@dinuja-VirtualBox:~$
```

iv. Writing the HTML code

```
<!DOCTYPE html>
<html>
    <head>
        <title>My Apache Web server page</title>
        <style>
            body{
                font-family: Arial, Helvetica, sans-serif;
                max-width: 800px;
                margin: 0 auto;
                padding: 20px;
            }
            header {
                background: #181f25;
                color: white;
                padding: 10px 20px;
                border-radius: 5px;
            }
        </style>
    </head>
    <body>
        <header>
            <h1>SNP - Assignment</h1>
            <p>By Dinuja Elesinghe</p>
            <p>IT23547742</p>
        </header>

        <section>
            <h2>Server Configuration Details</h2>
            <ul>
                <li><strong>IP Address:</strong>192.168.100.1</li>
                <li><strong>Web Server</strong>Apache</li>
            </ul>
        </section>
    </body>
</html>
```

v. Setting up correct permissions

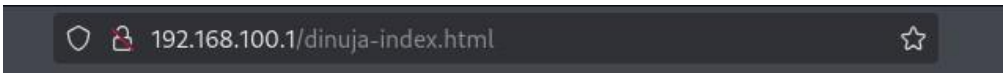
```
dinuja@dinuja-VirtualBox:~$ sudo chown www-data:www-data /var/www/html/dinuja-index.html
[sudo] password for dinuja:
dinuja@dinuja-VirtualBox:~$ sudo chmod 644 /var/www/html/dinuja-index.html
dinuja@dinuja-VirtualBox:~$
```

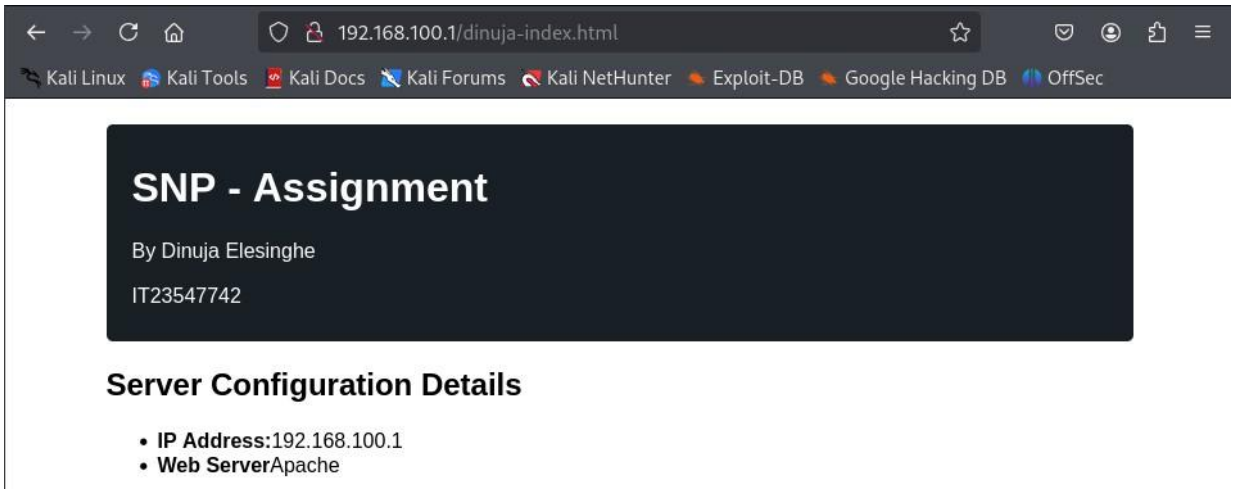
- In the first command the ownership of the dinuja-index.html file is given to www.data user.
- Because Apache runs as www.data user.
- In the second command sets file permissions to:
    - 6 (Owner: Read + Write)
    - 4 (Group: Read-only)
    - 4 (Others: Read-only)

```
dinuja@dinuja-VirtualBox:~$ ls -l /var/www/html/dinuja-index.html
-rw-r--r-- 1 www-data www-data 911 May 12 13:07 /var/www/html/dinuja-index.html
```

- We can further confirm that by running `ls` command

vi. Accessing the server using a client machine





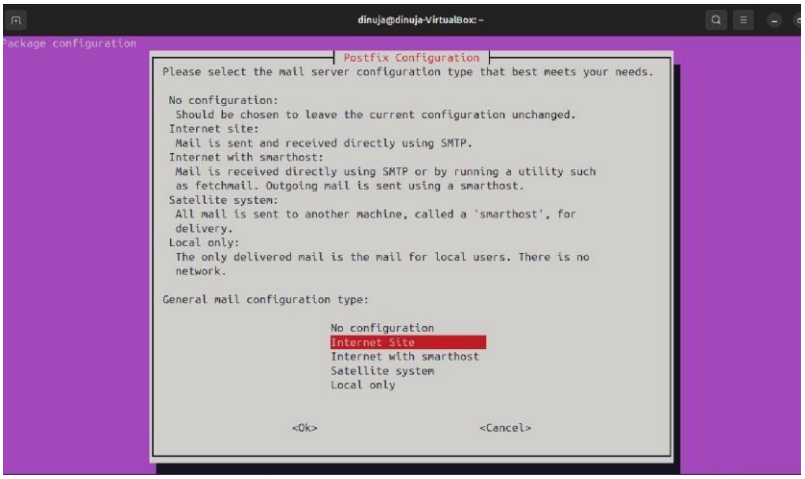
- You can access the server by typing the server IP address and the file in the web root directory in the address bar of your preferred browser.

3.5 Email Server (Postfix)

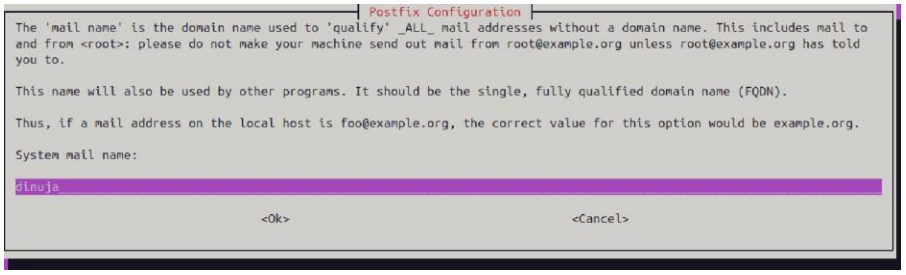
i. Installing Postfix

```
dinuja@dinuja-VirtualBox:~$ sudo apt install postfix mailutils -y
[sudo] password for dinuja:
Reading package lists... Done
```

- After the installation process a prompt will pop up
- Select **Internet site** from that list of options then hit enter



- In the next prompt enter your hostname



ii. Configuring Postfix

```
dinuja@dinuja-VirtualBox:~$ sudo nano /etc/postfix/main.cf
dinuja@dinuja-VirtualBox:~$
```

- I kept the default settings here.

iii. Restarting and checking status

```
dinuja@dinuja-VirtualBox:~$ sudo systemctl status postfix
● postfix.service - Postfix Mail Transport Agent
     Loaded: loaded (/usr/lib/systemd/system/postfix.service; enabled; preset: enabled)
     Active: active (exited) since Mon 2025-05-12 13:30:50 +0530; 12s ago
       Docs: man:postfix(1)
    Process: 57673 ExecStart=/bin/true (code=exited, status=0/SUCCESS)
```

iv. Testing functionality

```
dinuja@dinuja-VirtualBox:~$ echo "Testing Postfix" | mail -s "Test from $(date +%Y%M%D) Assignment" dinujaddk@gmail.com
```

- When attempting to send an email to my personal Gmail account, the message was rejected because it originated from a local server without proper authentication.
- Although this issue can be resolved by configuring Gmail as a relay host, the assignment criteria do not require this setup. Therefore, an alternative method will be used to test the mail functionality.
- To verify the functionality of the Postfix server, I will install and use **Mutt**, a lightweight email client commonly used on UNIX/Linux systems.

```
dinuja@dinuja-VirtualBox:~$ sudo apt install mutt
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

- After installing it I sent a message to myself using the Postfix server.

```
dinuja@dinuja-VirtualBox:~$ echo "Hello testing postfix" | mail -s "Testing functionality" dinuja
```

- Then I checked the mail logs

```
dinuja@dinuja-VirtualBox:~$ sudo tail -f /var/log/mail.log
```

- The following log entry confirms that the message was sent successfully.

```
2025-05-12T13:49:39.083633+05:30 dinuja-VirtualBox postfix/local[58876]: 0CE3B168214: to=<dinuja@dinuja-VirtualBox>, orig_to=<di
nuja>, relay=local, delay=0.14, delays=0.13/0/0/0.01, dsn=2.0.0, status=sent (delivered to mailbox)
```

- Here using mutt I can view the mail I sent before.

```
dinuja@dinuja-VirtualBox:~$ mutt -f /var/mail/dinuja
```

```
i:Exit  -:PrevPg  <Space>:NextPg v:View Attachm.  d:Del
Date: Mon, 12 May 2025 13:49:38 +0530
From: Dinuja <dinuja@dinuja-VirtualBox>
To: dinuja@dinuja-VirtualBox
Subject: Testing functionality
User-Agent: mail (GNU Mailutils 3.17)

Hello testing postfix
```

4. Linux GDB

4.1 Execution Process

i. Extracting the downloaded zip file

```
dinuja@dinuja-VirtualBox:~/SNP3/snp$ ls
created_by_kali_using_ssh.kali  Executables.zip  GDB  log_backups  log_cleanup.sh
dinuja@dinuja-VirtualBox:~/SNP3/snp$
```

```
dinuja@dinuja-VirtualBox:~/SNP3/snp$ unzip ./Executables.zip -d ./GDB
Archive:  ./Executables.zip
   creating: ./GDB/Executables/
  inflating: ./GDB/Executables/ARM
  inflating: ./GDB/__MACOSX/Executables/._ARM
  inflating: ./GDB/Executables/x86_64
  inflating: ./GDB/__MACOSX/Executables/._x86_64
dinuja@dinuja-VirtualBox:~/SNP3/snp$
```

ii. Verifying system architecture

```
dinuja@dinuja-VirtualBox:~/SNP3/snp$ uname -m
x86_64
dinuja@dinuja-VirtualBox:~/SNP3/snp$
```

iii. Changing permissions so the file is executable

```
dinuja@dinuja-VirtualBox:~/SNP3/snp/GDB$ ls Executables
ARM  x86_64
```

- Since the file was not executable I used `chmod` to grant execute permissions.

```
dinuja@dinuja-VirtualBox:~/SNP3/snp/GDB/Executables$ chmod +x x86_64
dinuja@dinuja-VirtualBox:~/SNP3/snp/GDB/Executables$ ls
ARM  x86_64
```

iv. Executing the file using root privileges

```
dinuja@dinuja-VirtualBox:~/SNP3/snp/GDB/Executables$ sudo ./x86_64
[sudo] password for dinuja:
Enter the student IT number: IT23547742
```

- This created a new executable file with my IT number as the filename

```
dinuja@dinuja-VirtualBox:~/SNP3/snp/GDB/Executables$ ls
ARM  IT23547742
```

v. Running the new file

```
dinuja@dinuja-VirtualBox:~/SNP3/snp/GDB/Executables$ ls -lh
total 96K
-rw-rw-r-- 1 dinuja dinuja 70K Mar 17 06:26 ARM
-rw-rw-r-- 1 dinuja dinuja  36 May 12 18:33 data.txt
-rwxr-xr-x 1 root   root   20K May 12 18:26 IT23547742
```

- When I ran the new file it created a file named *data.txt*

4.2 Initial Analysis

i. Using `cat` to read the content

```
dinuja@dinuja-VirtualBox:~/SNP3/snp/GDB/Executables$ cat data.txt
SO[SPT
W
dinuja@dinuja-VirtualBox:~/SNP3/snp/GDB/Executables$
```

- Before going through the debugging process, I tried reading file content using `cat`.

ii. Checking file type

```
dinuja@dinuja-VirtualBox:~/SNP3/snp/GDB/Executables$ file data.txt
data.txt: data
```

- Upon further analysis using the file command, it was determined that the file contains raw binary data.
- This suggests that the file may be encrypted or contain non-textual, encoded information.

iii. Using hexdump

```
dinuja@dinuja-VirtualBox:~/SNP3/snp/GDB/Executables$ xxd data.txt
00000000: 5303 4f5b 0118 5350 540a 571c 0a48 4a59  S.O[..SPT.W..HJY
00000010: 511d 4604 4b52 5c54 0f55 4153 5c41 0d54  Q.F.KR\T.UAS\A.T
00000020: 1b52 5641                                .RVA
```

- Then I used hexdump on the *data.txt* file.

iv. Checking for readable text

```
dinuja@dinuja-VirtualBox:~/SNP3/snp/GDB/Executables$ strings data.txt
HJYQ
KR\T
UAS\A
dinuja@dinuja-VirtualBox:~/SNP3/snp/GDB/Executables$
```

- Further analysis using the strings command revealed that the only readable characters in data.txt are a few uppercase English letters and slashes.

- This limited and non-meaningful output does not provide a clear understanding of the file's contents and further supports the likelihood that the file is encrypted.

v. Checking file metadata

```
dinuja@dinuja-VirtualBox:~/SNP3/snp/GDB/Executables$ stat data.txt
  File: data.txt
  Size: 36              Blocks: 8          IO Block: 4096   regular file
Device: 8,2     Inode: 1473282     Links: 1
Access: (0664/-rw-rw-r--)  Uid: ( 1000/  dinuja)   Gid: ( 1000/  dinuja)
Access: 2025-05-13 02:25:11.529674518 +0530
Modify: 2025-05-13 02:25:06.235674866 +0530
Change: 2025-05-13 02:25:06.235674866 +0530
 Birth: 2025-05-12 18:33:09.423827983 +0530
dinuja@dinuja-VirtualBox:~/SNP3/snp/GDB/Executables$
```

- The output indicates that the file data.txt has not been modified, nor have its permissions been changed since its creation.
- As no additional useful information could be extracted—aside from the assumption that the contents of data.txt are encrypted—I proceeded to analyze the IT23547742 executable file for further insights.

vi. Checking file type of the executable file

```
dinuja@dinuja-VirtualBox:~/SNP3/snp/GDB/Executables$ file ./IT23547742
./IT23547742: ELF 64-bit LSB pie executable, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/ld-linux-x86-64.so
.2, BuildID[sha1]=b4210e0a3295e34b89680f24cb79187fe81ae79a, for GNU/Linux 3.2.0, with debug_info, not stripped
```

- First I checked for the file type of *IT23547742* file.
- According to the output the file is an ***ELF*** file, which is used for
    - Executable files
    - Shared libraries
    - Object files

vii. Checking for readable text in the executable file

- I then used the strings command to examine the executable for any recognizable or readable strings that might provide insight into its functionality.

```
dinuja@dinuja-VirtualBox:~/SNP3/snp/GDB/Executables$ strings IT23547742
/lib64/ld-linux-x86-64.so.2
fgets
__stack_chk_fail
fopen
```

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
void xor_encrypt_decrypt(char *data, const char *key) {
    size_t data_len = strlen(data);
    size_t key_len = strlen(key);
    for (size_t i = 0; i < data_len; i++) {
        data[i] ^= key[i % key_len];
    }
int main() {
    FILE *fp = popen("sudo cat /sys/class/dmi/id/product_uuid", "r");
    if (fp == NULL) {
        printf("Error retrieving data\n");
        return 1;
    }
    char uuid[50];
    fgets(uuid, sizeof(uuid), fp);
    pclose(fp);
    uuid[strcspn(uuid, "\n")] = 0; // Remove newline
    const char *key = "key";
    xor_encrypt_decrypt(uuid, key);
    FILE *out = fopen("data.txt", "w");
    if (out == NULL) {
        printf("Error creating data.txt!\n");
        return 1;
    }
    fprintf(out, "%s", uuid);
    fclose(out);
    return 0;
```

- As a result of this step, I discovered that the entire source code is embedded within the executable file, which is an unusual occurrence.

- While having access to the source code would allow me to easily understand its purpose and potentially decrypt the content in data.txt, the assignment criteria specify that I should debug the executable. Therefore, I will proceed with debugging the executable as the primary method.

4.3 Debugging Process

4.3.1 Initial Debugging

i. Starting the debugging process

```
dinuja@dinuja-VirtualBox:~/SNP3/snp/GDB/Executables$ gdb -q IT23547742
Reading symbols from IT23547742...
```

- First I used GDB on the executable

ii. Checking for functions

```
(gdb) info function
All defined functions:

File IT23547742.c:
13:     int main();
5:      void xor_encrypt_decrypt(char *, const char *);

Non-debugging symbols:
0x0000000000001000  _init
0x00000000000010d0  __cxa_finalize@plt
0x00000000000010e0  puts@plt
0x00000000000010f0  fclose@plt
0x0000000000001100  strlen@plt
0x0000000000001110  __stack_chk_fail@plt
0x0000000000001120  pclose@plt
0x0000000000001130  fputs@plt
0x0000000000001140  strcspn@plt
0x0000000000001150  fgets@plt
0x0000000000001160  popen@plt
0x0000000000001170  fopen@plt
0x0000000000001180  _start
0x00000000000011b0  deregister_tm_clones
0x00000000000011e0  register_tm_clones
0x0000000000001220  __do_global_dtors_aux
--Type <RET> for more, q to quit, c to continue without paging--
```

- As shown in the image above, analysis of the source code revealed two functions.
    i. `Int main ()` - *the entry point of the program.*
    ii. `void xor_encrypt_decrypt(char *, const char *)` - *a function responsible for performing XOR-based encryption and decryption on the given data using the specified key.*

- This supports the earlier assumption that the contents of data.txt are indeed encrypted.
- Given that the function utilizes XOR logic, the encryption and decryption processes are symmetric—applying the same operation with the same key will correctly decrypt the data.

iii. Disassembling main function

I changed the syntax style of disassembled machine instructions to Intel syntax.

```
(gdb) set disassembly-flavor intel
(gdb) disas main
Dump of assembler code for function main:
   0x00000000000012f0 <+0>:     endbr64
   0x00000000000012f4 <+4>:     push   rbp
   0x00000000000012f5 <+5>:     mov    rbp,rsp
   0x00000000000012f8 <+8>:     sub    rsp,0x60
   0x00000000000012fc <+12>:    mov    rax,QWORD PTR fs:0x28
   0x0000000000001305 <+21>:    mov    QWORD PTR [rbp-0x8],rax
   0x0000000000001309 <+25>:    xor    eax,eax
   0x000000000000130b <+27>:    lea    rax,[rip+0xcf6]        # 0x2008
   0x0000000000001312 <+34>:    mov    rsi,rax
   0x0000000000001315 <+37>:    lea    rax,[rip+0xcf4]        # 0x2010
   0x000000000000131c <+44>:    mov    rdi,rax
   0x000000000000131f <+47>:    call   0x1160 <popen@plt>
   0x0000000000001324 <+52>:    mov    QWORD PTR [rbp-0x58],rax
   0x0000000000001328 <+56>:    cmp    QWORD PTR [rbp-0x58],0x0
   0x000000000000132d <+61>:    jne    0x1348 <main+88>
   0x000000000000132f <+63>:    lea    rax,[rip+0xd02]        # 0x2038
   0x0000000000001336 <+70>:    mov    rdi,rax
   0x0000000000001339 <+73>:    call   0x10e0 <puts@plt>
   0x000000000000133e <+78>:    mov    eax,0x1
   0x0000000000001343 <+83>:    jmp    0x1400 <main+272>
   0x0000000000001348 <+88>:    mov    rdx,QWORD PTR [rbp-0x58]
   0x000000000000134c <+92>:    lea    rax,[rbp-0x40]
   0x0000000000001350 <+96>:    mov    esi,0x32
   0x0000000000001355 <+101>:   mov    rdi,rax
--Type <RET> for more, q to quit, c to continue without paging--
```

```
--Type <RET> for more, q to quit, c to continue without paging--<RET>
   0x0000000000001358 <+104>:    call   0x1150 <fgets@plt>
   0x000000000000135d <+109>:    mov    rax,QWORD PTR [rbp-0x58]
   0x0000000000001361 <+113>:    mov    rdi,rax
   0x0000000000001364 <+116>:    call   0x1120 <pclose@plt>
   0x0000000000001369 <+121>:    lea    rax,[rbp-0x40]
   0x000000000000136d <+125>:    lea    rdx,[rip+0xcda]        # 0x204e
   0x0000000000001374 <+132>:    mov    rsi,rdx
   0x0000000000001377 <+135>:    mov    rdi,rax
   0x000000000000137a <+138>:    call   0x1140 <strcspn@plt>
   0x000000000000137f <+143>:    mov    BYTE PTR [rbp+rax*1-0x40],0x0
   0x0000000000001384 <+148>:    lea    rax,[rip+0xcc5]        # 0x2050
   0x000000000000138b <+155>:    mov    QWORD PTR [rbp-0x50],rax
   0x000000000000138f <+159>:    mov    rdx,QWORD PTR [rbp-0x50]
   0x0000000000001393 <+163>:    lea    rax,[rbp-0x40]
   0x0000000000001397 <+167>:    mov    rsi,rdx
   0x000000000000139a <+170>:    mov    rdi,rax
   0x000000000000139d <+173>:    call   0x1269 <xor_encrypt_decrypt>
   0x00000000000013a2 <+178>:    lea    rax,[rip+0xcab]        # 0x2054
   0x00000000000013a9 <+185>:    mov    rsi,rax
   0x00000000000013ac <+188>:    lea    rax,[rip+0xca3]        # 0x2056
   0x00000000000013b3 <+195>:    mov    rdi,rax
   0x00000000000013b6 <+198>:    call   0x1170 <fopen@plt>
   0x00000000000013bb <+203>:    mov    QWORD PTR [rbp-0x48],rax
   0x00000000000013bf <+207>:    cmp    QWORD PTR [rbp-0x48],0x0
   0x00000000000013c4 <+212>:    jne    0x13dc <main+236>
   0x00000000000013c6 <+214>:    lea    rax,[rip+0xc92]        # 0x205f
   0x00000000000013cd <+221>:    mov    rdi,rax
   0x00000000000013d0 <+224>:    call   0x10e0 <puts@plt>
   0x00000000000013d5 <+229>:    mov    eax,0x1
   0x00000000000013da <+234>:    jmp    0x1400 <main+272>
```

```
--Type <RET> for more, q to quit, c to continue without paging--<RET>
   0x00000000000013dc <+236>:    mov    rdx,QWORD PTR [rbp-0x48]
   0x00000000000013e0 <+240>:    lea    rax,[rbp-0x40]
   0x00000000000013e4 <+244>:    mov    rsi,rdx
   0x00000000000013e7 <+247>:    mov    rdi,rax
   0x00000000000013ea <+250>:    call   0x1130 <fputs@plt>
   0x00000000000013ef <+255>:    mov    rax,QWORD PTR [rbp-0x48]
   0x00000000000013f3 <+259>:    mov    rdi,rax
   0x00000000000013f6 <+262>:    call   0x10f0 <fclose@plt>
   0x00000000000013fb <+267>:    mov    eax,0x0
   0x0000000000001400 <+272>:    mov    rdx,QWORD PTR [rbp-0x8]
   0x0000000000001404 <+276>:    sub    rdx,QWORD PTR fs:0x28
   0x000000000000140d <+285>:    je     0x1414 <main+292>
   0x000000000000140f <+287>:    call   0x1110 <__stack_chk_fail@plt>
   0x0000000000001414 <+292>:    leave
   0x0000000000001415 <+293>:    ret
End of assembler dump.
```

iv. Disassembling the xor function

```
Dump of assembler code for function xor_encrypt_decrypt:
   0x0000000000001269 <+0>:     endbr64
   0x000000000000126d <+4>:     push   rbp
   0x000000000000126e <+5>:     mov    rbp,rsp
   0x0000000000001271 <+8>:     sub    rsp,0x30
   0x0000000000001275 <+12>:    mov    QWORD PTR [rbp-0x28],rdi
   0x0000000000001279 <+16>:    mov    QWORD PTR [rbp-0x30],rsi
   0x000000000000127d <+20>:    mov    rax,QWORD PTR [rbp-0x28]
   0x0000000000001281 <+24>:    mov    rdi,rax
   0x0000000000001284 <+27>:    call   0x1100 <strlen@plt>
   0x0000000000001289 <+32>:    mov    QWORD PTR [rbp-0x10],rax
   0x000000000000128d <+36>:    mov    rax,QWORD PTR [rbp-0x30]
   0x0000000000001291 <+40>:    mov    rdi,rax
   0x0000000000001294 <+43>:    call   0x1100 <strlen@plt>
   0x0000000000001299 <+48>:    mov    QWORD PTR [rbp-0x8],rax
   0x000000000000129d <+52>:    mov    QWORD PTR [rbp-0x18],0x0
   0x00000000000012a5 <+60>:    jmp    0x12e2 <xor_encrypt_decrypt+121>
   0x00000000000012a7 <+62>:    mov    rdx,QWORD PTR [rbp-0x28]
   0x00000000000012ab <+66>:    mov    rax,QWORD PTR [rbp-0x18]
   0x00000000000012af <+70>:    add    rax,rdx
   0x00000000000012b2 <+73>:    movzx  esi,BYTE PTR [rax]
   0x00000000000012b5 <+76>:    mov    rax,QWORD PTR [rbp-0x18]
   0x00000000000012b9 <+80>:    mov    edx,0x0
   0x00000000000012be <+85>:    div    QWORD PTR [rbp-0x8]
   0x00000000000012c2 <+89>:    mov    rax,QWORD PTR [rbp-0x30]
   0x00000000000012c6 <+93>:    add    rax,rdx
   0x00000000000012c9 <+96>:    movzx  ecx,BYTE PTR [rax]
   0x00000000000012cc <+99>:    mov    rdx,QWORD PTR [rbp-0x28]
   0x00000000000012d0 <+103>:   mov    rax,QWORD PTR [rbp-0x18]
   0x00000000000012d4 <+107>:   add    rax,rdx
   0x00000000000012d7 <+110>:   xor    esi,ecx
   0x00000000000012d9 <+112>:   mov    edx,esi
   0x00000000000012db <+114>:   mov    BYTE PTR [rax],dl
   0x00000000000012dd <+116>:   add    QWORD PTR [rbp-0x18],0x1
   0x00000000000012e2 <+121>:   mov    rax,QWORD PTR [rbp-0x18]
   0x00000000000012e6 <+125>:   cmp    rax,QWORD PTR [rbp-0x10]
   0x00000000000012ea <+129>:   jb     0x12a7 <xor_encrypt_decrypt+62>
   0x00000000000012ec <+131>:   nop
   0x00000000000012ed <+132>:   nop
   0x00000000000012ee <+133>:   leave
   0x00000000000012ef <+134>:   ret
End of assembler dump.
```

v. Setting Breakpoints

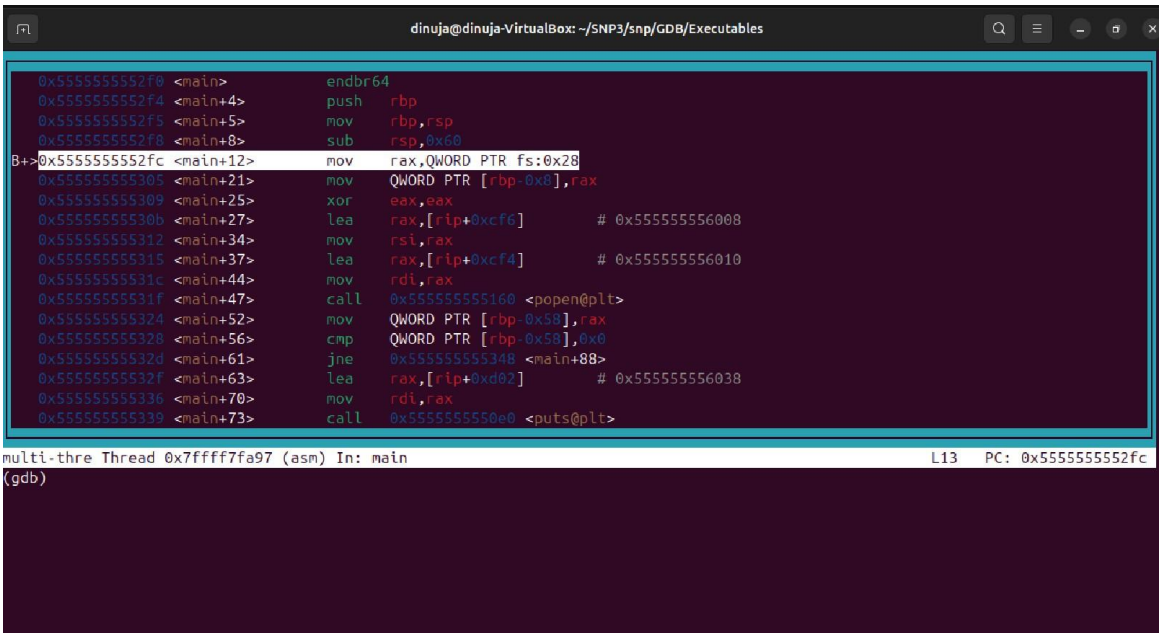
```
(gdb) break main

This GDB supports auto-downloading debuginfo from the following URLs:
  <https://debuginfod.ubuntu.com>
Enable debuginfod for this session? (y or [n]) y
Debuginfod has been enabled.
To make this setting permanent, add 'set debuginfod enabled on' to .gdbinit.
Downloading source file /home/dinuja/SNP3/snp/GDB/Executables/IT23547742.c
Breakpoint 1 at 0x12fc: file IT23547742.c, line 13.
(gdb)
```

```
(gdb) break xor_encrypt_decrypt
Breakpoint 2 at 0x127d: file IT23547742.c, line 6.
(gdb)
```

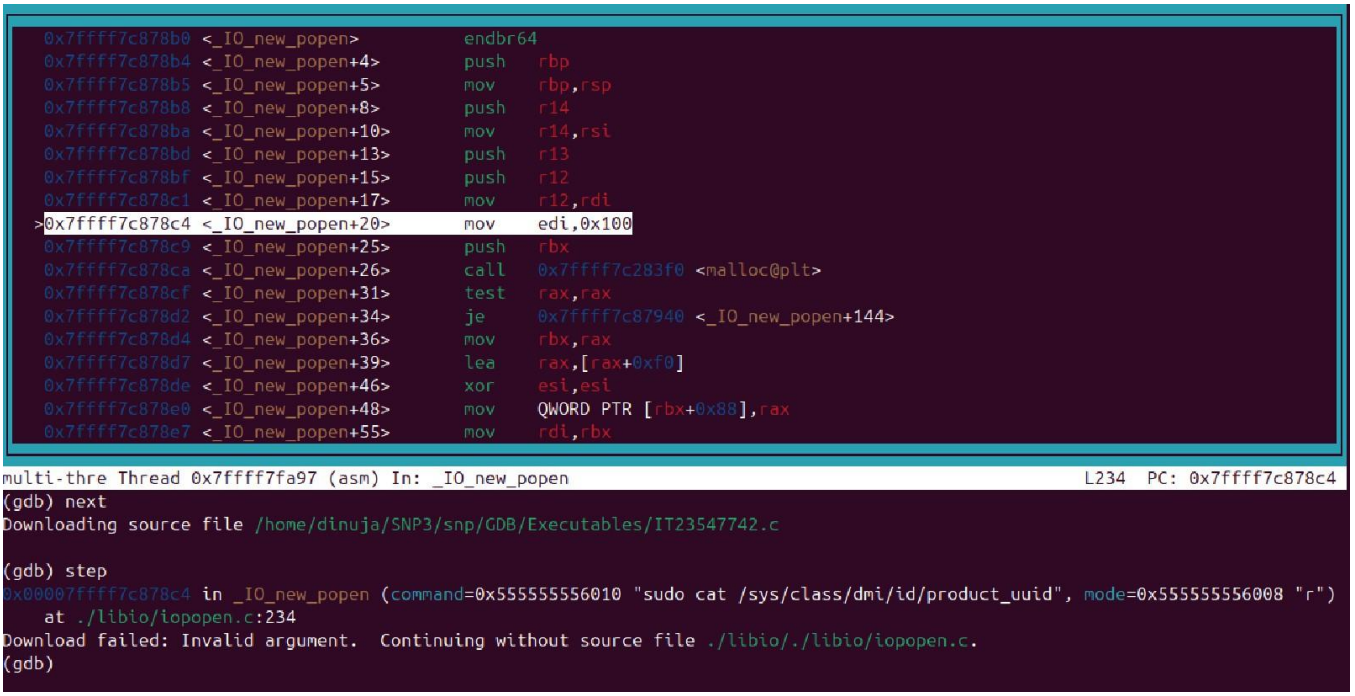
- **After setting breakpoints, I ran these commands .**

```
(gdb) run
Starting program: /home/dinuja/SNP3/snp/GDB/Executables/IT23547742
Downloading separate debug info for system-supplied DSO at 0x7ffff7fc3000
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".

Breakpoint 1, main () at IT23547742.c:13
warning: 13     IT23547742.c: No such file or directory
(gdb)
```

Since I cannot understand anything from this result, I got the layout in assemble format using "layout asm" command.



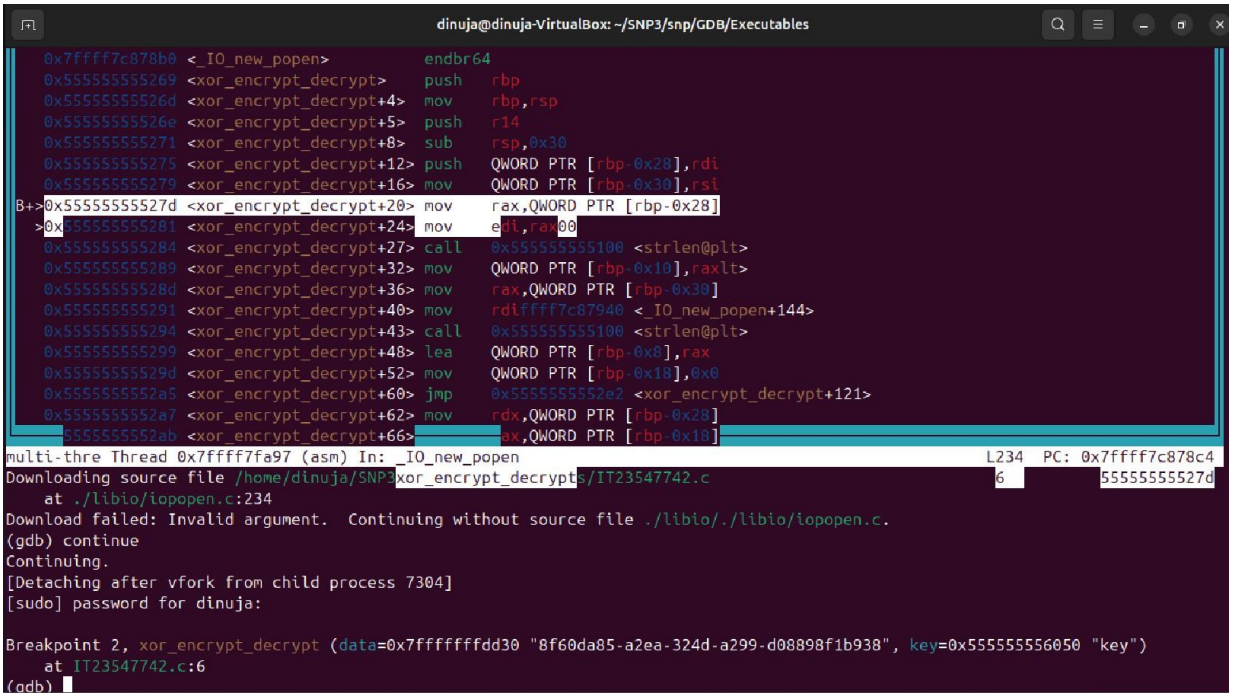
Now we are at breakpoint 1



- Next - Executes the current line of code without stepping into function calls.
- Step - Executes the current line of code and steps into functions.

I used “continue” command to get to breakpoint 2.

Breakpoint 2



As you can see here, now we are at breakpoint 2, where xor_encrypt_decrypt function is. You can clearly see what the inputs that this function is use.

- You can see the first parameter (value to be encrypted) and its memory address.
- Second parameter, the key that use for this encryption, which is also “key”.

```
0x5555555553ea <main+250>       call   0x555555555130 <fputs@plt>
```

- This writes encrypted UUID to data.txt

1.1.1 End the debugging session

```
(gdb) q
A debugging session is active.

        Inferior 1 [process 7189] will be killed.

Quit anyway? (y or n)
```

1.1.2 File System Analysis

1. Using cat to read the content

```
dinuja@dinuja-VirtualBox:~/SNP3/snp/GDB/Executables$ cat data.txt
SO[SPT
W
TVAdinuja@dinuja-VirtualBox:~/SNP3/snp/GDB/Executables$
```

2. Using hexdump

```
TVAdinuja@dinuja-VirtualBox:~/SNP3/snp/GDB/Executables$ hexdump data.txt
0000000 0353 5b4f 1801 5053 0a54 1c57 480a 594a
0000010 1d51 0446 524b 545c 550f 5341 415c 540d
0000020 521b 4156
0000024
```

3. Checking for readable text

```
dinuja@dinuja-VirtualBox:~/SNP3/snp/GDB/Executables$ strings data.txt
HJYQ
KR\T
UAS\A
dinuja@dinuja-VirtualBox:~/SNP3/snp/GDB/Executables$
```

4. Checking file metadata

```
dinuja@dinuja-VirtualBox:~/SNP3/snp/GDB/Executables$ stat data.txt
  File: data.txt
  Size: 36              Blocks: 8          IO Block: 4096   regular file
Device: 8,2     Inode: 1473282     Links: 1
Access: (0664/-rw-rw-r--)  Uid: ( 1000/  dinuja)   Gid: ( 1000/  dinuja)
Access: 2025-05-13 02:25:11.529674518 +0530
Modify: 2025-05-13 02:25:06.235674866 +0530
Change: 2025-05-13 02:25:06.235674866 +0530
 Birth: 2025-05-12 18:33:09.423827983 +0530
```

5. *Checking file metadata*

```
(gdb) info breakpoints
Num     Type           Disp Enb Address            What
1       breakpoint     keep y   0x000000000000131f in main at IT23547742.c:14
2       breakpoint     keep y   0x0000000000001358 in main at IT23547742.c:20
3       breakpoint     keep y   0x000000000000137a in main at IT23547742.c:23
4       breakpoint     keep y   0x000000000000139d in main at IT23547742.c:26
5       breakpoint     keep y   0x00000000000013b6 in main at IT23547742.c:27
6       breakpoint     keep y   0x00000000000013ea in main at IT23547742.c:32
```

6. *Monitoring File System Changes After Executable Execution*

```
dinuja@dinuja-VirtualBox:~/SNP3/snp/GDB/Executables$ ls -lt
total 96
-rw-rw-r-- 1 dinuja dinuja    36 May 13 02:25 data.txt
-rwxr-xr-x 1 root   root   20360 May 12 18:26 IT23547742
-rw-rw-r-- 1 dinuja dinuja 70824 Mar 17 06:26 ARM
dinuja@dinuja-VirtualBox:~/SNP3/snp/GDB/Executables$ ./IT23547742
[sudo] password for dinuja:
dinuja@dinuja-VirtualBox:~/SNP3/snp/GDB/Executables$ ls -lt
total 96
-rw-rw-r-- 1 dinuja dinuja    36 May 13 13:34 data.txt
-rwxr-xr-x 1 root   root   20360 May 12 18:26 IT23547742
-rw-rw-r-- 1 dinuja dinuja 70824 Mar 17 06:26 ARM
```

- Repeated execution of the IT23547742 executable modifies the data.txt file, as evidenced by the updated timestamps and the differing content between each run.

4.3.2 *Analysis of "data.txt"*

```
dinuja@dinuja-VirtualBox:~/SNP3/snp/GDB/Executables$ sudo nano IT23547742
```

```c
#include <stdlib.h>
#include <string.h>

void xor_encrypt_decrypt(char *data, const char *key) {
    size_t data_len = strlen(data);
    size_t key_len = strlen(key);
    for (size_t i = 0; i < data_len; i++) {
        data[i] ^= key[i % key_len];
    }
}

int main() {
    FILE *fp = popen("sudo cat /sys/class/dmi/id/product_uuid", "r");
    if (fp == NULL) {
        printf("Error retrieving data\n");
        return 1;
    }
    char uuid[50];
    fgets(uuid, sizeof(uuid), fp);
    pclose(fp);

    uuid[strcspn(uuid, "\n")] = 0; // Remove newline

    const char *key = "key";
    xor_encrypt_decrypt(uuid, key);
    FILE *out = fopen("data.txt", "w");
```

```
    FILE *out = fopen("data.txt", "w");
    if (out == NULL) {
        printf("Error creating data.txt!\n");
        return 1;
    }
    fprintf(out, "%s", uuid);
    fclose(out);

    return 0;
}
```

- Based on this observation, I can now confirm that the code is used to encrypt data using two parameters: one is the value to be encrypted, and the other is a constant key.

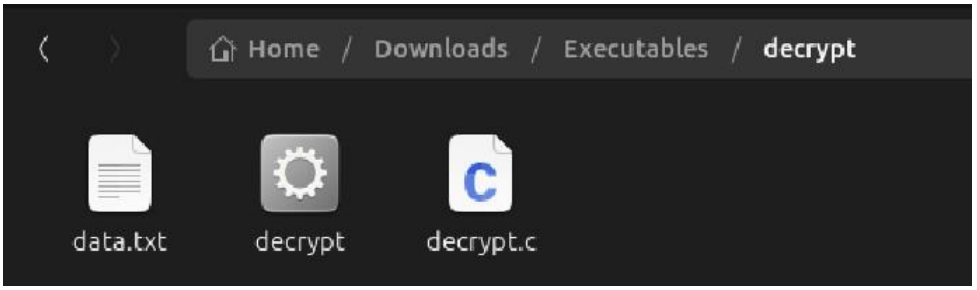
We can find the original code by using *sudo cat/sys/class/dmi/id_product*

```
dinuja@dinuja-VirtualBox:~/SNP3/snp/GDB/Executables$ sudo cat /sys/class/dmi/id/product_uuid
8f60da85-a2ea-324d-a299-d08898f1b938
```

- I created another c code to decrypt and get back this original code.
  - To do this I created a new folder, where I put both data.txt file and this new c file

```
dinuja@dinuja-VirtualBox:~/Downloads/Executables/decrypt$ touch decrypt.c
dinuja@dinuja-VirtualBox:~/Downloads/Executables/decrypt$ ls
data.txt  decrypt.c
```

```
dinuja@dinuja-VirtualBox:~/Downloads/Executables/decrypt$ code decrypt.c
```



• The code is on the next page.

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

void xor_decrypt(unsigned char *data, size_t len, const char *key)
{
    size_t key_len = strlen(key);
    for (size_t i = 0; i < len; i++) {
        data[i] ^= key[i % key_len];
    }
}

int main() {
    FILE *in = fopen("data.txt", "rb");
    if (in == NULL) {
        printf("Failed to open data.txt\n");
        return 1;
    }

    fseek(in, 0, SEEK_END);
    long size = ftell(in);
    rewind(in);

    unsigned char *buffer = malloc(size + 1);
    if (!buffer) {
        fclose(in);
        printf("Memory allocation failed\n");
        return 1;
    }

    fread(buffer, 1, size, in);
    fclose(in);
    buffer[size] = '\0'; // Null-terminate for safe printing

    const char *key = "key";
    xor_decrypt(buffer, size, key);

    printf("Decrypted output:\n%s\n", buffer);

    free(buffer);
    return 0;
}
```

```
dinuja@dinuja-VirtualBox:~/Downloads/Executables/decrypt$ gcc decrypt.c -o decrypt
dinuja@dinuja-VirtualBox:~/Downloads/Executables/decrypt$ ./decrypt
Decrypted output:
8f60da85-a2ea-324d-a299-d08898f1b938
dinuja@dinuja-VirtualBox:~/Downloads/Executables/decrypt$
```

VI. Tools Used for Analysis

• Cat – Display full contents

• String – Extract readable ASCII text from the file

• hexdump – View file as hexadecimal with ASCII

5.Conclusion

- Through the debugging process, it was determined that the analyzed program employs a XOR-based encryption algorithm.
- A detailed analysis of the program's execution flow and inspection of function parameters led to the successful identification of both the plaintext input and the encryption key.
- The encrypted output is stored in the file data.txt.
- To confirm these findings, a custom C program was created to decrypt the contents of data.txt using the extracted key. The program successfully recovered the original plaintext, validating the accuracy of the identified encryption method and associated parameters.