FILE SYSTEM SUPPORT

- Parrot Security OS supports several Linux file systems
- the default and most commonly used is:ext4

❖ **<u>why ext4?</u>**

✓ It is stable and mature, widely used in Linux environments.

✓ It supports large files and partitions.

✓ Provides journaling which helps protect data in case of power failure.

✓ It has good performance and is less prone to corruption.

❖**<u>Other file systems supported</u>**

✓Btrfs – advanced features like snapshotting and compression.

✓NTFS, FAT32, exFAT – for USB drives and compatibility with Windows systems.

✓ZFS HFS+, APFS – available but less common in Parrot setups.

ADVANTAGES OF PARROT OS

- The entire source code is accessible for anyone to read, modify, and contribute to.
- Designed with security in mind—offers full disk encryption, frequent updates, and privacy-protection tools.
- Users can choose from MATE, KDE, and Xfce according to their preference.
- Multiple editions available Security, Home, Studio, IoT, and Cloud allow users to pick one suited to their needs.
- Parrot OS is smaller in size than alternatives like Kali Linux and works with as little as 320MB of RAM.
- Parrot can be used for daily tasks with a good range of general-purpose applications.

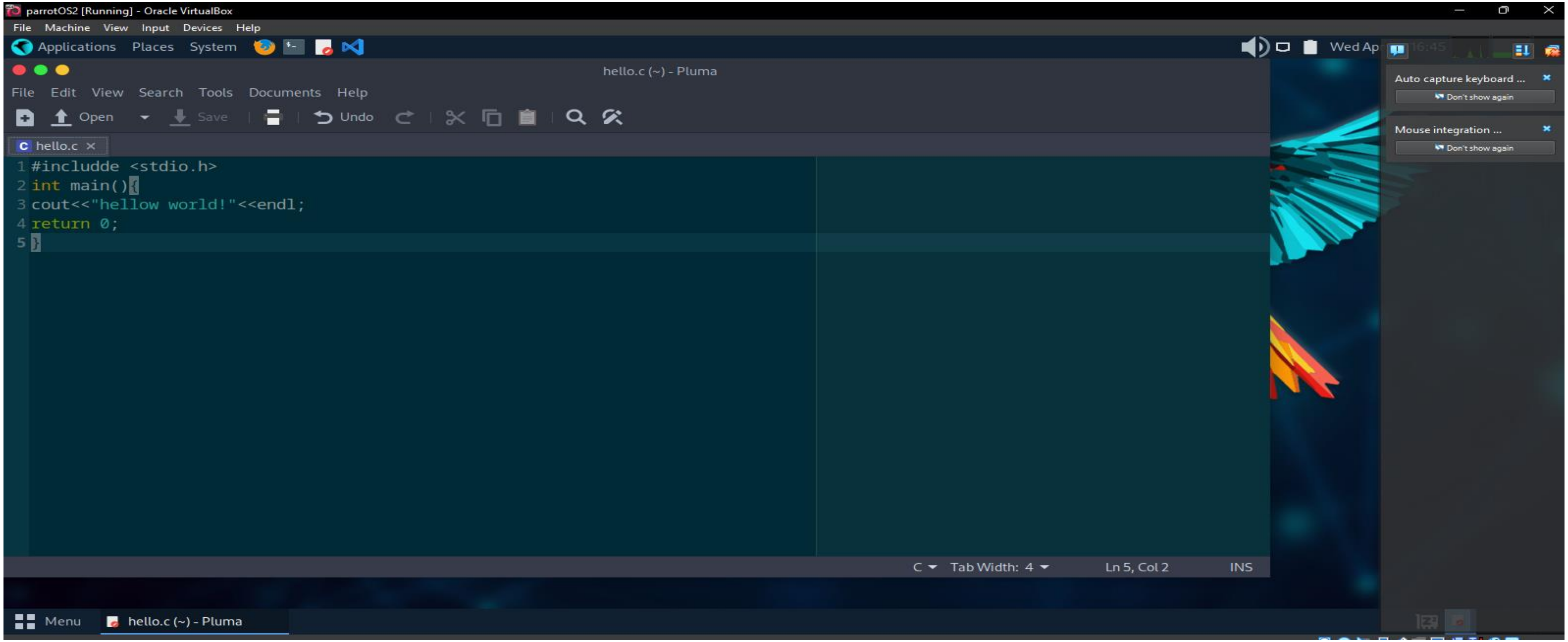
Disadvantages of Parrot OS

- Pre-installed tools can slow down older hardware due to increased storage and resource usage.
- Compared to Ubuntu or Kali, the community is still growing, so finding solutions or support can be a bit limited.
- The large number of tools and features may be confusing for users new to cybersecurity.

Implemented System Call

- A system call is a way for programs to request services from the operating system, such as accessing files, creating processes, or using hardware devices
- a bridge between the program and the OS kernel.

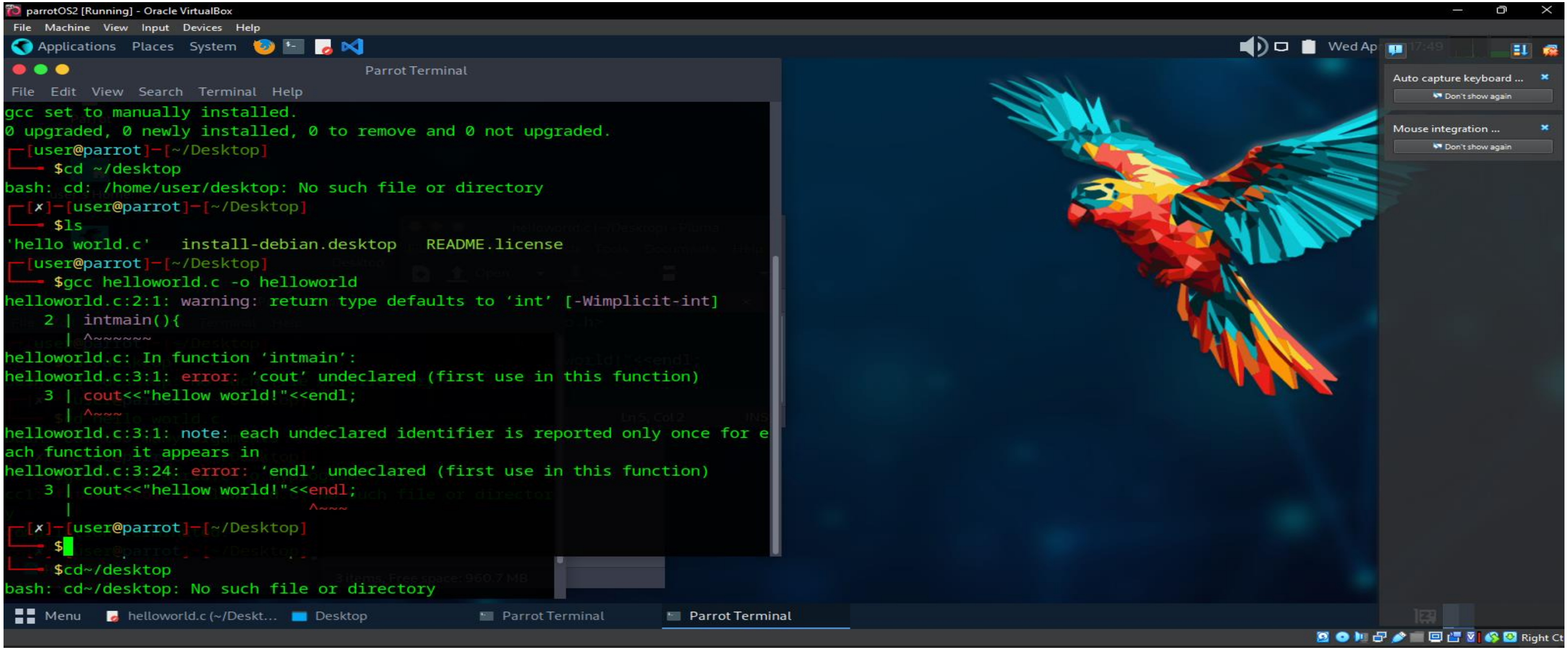
Step 1: Open Pluma,Write the Code and save.
pic 5.A:writting code on pluma



parrotOS2 [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Applications Places System
Wed Ap
hello.c (~) - Pluma
File Edit View Search Tools Documents Help
Open
Save
Undo
hello.c
Auto capture keyboard ...
Don't show again
Mouse integration ...
Don't show again
C
Tab Width: 4
Ln 5, Col 2
INS
Menu
hello.c (~) - Pluma

```
#includde <stdio.h>
int main(){
cout<<"hellow world!"<<endl;
return 0;
}
```

Step 2: Open Terminal : parrot icon, system tool, terminal pic 5.B:open terminal



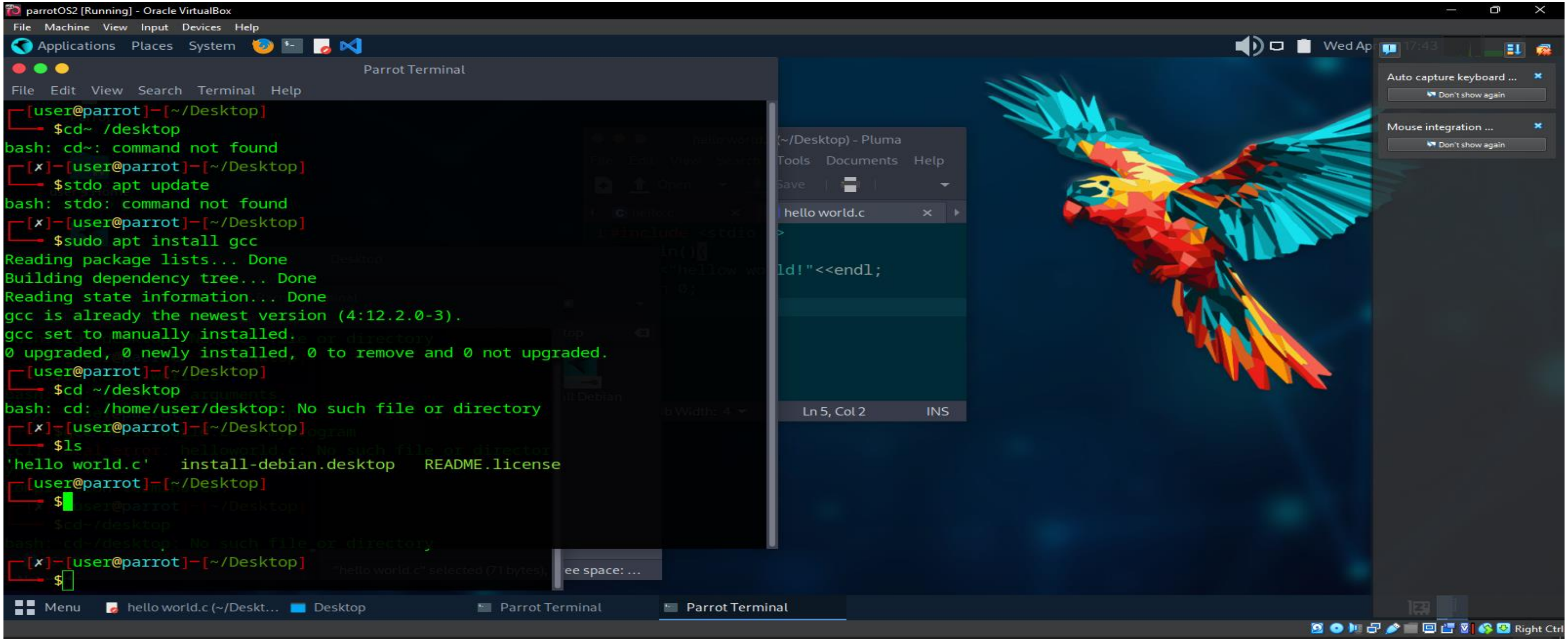
parrotOS2 [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Applications Places System
Wed Ap
Parrot Terminal
File Edit View Search Terminal Help
Auto capture keyboard ...
Don't show again
Mouse integration ...
Don't show again
Menu
helloworld.c (~/Deskt...
Desktop
Parrot Terminal
Parrot Terminal
Right Ct

```
gcc set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
┌─[user@parrot]─[~/Desktop]
└──╼ $cd ~/desktop
bash: cd: /home/user/desktop: No such file or directory
┌─[✗]─[user@parrot]─[~/Desktop]
└──╼ $ls
'hello world.c'   install-debian.desktop   README.license
┌─[user@parrot]─[~/Desktop]
└──╼ $gcc helloworld.c -o helloworld
helloworld.c:2:1: warning: return type defaults to 'int' [-Wimplicit-int]
    2 | intmain(){
      | ^~~~~~~
helloworld.c: In function 'intmain':
helloworld.c:3:1: error: 'cout' undeclared (first use in this function)
    3 | cout<<"hellow world!"<<endl;
      | ^~~~
helloworld.c:3:1: note: each undeclared identifier is reported only once for e
ach function it appears in
helloworld.c:3:24: error: 'endl' undeclared (first use in this function)
    3 | cout<<"hellow world!"<<endl;
      |                        ^~~~
┌─[✗]─[user@parrot]─[~/Desktop]
└──╼ $
└──╼ $cd~/desktop
bash: cd~/desktop: No such file or directory
```

Step 4: Navigate to the File Location and compile program

- ❖Terminal : bash:
- ✓ cd ~/Desktop
- ✓ls
- ❖Now compile using GCC: bash:
- ✓gcc hello_world.c -o hello_world
- ❖Run the Program :Execute the compiled : bash:
- ✓./hello_world

Pic 5.C finish implementation of hello_worled program



```
┌─[user@parrot]─[~/Desktop]
└──╼ $cd~ /desktop
bash: cd~: command not found
┌─[✗]─[user@parrot]─[~/Desktop]
└──╼ $stdo apt update
bash: stdo: command not found
┌─[✗]─[user@parrot]─[~/Desktop]
└──╼ $sudo apt install gcc
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
gcc is already the newest version (4:12.2.0-3).
gcc set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
┌─[user@parrot]─[~/Desktop]
└──╼ $cd ~/desktop
bash: cd: /home/user/desktop: No such file or directory
┌─[✗]─[user@parrot]─[~/Desktop]
└──╼ $ls
'hello world.c'   install-debian.desktop   README.license
┌─[user@parrot]─[~/Desktop]
└──╼ $

┌─[✗]─[user@parrot]─[~/Desktop]
└──╼ $
```

RECOMMENDATIONS

- Make sure your virtual machine has at least 15 GB of disk space and 4 GB of RAM to avoid installation errors.
- Make sure you download the proper edition of Parrot OS (Security, Home, etc.) based on your needs.
- This caused issues during installation, so use simple names with no special characters or spaces.
- The process can be slow, especially on low-end systems, but it's worth the wait.
- This helps in understanding and also makes your report more complete and professional.

CONCLUSION

- Working on this project gave me hands-on experience with installing and using Parrot OS in a virtual environment.
- Throughout the process, I learned not only the technical steps but also how to troubleshoot real problems, like system requirements, partitioning errors, and ISO setup issues.
- These challenges were frustrating at times, but they helped me understand how operating systems work at a deeper level.

Conclusion Cont.......

- Parrot OS stood out as a flexible and secure platform, especially useful for cybersecurity students and professionals.
- It combines powerful security tools with general-use features, making it useful for both daily tasks and ethical hacking.
- Overall, this project was a valuable learning journey that improved my technical skills and made me more confident in working with Linux-based systems.

REFERENCES

- Parrot OS Official : https://www.parrotsec.org

  Information about different Parrot editions, features, and system requirements.

- Parrot OS Documentation : https://docs.parrotsec.org

  Step-by-step guides, installation instructions, and FAQs.

- VirtualBox Official: https://www.virtualbox.org

  For learning about virtualization and setting up VMs.

- Geeks for Geeks – System Calls in Operating Systems: https://www.geeksforgeeks.org/introduction-of-system-call/

  Easy explanation of what system calls are and how they work in Linux.