Introduction to Ubuntu

Introduction

As discussed in the previous chapter, an operating system (OS) is a system software that manages the hardware and software resources of a computer and provides a common platform to run various computer programs. It acts as an intermediary between the user and the computer hardware, enabling users to interact with the computer and run applications. In this chapter, we will learn about a very popular operating system, named Ubuntu. The word Ubuntu in Southern African philosophy means "humanity to others". With the goal of providing access to computer technology to everyone, the Ubuntu operating system was first released in 2004. Since then Ubuntu has emerged as a very popular open source, free to use operating system. In this chapter, we will learn about the basic functionalities of the Ubuntu operating system.

Some Popular Operating Systems

Before discussing the Ubuntu operating system in detail, let us have a quick overview of some popular operating systems.

microsoft Windows

microsoft Windows is a widely used operating system developed by Microsoft, known for its user-friendly graphical interface. It supports a broad range of applications, hardware, and devices, making it popular in homes, businesses, and educational institutions. Windows offers features like multitasking, file and folder management, system security, networking, and regular updates. It includes tools like File Explorer, Task Manager, and Control Panel for easy system navigation and control.

macOS

macOS is Apple's proprietary operating system for Mac computers, known for its sleek design, stability, and strong integration with Apple hardware and software. Built on a Unix-based foundation, it offers high performance, security, and reliability. The user interface is intuitive, featuring the Dock, Finder, and Mission Control for efficient navigation and multitasking. MacOS supports a wide range of applications, including productivity, creativity and development tools. MacOS provides seamless integration with other Apple devices through features like Handoff, AirDrop, and iCloud which enhances overall user experience.

Linux

Unlike Windows and macOS which are commercially available operating systems, Linux is a free, open source operating system. Following the philosophy of the Unix operating system, Linux provides high performance, security, reliability and portability. It provides a Command Line Interface (CLI) along with an impressive Graphical User Interface (GUI) which makes it very user-friendly. A major advantage of Linux is that it provides the complete freedom to customize and configure as per the user's choice. Linux also provides excellent interoperability with Microsoft Windows. We can easily access the Windows file system from the Linux environment. The office suites on Linux OpenOffice and LibreOffice provide good interoperability with the Microsoft Office suite.

Linux has many flavours, known as distributions, which are designed for different needs. RedHat and Debian are two major Linux distributions which offer great stability and are ideal for servers. Over the years many different Linux variants like Fedora, CentOS, Kali, etc. have emerged. Ubuntu is one of the fastest growing Linux distributions.

Ubuntu

The Ubuntu operating system is free to download, use, and modify, which makes it particularly appealing for students, developers, educational institutions, and startups. Ubuntu is a gateway into the Linux ecosystem and open-source world. It empowers users with a free, customizable, secure, and powerful computing environment. Ubuntu comes pre-packaged with essential software such as a web browser (Firefox), email client (Thunderbird), office suite (LibreOffice), media player, and more. Additional software can be easily installed through its Software Center or via command-line tools like Advanced Package Tool (APT).

Basic Terminology related to Ubuntu

Table 4.1 summarizes some basic terminology which we will use to discuss Ubuntu Operating system.

Term	Meaning
Ubuntu	A user-friendly Linux distribution developed by Canonical Ltd.
Kernel	The core of the operating system that manages hardware and system processes.
Terminal	A command-line interface used to execute commands.
Shell	The program that interprets and runs commands typed into the terminal (e.g., Bash).
Desktop Environment (DE)	Graphical interface including panels, windows, icons, etc. (e.g., GNOME, KDE, XFCE).
Launcher / Dock	The panel or bar that gives access to favourite applications.
Package	A bundle containing a program and metadata needed for installation.
Repository (Repo)	A server or database where Ubuntu software packages are stored.
Root	The superuser account with full administrative privileges.
Sudo	Stands for “superuser do”; allows a permitted user to run administrative commands.
Permissions	Rules that determine who can read, write, or execute a file or directory.

Table 4.1 : Basic Terminology

Working with Ubuntu Linux

Ubuntu Linux has a very user-friendly Graphical Interface and a very useful Command Line Interface through which users can run various programs, create files and manage directories. Being a secured operating system Ubuntu allows us to create a username and password through which we can protect the access to the system. At the start, Ubuntu greets with the login screen similar to the one shown in Figure 4.1.

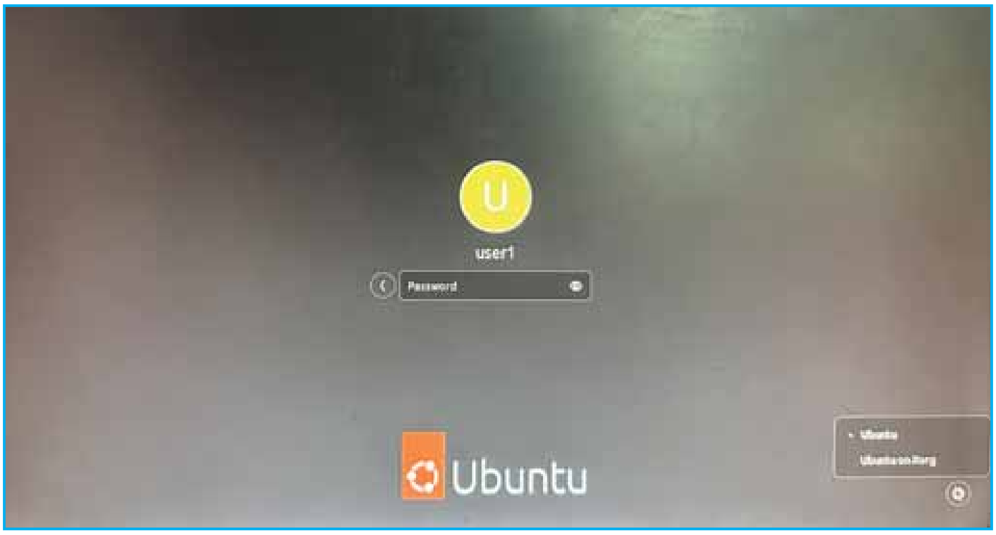


Figure 4.1 : Ubuntu Login Screen

Ubuntu User types

Ubuntu has predefined types of users who have different access privileges. The different Ubuntu user types are summarized in table 4.2. Every Ubuntu user belongs to a specific group.

User Type	Tasks
Root User (Superuser)	Username: root Permissions: Full access to all commands and files on the system. Typical Use: System administration tasks such as installing packages, managing users, changing critical system settings. Note: In Ubuntu, direct login as root is disabled by default for security reasons. Instead, users gain root privileges using the *sudo* command.
Sudo User	Permissions: Regular user with administrative privileges via sudo. Typical Use: Can perform administrative tasks by prefacing commands with sudo, e.g., *sudo apt update.*
Regular User	Permissions: Limited to their own home directory and certain allowed system-wide read access. Typical Use: Daily usage such as running applications, editing personal files, browsing the web, and compiling programs in user space. Examples: Created during installation or by administrators for other users.
System Users	Purpose: Used by system services and daemons (background processes). Characteristics: No login shell (/usr/sbin/nologin or /bin/false), typically have UID < 1000. Examples: www-data (used by web servers like Apache), mysql (used by MySQL server), daemon, syslog.
Guest User (if enabled)	Permissions: Very limited, temporary user with no access to system settings or permanent file storage. Typical Use: Temporary access for someone who needs to use the system without affecting it. Note: Disabled by default in newer Ubuntu versions due to security concerns.

Table 4.2 : Ubuntu User types

Usually, we access the Ubuntu operating system as a regular user. A regular user can create files and run application programs. To perform daily office work like making new documents, listening to music, playing games, or browsing the Internet, we need to login as a regular user.

Ubuntu Directory Structure

Once we login to the Ubuntu system we get access to the file and directories stored in the system. The overall directory structure of Ubuntu is shown in figure 4.2. Note that, the subdirectories shown in the figure 4.2 are not a comprehensive list of all Ubuntu subdirectories, but it should be sufficient to understand the overall directory structure.

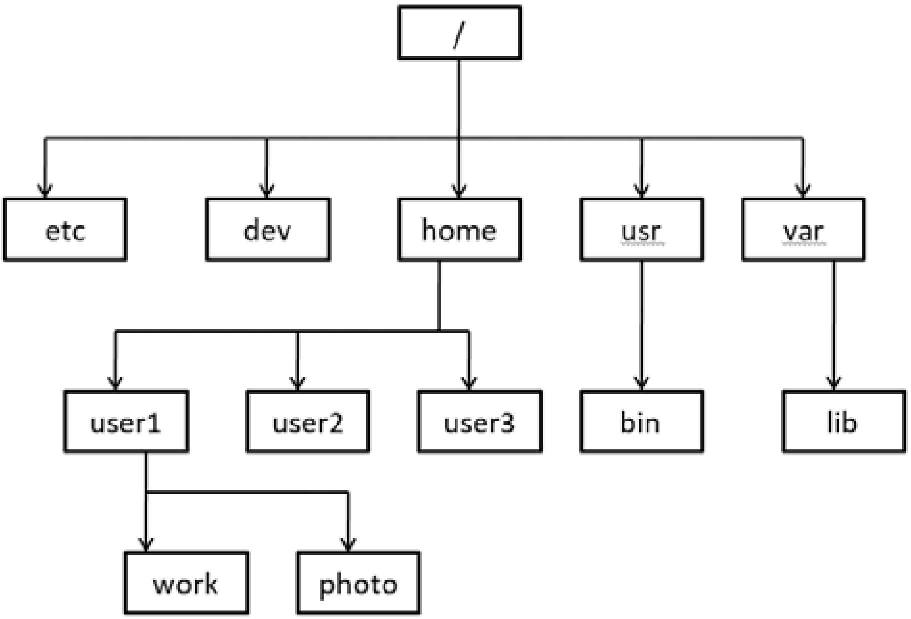


Figure 4.2 : Ubuntu Directory Structure

As shown in the figure 4.2, Ubuntu uses a hierarchical directory structure that starts from the root directory (/). Each directory under / serves a specific purpose. */home* directory stores subdirectories specific to 'regular users' of the system, e.g. user1, user2 and user3 are directories of regular users user1, user2 and user3, respectively. Table 4.3 summarizes the purpose of directories shown in Figure 4.2

Directory Name	Purpose
/etc	Contains system-wide configuration files and shell scripts. Example: ● /etc/passwd – user account information ● /etc/hostname – name of the system
/dev	Contains special device files representing hardware. Example: ● /dev/sda1 – first partition of the first hard disk ● /dev/null – null device
/home	Contains personal directories for users. Example: ● /home/user1 – home directory for user "user1" ● /home/user/Documents – personal documents

/usr	Secondary hierarchy for read-only user data and programs. Contains many subdirectories: ● /usr/bin – non-essential user binaries (e.g., firefox) ● /usr/lib – libraries ● /usr/share – shared data
/var	Contains files that change frequently (logs, spool files). Example: ● /var/log/syslog – system log ● /var/mail/username – user mailbox

Table 4.3 : Some Useful Ubuntu Directories

Ubuntu Desktop Interface

Once we successfully login to the Ubuntu operating system, its graphical interface, known as Desktop Environment (DE), provides a graphical way to access various files, directories and programs. The Ubuntu operating system allows users to choose from multiple desktop environments. GNOME and KDE are two most popular DEs available in not just Ubuntu but almost all Linux distributions. Here, we will discuss the GNOME desktop environment.

GNOME Desktop

GNOME (GNU Network Object Model Environment) is a free, open-source desktop environment used by default in Ubuntu and many other Linux distributions. Designed for simplicity and ease of use, GNOME provides a clean, modern interface with intuitive navigation, minimal distractions, and strong accessibility support. It includes a suite of core applications like Files, Terminal, Settings, and GNOME Software. Figure 4.3 shows a typical GNOME desktop screen. GNOME uses a minimalist approach, avoiding desktop clutter and emphasizing keyboard shortcuts and gestures for navigation. It supports extensions for customization, allowing users to tailor the desktop to their preferences.

The GNOME desktop environment emphasizes productivity through features like activity overview, dynamic workspaces, and keyboard shortcuts. As shown in the figure 4.3, the core GNOME Shell includes a top bar, which includes *activities*, *notification* and *system menu*. The left side panel of the interface is known as *Dash* (also referred as *Dock*) which provides access to various applications. The central area is used to display files and directories. As we can see in figure 4.3, the home directory is visible on the desktop by default. We can access the files stored in the home directory by clicking on it. We can also create other directories and files, and place them on the Desktop along with the home directory.

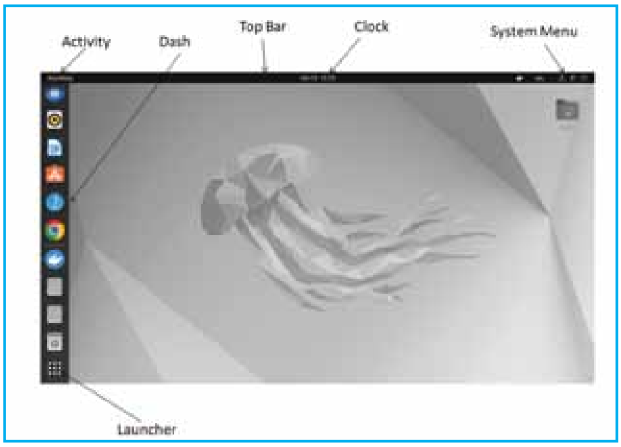


Figure 4.3 : Ubuntu GNOME Desktop

The common components of GNOME desktop are as follows,

Top Bar: As the name suggests, the top bar is located at the top of the desktop. It includes the following components,

- Activities: One of GNOME's key features is the *Activities Overview*, accessed by clicking *Activities* or pressing the Super (Windows) key. This view displays all open windows, virtual workspaces, and a search bar that allows quick access to applications, files, and settings. Figure 4.4 shows activity overview on a typical Ubuntu desktop.

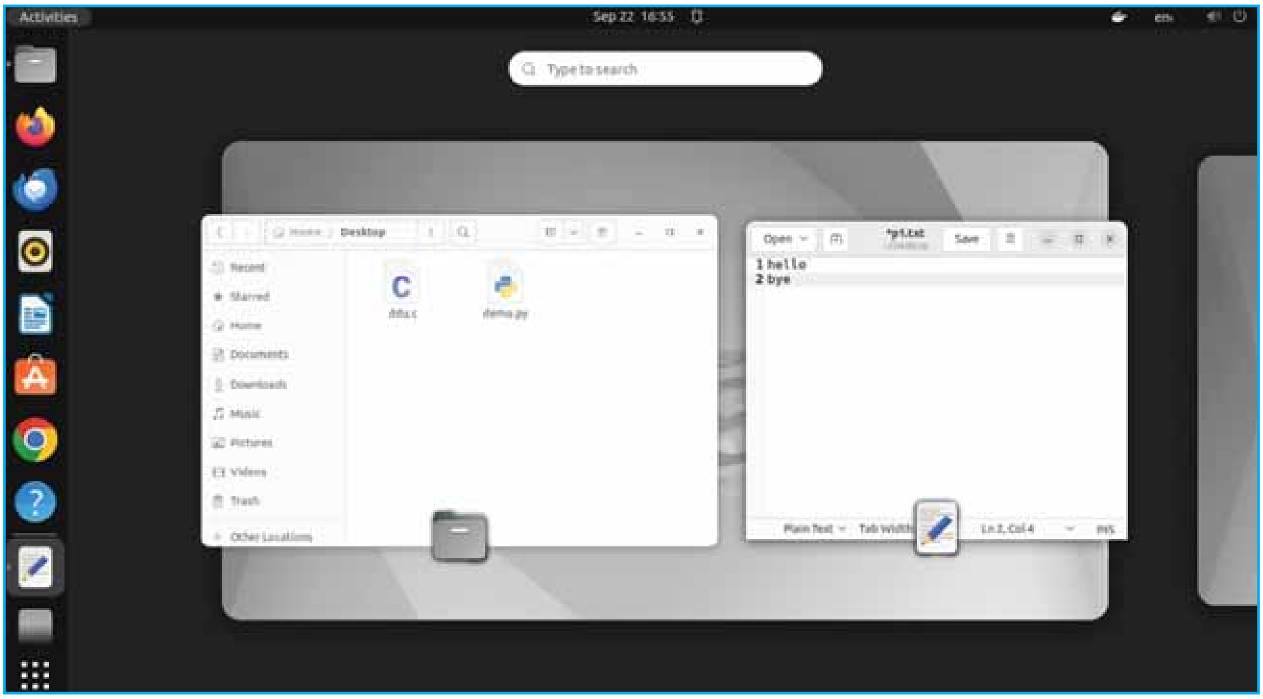


Figure 4.4 : Ubuntu Activity Overview

- Notifications: The center part of the top bar shows a clock. By clicking on the clock, we can see the current date, a month-by-month calendar, a list of upcoming appointments and new notifications. We can access the date and time settings and open the calendar application directly from the menu.

- System Menu: The top right corner of the top bar shows the system menu. By clicking on the system menu we can access some basic system settings. We can set audio, WiFi, Bluetooth etc features from the system menu. Figure 4.5 shows an expanded view of the system menu. The top part of the menu shows volume control, the battery status (in case of laptop), and buttons to launch Settings. The button allows you to suspend or power off the computer. Sliders allow you to control the sound volume or screen brightness.

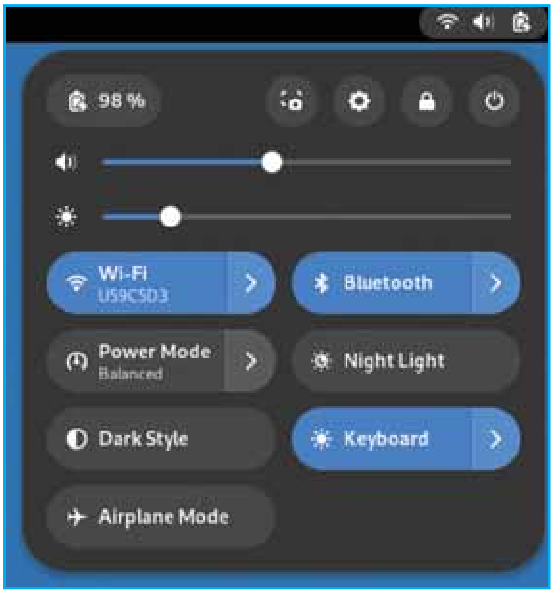


Figure 4.5 : Gnome System Menu

GNOME Dash bar: In Ubuntu, the GNOME Dash bar is reconfigured as a *dock*. As shown in the figure 4.6. the left panel of the screen shows a typical Ubuntu *dock*. It provides quick access to frequently used softwares and directories. We can click on the application icon and open it directly from the *dock*. When an application is open, the *dock* shows a small dot below its icon. A small grid icon (9 dots) located at the bottom of the *dock* is known as the *launcher*. We can click on the launcher to access all the applications, as shown in the figure 4.6.

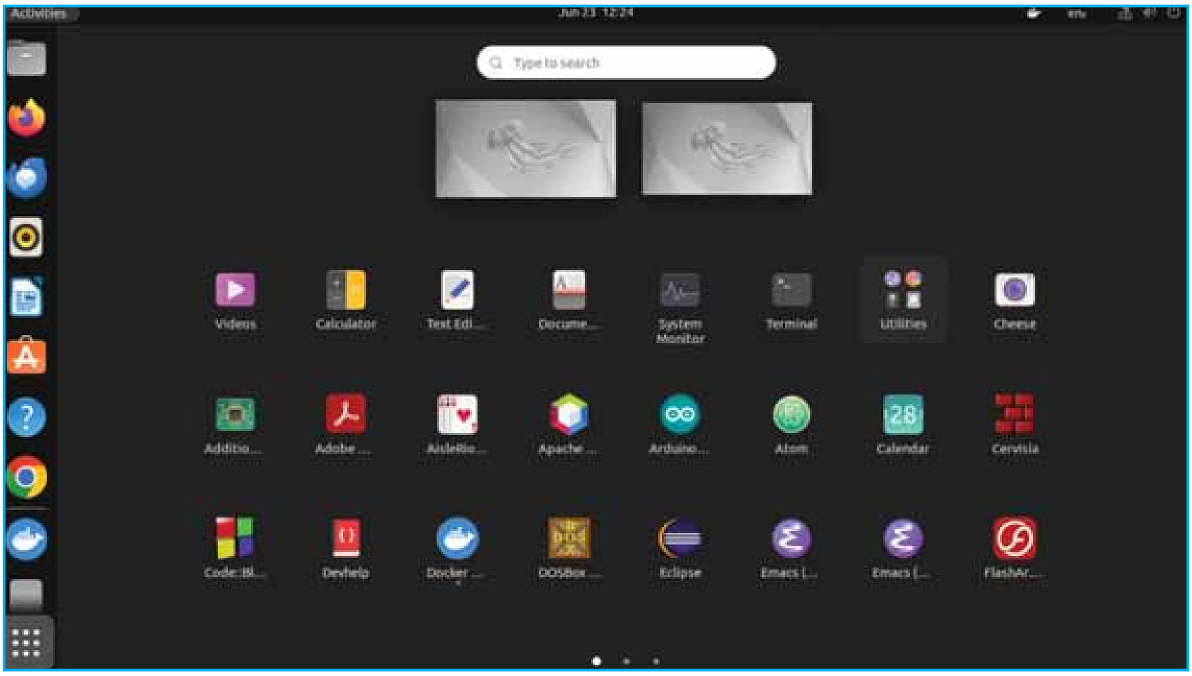


Figure 4.6 : GNOME Launcher View

File and Directory management

In Ubuntu 22.04 LTS, file and directory management is facilitated through the GNOME Files (formerly Nautilus file manager). When we click on the *Home* directory icon on the desktop or click on any of the directory icons on the dashbar, the file interface will appear as shown in Figure 4.7.

The left panel of the file manager shows some useful directories. Clicking on a directory in the left panel will show all the files and directories available under that directory. We can perform various file operations like creating, moving, renaming, and deleting files and directories through file manager. By dragging the mouse to any file or directory icon and right clicking on it will allow us to cut, copy and rename that file or directory. Right clicking of the mouse in the empty area will open the context menu through which we can create a new file.

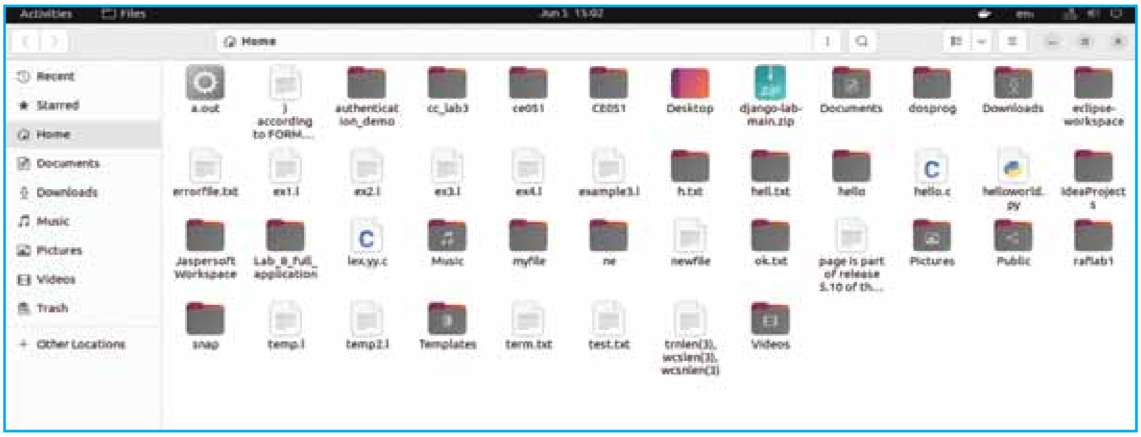


Figure 4.7 : Graphical Interface (GNOME Files)

We can open multiple instances of the GNOME files as shown in Figure 4.8. In figure 4.8, the left side interface shows */Home/Pictures* directory and the right side interface shows */Home* directory (We can read the path at the top). Now let us move a file from */Home/Pictures* directory (Left Window) to *Home* directory (Right Window). In order to do this, we will first hover the mouse over the file which we want to move (*Ubuntu1.png* in the figure). We click on the file and drag the mouse towards the right side window.

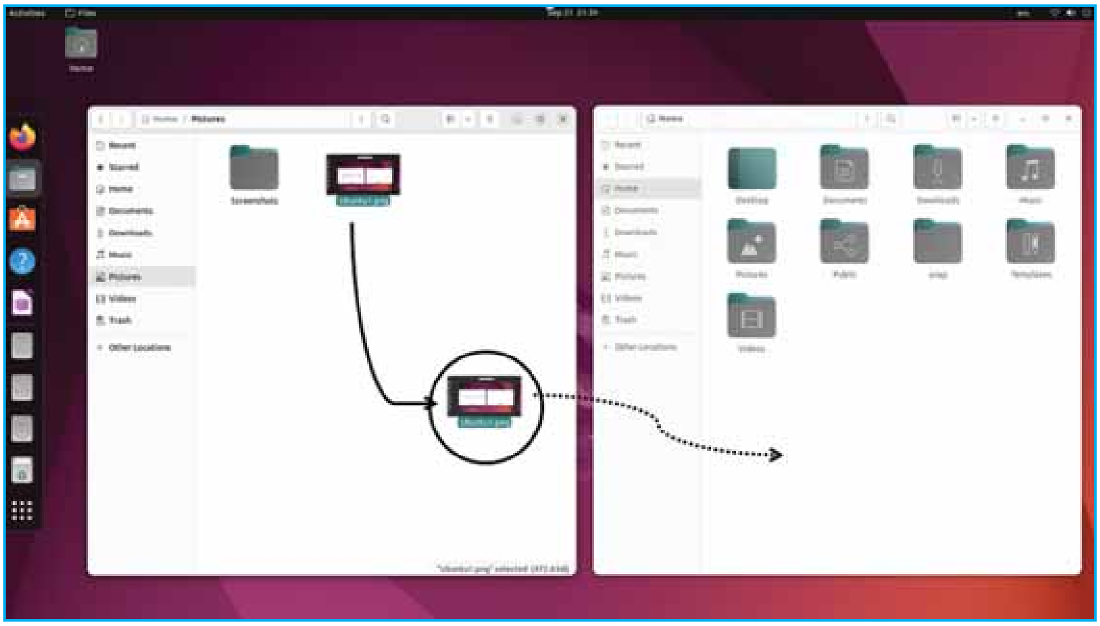


Figure 4.8 : Dragging the File from the Left Side Window

The file icon will also move along the mouse pointer. Once we reach the right side window ("Home directory") we release the mouse and the file will be moved to the "Home directory" as shown in the figure 4.9.

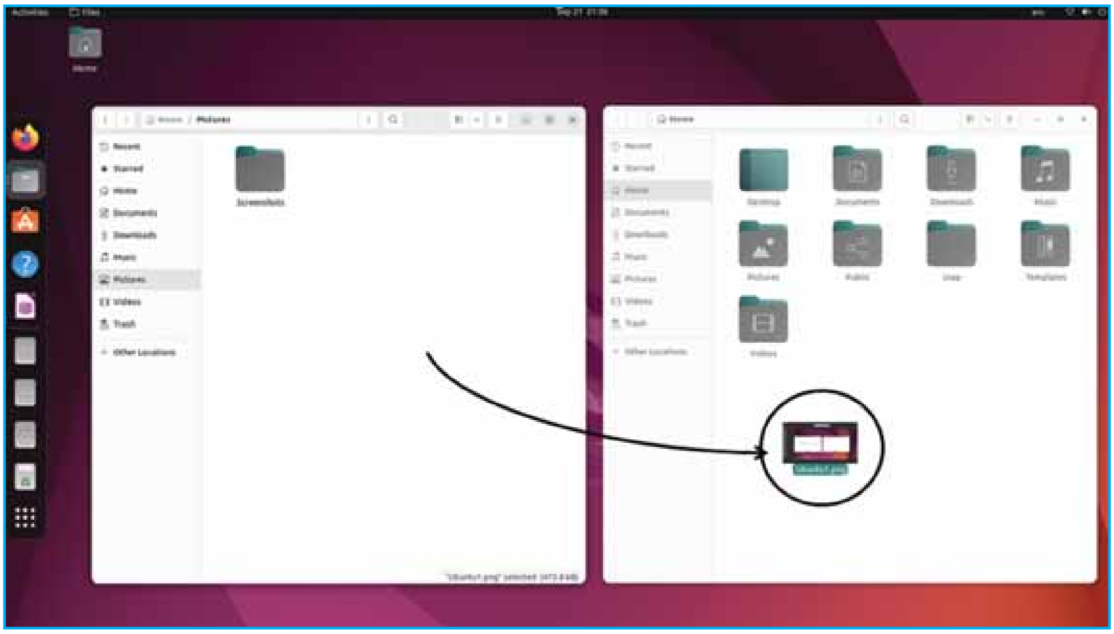


Figure 4.9 : Dropping the File in the Right Side Window

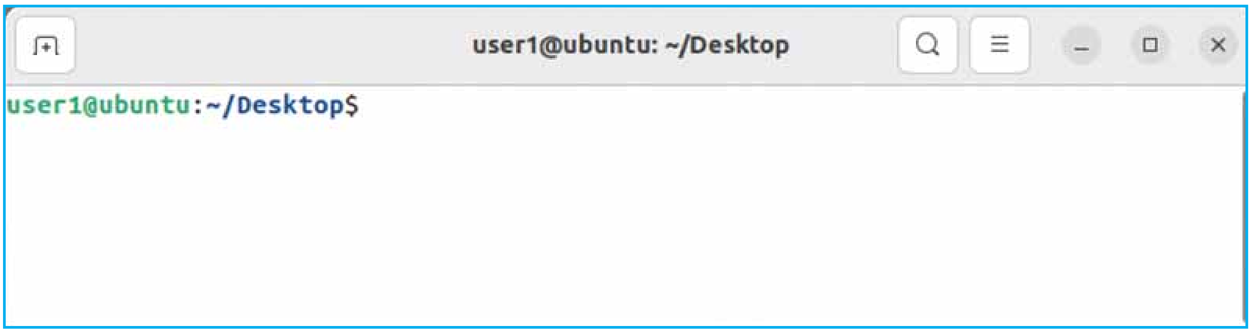
We will learn about GNOME file manager in more detail in the next chapter.

Command-Line Interface (Terminal)

We have seen how we can use Graphical Interface to perform various file operations, however the real power of Linux comes through its command line interface. Linux based operating systems are known for very powerful command line interfaces, and Ubuntu is not an exception. Almost all tasks which we can perform through Graphical Interface, can also be performed very efficiently through Command Line Interface (CLI).

The Ubuntu shell is a command line interface that allows users to interact with the operating system. It typically uses Bash (Bourne Again SHell) as the default shell. The terminal is the graphical application that provides access to the shell. Users can run commands to manage files, install software, configure the system, and automate tasks. The shell supports powerful scripting and text processing capabilities. It's essential for developers, system administrators, and advanced users. With tools like apt, grep, chmod, and more, the Ubuntu shell and terminal offer efficient control and customization beyond what graphical interfaces can provide.

We can access the shell by clicking on the terminal icon in the launcher. The CLI will appear as shown in the figure 4.10. Inside the terminal window, we see a text string ending with *$*. It is known as the prompt. (The text before the *$* indicates user name followed by system information.) We can supply shell commands after *$*. The command will be processed by the operating system and corresponding output will be generated.



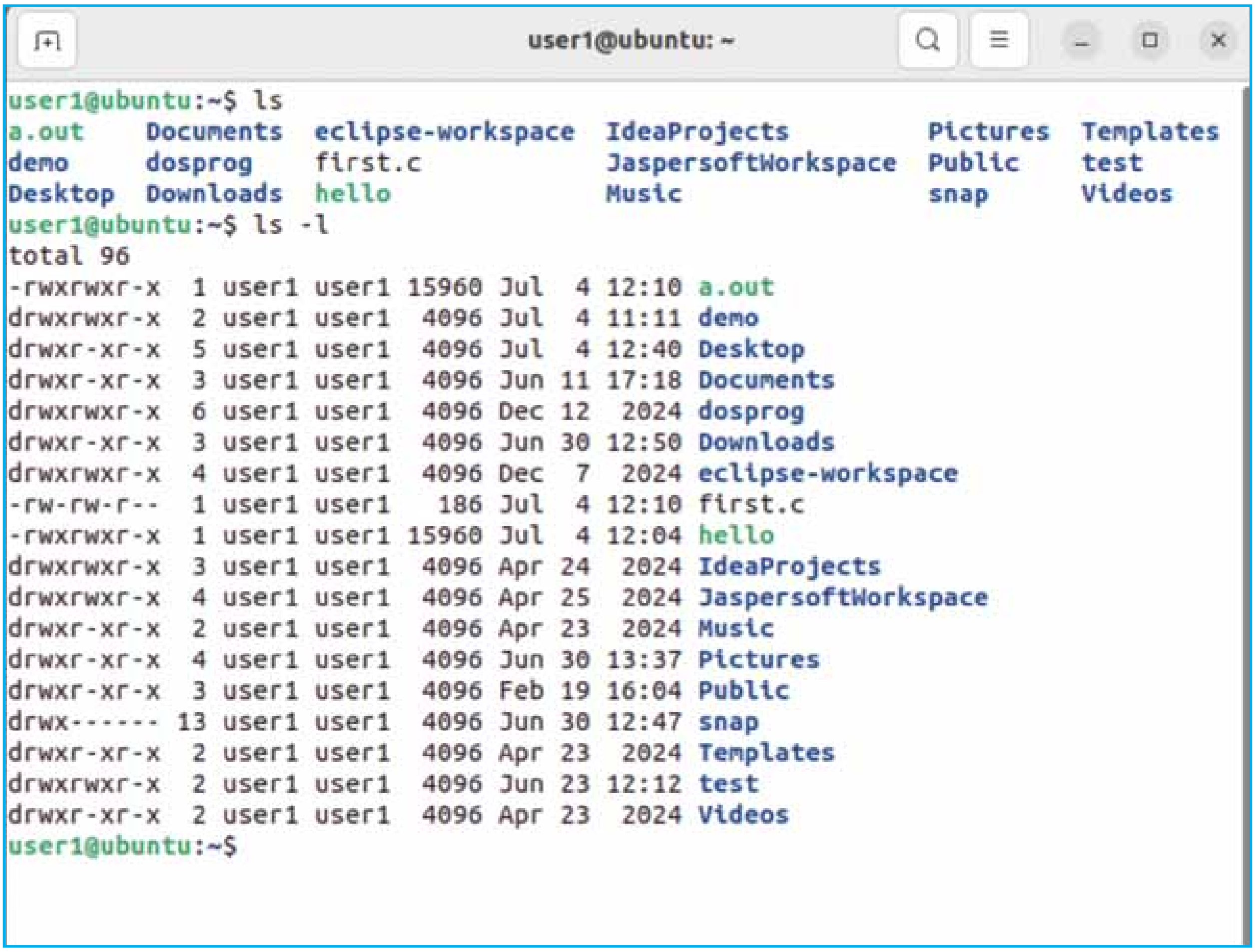
```
user1@ubuntu:~/Desktop$
```

Figure 4.10 : Ubuntu Command Line Interface (Bash terminal)

File management Commands

ls: Listing files in a directory

The *ls* command in Ubuntu lists directory contents. It displays files and folders within the specified path. Common options include *-l* for detailed info, *-a* to show hidden files, and *-h* for human-readable sizes. It helps users navigate and manage files efficiently from the command line interface. Figure 4.11 shows the output of *ls* and *ls -l* commands. The *ls* command simply lists the file names, while *ls -l* command provides detailed information about the file.



```
user1@ubuntu:~$ ls
a.out    Documents  eclipse-workspace  IdeaProjects         Pictures  Templates
demo     dosprog    first.c            JaspersoftWorkspace  Public    test
Desktop  Downloads  hello              Music                snap      Videos
user1@ubuntu:~$ ls -l
total 96
-rwxrwxr-x  1 user1 user1 15960 Jul  4 12:10 a.out
drwxrwxr-x  2 user1 user1  4096 Jul  4 11:11 demo
drwxr-xr-x  5 user1 user1  4096 Jul  4 12:40 Desktop
drwxr-xr-x  3 user1 user1  4096 Jun 11 17:18 Documents
drwxrwxr-x  6 user1 user1  4096 Dec 12  2024 dosprog
drwxr-xr-x  3 user1 user1  4096 Jun 30 12:50 Downloads
drwxrwxr-x  4 user1 user1  4096 Dec  7  2024 eclipse-workspace
-rw-rw-r--  1 user1 user1   186 Jul  4 12:10 first.c
-rwxrwxr-x  1 user1 user1 15960 Jul  4 12:04 hello
drwxrwxr-x  3 user1 user1  4096 Apr 24  2024 IdeaProjects
drwxrwxr-x  4 user1 user1  4096 Apr 25  2024 JaspersoftWorkspace
drwxr-xr-x  2 user1 user1  4096 Apr 23  2024 Music
drwxr-xr-x  4 user1 user1  4096 Jun 30 13:37 Pictures
drwxr-xr-x  3 user1 user1  4096 Feb 19 16:04 Public
drwx------ 13 user1 user1  4096 Jun 30 12:47 snap
drwxr-xr-x  2 user1 user1  4096 Apr 23  2024 Templates
drwxrwxr-x  2 user1 user1  4096 Jun 23 12:12 test
drwxr-xr-x  2 user1 user1  4096 Apr 23  2024 Videos
user1@ubuntu:~$
```

Figure 4.11 : ls and ls -l Command Output

The output structure of *ls -l* command is structured as shown in the table 4.4.

File Type and Permission	Number of hardlinks to the file	Owner Name	Group Name	File size	Date of modification	Name of the file
-rwxr-xr--	1	user	group	1234	Jun 9 12:00	filename

Table 4.4 : ls -l Command Output Structure

File type and permission is a 10 letter string. The first letter can be either '-' or 'd', d indicates directory and '-' indicates file. The next nine letters are 3 groups of 3 letters each, 'rwx'. Here, r = read, w=write, and x= execute. It indicates file permission. The first three letters 'rwx' indicate permission for the user, i.e. the user can read, write and execute the file. The second three letters 'r-x' indicate that the group can read and execute the file, but cannot write as 'w' is absent. The last three letters 'r - -' indicate that users other than the owner can read the file but cannot write or execute the file.

Number of hardlinks to the file indicates how many copies of the file exist on the hard disk. For every file created on the Ubuntu system there is a defined Owner and Group. The user who creates the file is the default owner of the file. The group which the owner belongs to, is the group of the file.

File size indicates how many bytes the file occupies on the hard disk.

Date of modification indicates the last modified date of the file.

touch: Creating a new file

The *touch* command in Ubuntu is used to create a new empty file or update the access and modification timestamps of existing files without changing their content on the terminal. We can type the following command to create a text file named *example.txt.*

```
$touch example.txt
```

mkdir: Creating a new directory

The *mkdir* command creates new directories. Users can specify the name and optional path, like *mkdir new_folder*. To create parent directories automatically, use *mkdir -p path/to/folder*. It's useful for organizing files and structuring projects within the filesystem directly from the command line.

For example the following command will create a new directory named *Projects* under the current directory,

```
$mkdir Projects
```

The following command will create a new directory named *Project1* under the parent directory *Projects*.

```
$mkdir -p Projects/Project1
```

The mkdir command is useful for organizing files and setting up folder structures.

cd: Changing the working directory

The *cd* (change directory) command in Ubuntu is used to navigate between directories in the terminal. It allows users to move to a different directory to access or manage files. For example,

```
$cd Documents
```

changes the current directory to the *Documents* folder within the user's home directory.

To move back one level, we can use,

```
$cd ..
```

and to return to the home directory, use

```
$cd ~
```

We can directly go to a specific directory (etc in the below example) by giving the following command.

```
$cd /etc
```

rm: Removing (Deleting) a file or directory

The *rm* command in Ubuntu is used to delete files and directories from the file system. For example, the following command will delete the file named abc.txt

```
        $rm abc.txt
```

We can delete multiple files by writing their names sequentially. For example, the following command will delete two files named *file1.txt* and *file2.txt*

```
        $rm file1.txt file2.txt
```

We can also delete a directory and all files and subdirectories under it as shown in the following command.

```
        $rm -r mydir
```

In the above command, the option *'-r'* indicates that all subdirectories and files under the directory 'mydir' will be deleted.

cp: Copy file from one location to another

The *cp* command is used to copy files and directories. It allows users to duplicate content from one location to another. For example, the following command copies a file named *file.txt* to a new file named *backup.txt*.

```
        $cp file.txt backup.txt
```

To copy a file to a particular directory, we can supply the following command.

```
        $cp file.txt /home/user/Documents/
```

The above command will create a copy of the file named *file.txt* under the directory */home/user/Documents/*

We can also copy the content of a directory to another directory by supplying the following command.

```
        $cp -r folder1 folder2
```

mv: Moving a file from one location to another

The *mv* command is used to move or rename files and directories. For example, we can move a file name *abc.txt* to the directory */home/user/Documents/* by supplying the following command,

```
        $mv file.txt /home/user/Documents/
```

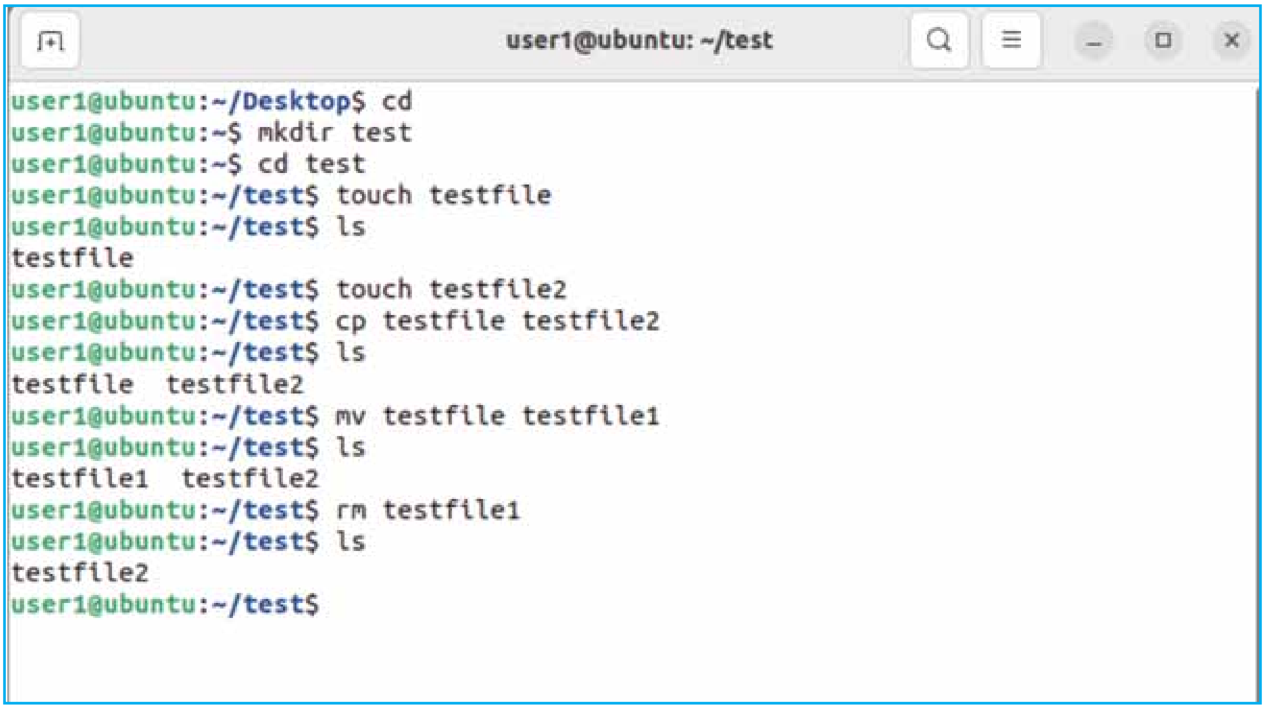
Here note an important distinction between the *mv* and *cp* commands. As the name suggests, *cp* command creates a new copy of the file with the same content. On the other hand, the *mv* command moves the file from one directory to another. In other words, *cp* command is used to create multiple copies of the same file, while mv command is used to move files from one directory to another without creating multiple copies.

We can also move and rename a directory as shown in the following command.

```
        $mv myfolder /home/user/Projects/newfolder
```

The above command will move the directory named *myfolder* to the directory */home/user/Projects* and rename it to *newfolder*.

Figure 4.12 shows output of the above commands on a linux terminal.



```
user1@ubuntu:~/Desktop$ cd
user1@ubuntu:~$ mkdir test
user1@ubuntu:~$ cd test
user1@ubuntu:~/test$ touch testfile
user1@ubuntu:~/test$ ls
testfile
user1@ubuntu:~/test$ touch testfile2
user1@ubuntu:~/test$ cp testfile testfile2
user1@ubuntu:~/test$ ls
testfile  testfile2
user1@ubuntu:~/test$ mv testfile testfile1
user1@ubuntu:~/test$ ls
testfile1  testfile2
user1@ubuntu:~/test$ rm testfile1
user1@ubuntu:~/test$ ls
testfile2
user1@ubuntu:~/test$
```

Figure 4.12 : File Management Commands on Linux Terminal

Apart from the above file management commands, there are some other useful commands which will be useful in day to day working. These commands are summarized in table 4.5.

Comand	Use
man	opens the manual for a command. Example: *$man ls*
who	displays information about users who are currently logged in to the system Example: *$who*
date	display and set the system date and time. Example: *$date*
pwd	Displays the current working directory. Example: *$/home/user1*
cat	Displays the contents of a file. Example: *$cat file.txt*
chown	Changes the owner and/or group of a file or directory. Example: *$chown user:group filename*
chmod	Modifies the permissions (read, write, execute) for user, group, and others on a file or directory. Example: *$chmod 755 filename*
chgrp	Changes the group ownership of a file or directory. Example: *$chgrp groupname filename*

Table 4.5 : Useful Ubuntu Linux Commands

Summary

Ubuntu is a freely available operating system compatible with most of the modern day processors. Ubuntu is derived from Debian Linux and is widely used amongst students and researchers. Ubuntu provides a powerful command line interface along with Graphical user interface, which makes it very user friendly and provides full control to the end user. Ubuntu comes with a very powerful User Interface, GNOME, through which we can access various applications, create files and manage directories. Ubuntu also has a very strong Command Line Interface. The Ubuntu terminal provides access to bash shell, through which we can manage files and directories by applying commands.

EXERCISE

1. Explain the functionalities of Ubuntu File manager.

2. Compare Ubuntu with Windows operating system.

3. What makes Ubuntu a useful operating system for students and researchers?

4. Which functionalities can be accessed through the top bar in GNOME desktop IDE?

5. What is Shell? Which is the default shell available with Ubuntu?

6. Which Linux flavour is Ubuntu based on?

7. What is the use of ls command?

8. What is the role of root user in Ubuntu?

9. Which are the different types of users in Ubuntu?

10. Explain different ways to create a new directory in Ubuntu.

11. State whether true or false.

(1) Windows operating system is freely available.

(2) Ubuntu operating system is based on MacOS.

(3) Sudo command is used to perform administrative tasks in Ubuntu.

(4) Touch command is used to create a new file.

(5) Ubuntu does not allow root user login.

12. Fill-in the blanks.

(1) Ubuntu is ____________ and __________ operating system.

(2) Ubuntu uses __________ kernel.

(3) The panel which gives quick access to frequently used applications is called ________

(4) A server database where Ubuntu packages are stored is called _________.

(5) Every Ubuntu user belongs to a _________.

13. Multi-choice questions. Choose the most correct answer.

(1) Who is the developer of Ubuntu?

(a) RedHat (b) Fedora

(c) Canonical (d) Microsoft

(2) Which command is used to list files in Ubuntu?

(a) List (b) Show (c) ls (d) dir

(3) Which desktop environment does Ubuntu use by default?

(a) KDE (b) LXDE (c) XFCE (d) GNOME

(4) Which Ubuntu command is used to change file permissions?

(a) chmod (b) Chown (c) Perm (d) access

(5) Which command is used to copy files in Ubuntu?

(a) mv (b) cp (c) rm (d) copy

(6) What does the sudo command do in Ubuntu?

(a) Runs commands as a guest user (b) Saves user data

(c) Runs commands with root privileges (d) Shuts down the system

(7) What is GNOME primarily used for in Ubuntu?

(a) Server management (b) Desktop interface

(c) File compression (d) Package installation

(8) What does the rm command do?

(a) Removes a file or directory (b) Renames a file

(c) Creates a new file (d) Reads a file

(9) What does the touch command do?

(a) Displays file contents (b) Deletes a file

(c) Creates an empty file (d) Copies a file

(10) Which user can perform administrative tasks in Ubuntu?

(a) Root (b) User

(c) System (d) Guest

Laboratory Exercise

1. Perform the following file operations through GNOME Desktop UI

a. Create a new text file with name test.txt and store it into the home directory.

b. Move the file from the home directory to the document directory.

2. Perform the following tasks using Ubuntu Shell commands

a. Create a new directory

b. Create a new file named 'abc’

c. Check permissions of the file

d. Rename the file 'abc' to 'xyz’

3. Perform the following tasks using Ubuntu Shell commands

a. Create a file named 'test.txt’

b. Create a copy of the 'test.txt' file under the directory 'Documents’

c. Create a new directory 'Project’

d. Move the file 'test.txt' under the directory 'Project’