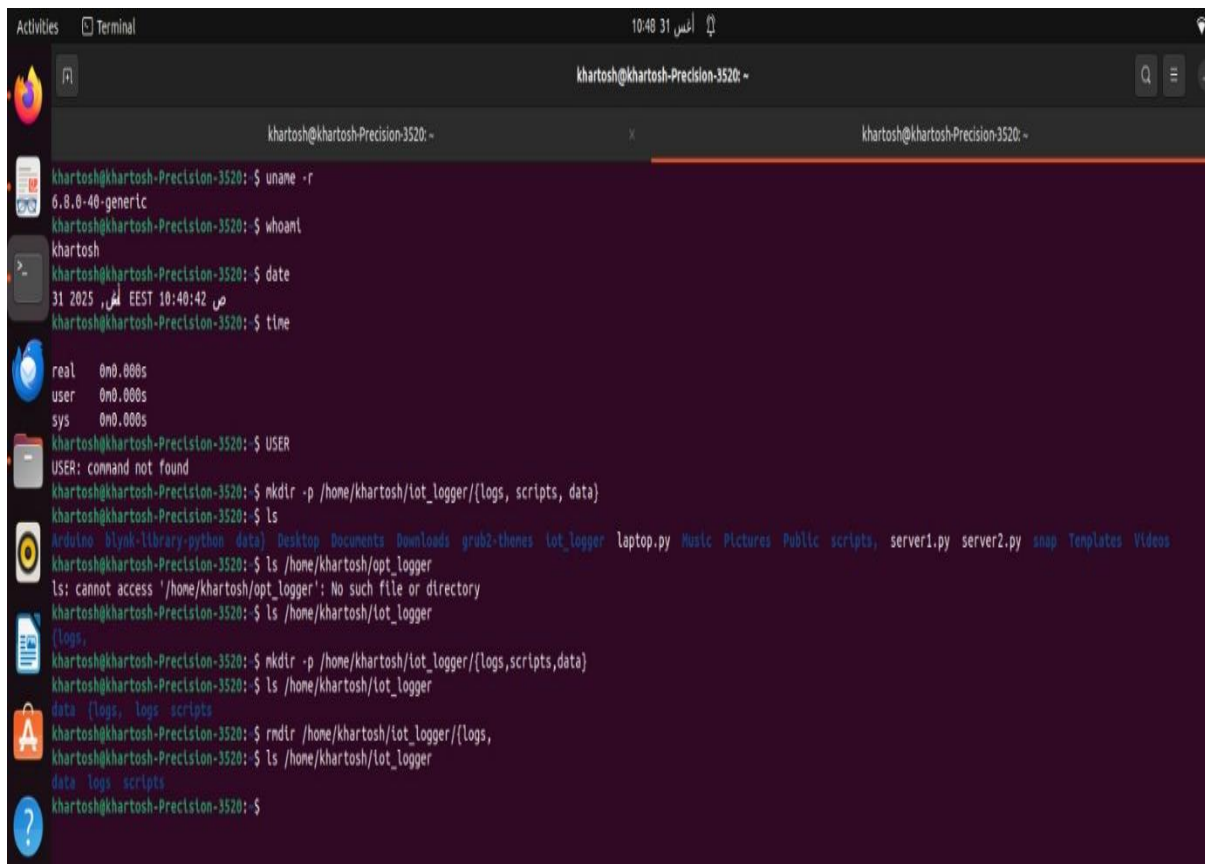


Tasks

```
150 sudo apt update
151 sudo apt upgrade
152 hstory
```

Figure 1 Tasks_1



The screenshot shows a terminal window titled "Terminal" with the user "kheartosh" on a machine named "kheartosh-Precision-3520". The terminal displays the following commands and their outputs:

```
kheartosh@kheartosh-Precision-3520:~$ uname -r
6.8.0-40-generic
kheartosh@kheartosh-Precision-3520:~$ whoami
kheartosh
kheartosh@kheartosh-Precision-3520:~$ date
31 2025 10:40:42 EEST
kheartosh@kheartosh-Precision-3520:~$ time
real    0m0.000s
user    0m0.000s
sys     0m0.000s
kheartosh@kheartosh-Precision-3520:~$ USER
USER: command not found
kheartosh@kheartosh-Precision-3520:~$ mkdir -p /home/kheartosh/iot_logger/{logs, scripts, data}
kheartosh@kheartosh-Precision-3520:~$ ls
Arduino  Blynk-library-python  data  Desktop  Documents  Downloads  grub2-themes  iot_logger  laptop.py  Music  Pictures  Public  scripts  server1.py  server2.py  snap  Templates  Videos
kheartosh@kheartosh-Precision-3520:~$ ls /home/kheartosh/opt_logger
ls: cannot access '/home/kheartosh/opt_logger': No such file or directory
kheartosh@kheartosh-Precision-3520:~$ ls /home/kheartosh/iot_logger
{logs,
kheartosh@kheartosh-Precision-3520:~$ mkdir -p /home/kheartosh/iot_logger/{logs,scripts,data}
kheartosh@kheartosh-Precision-3520:~$ ls /home/kheartosh/iot_logger
data {logs, logs scripts
kheartosh@kheartosh-Precision-3520:~$ rmdir /home/kheartosh/iot_logger/{logs,
kheartosh@kheartosh-Precision-3520:~$ ls /home/kheartosh/iot_logger
data logs scripts
kheartosh@kheartosh-Precision-3520:~$
```

Figure 2 Tasks_2,3

Open-Ended Questions

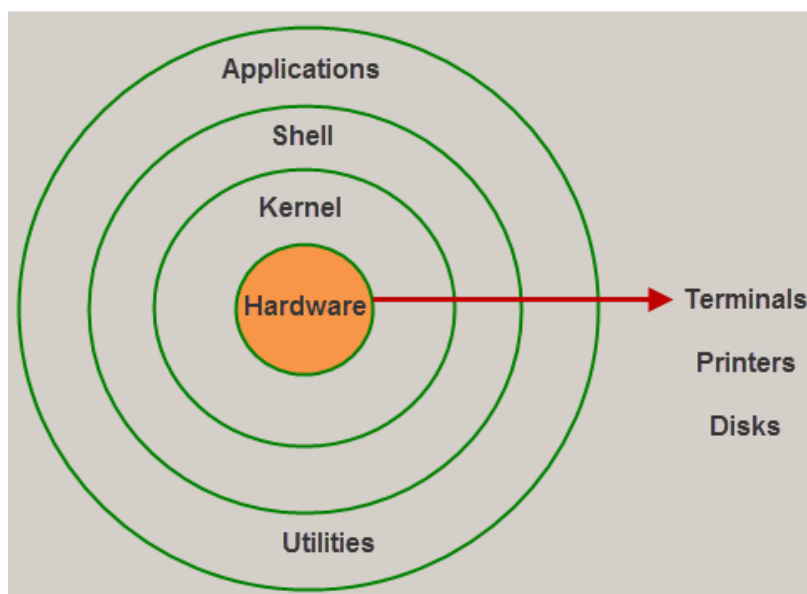
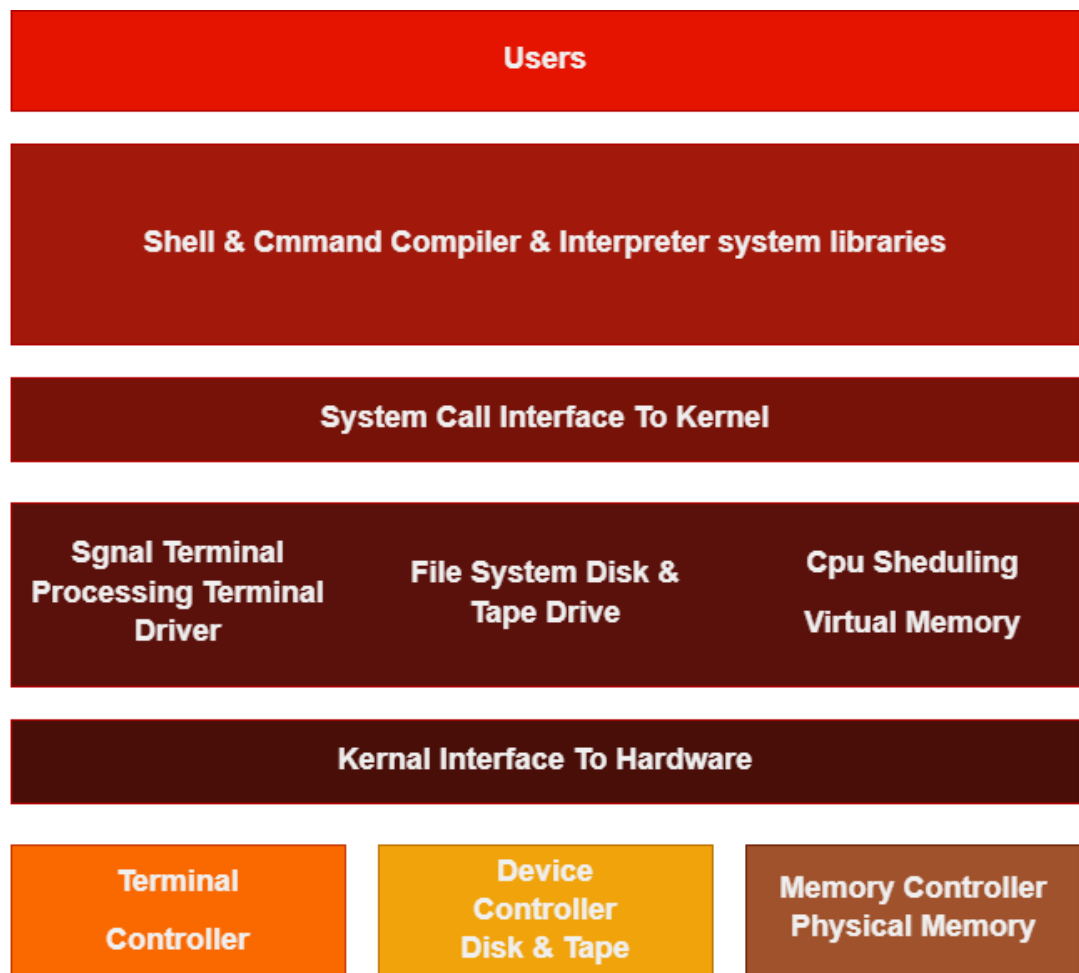


Figure 3 the Linux architecture layers

/, /bin, /sbin, /usr, /etc, /var

/

Root directory, it refers for the base of the file system and the foundation of the Linux file system.

The Linux file system follows a tree-like hierarchy, starting from a single root directory, denoted by a forward slash. All other directories and files branch out from this root

/bin

The */bin* directory contains binaries that are essential to the operating system.

This is used for trivial binaries used in the very early boot stage or ones that you need to have available in booting single-user mode. Think of binaries like cat, ls, etc.

/sbin

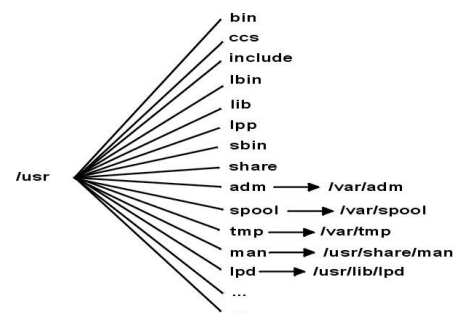
For binaries with superuser (root) privileges required

It is like bin, but it specifically for administration commands.

/usr

The */usr* file system contains executable files that can be shared among machines.

The major subdirectories of the */usr* directory are shown in the following diagram.



/etc

The */etc* directory in Linux is a critical system directory that contains configuration files and directories for the system and installed applications.

Its primary purpose is to store system-wide configuration settings that control the behavior of the operating system and various services.

/etc/passwd

/etc/fstab

/etc/hosts

/etc/network/interfaces

/var

/var is a standard directory that stands for "variable files"

This directory contains data that changes frequently while the system is running.

This directory is used to store data such as:

System Logs-Spool files-Cache from proxy servers-Databases-Temporary files required by programs

```
$ tree /var -L 1
```

/var

├─ account

├─ adm

├─ cache

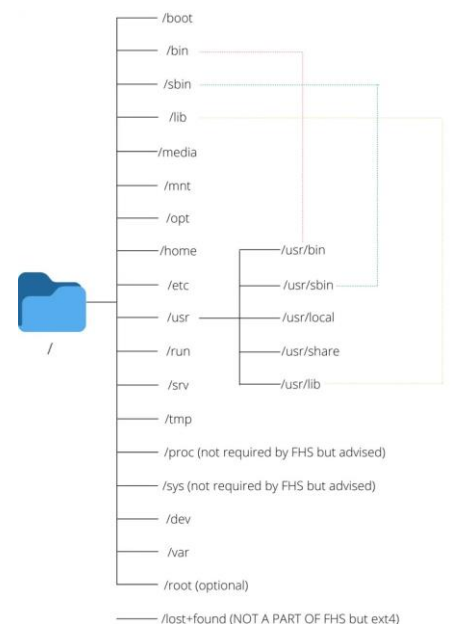
├─ db

├─ lib

├─ local

└─ lock -> ../run/lock

.....



Why does Linux treat everything as a file?

This statement is true because there are special files that are more than just files (named pipes and sockets, for instance), but to keep things simple, saying that everything is a file is an acceptable generalization. A Linux system, just like UNIX, makes no difference between a file and a directory, since a directory is just a file containing names of other files. Programs, services, texts, images, and so forth, are all files. Input and output devices, and generally all devices, are files, according to the system.

When you create a file or transfer a file to your system, it occupies some space on the physical disk and it is considered to be in a specific format (file type).

Explain the difference between a program and a process?

In Computer Science, there are two fundamental terms in operating system: Program and Process. Program is a set of instructions written to perform a task, stored in memory. A process is the active execution of a program, using system resources like CPU and memory.

| Program | Process |
|---|--|
| The program contains a set of instructions designed to complete a specific task. | The process is an instance of an executing program. |
| A program is a passive entity as it resides in the secondary memory. | The process is an active entity as it is created during execution and loaded into the main memory. |
| Program exists at a single place and continues to exist until it is deleted. | The process exists for a limited period as it gets terminated after the completion of the task. |
| A program is a static entity. | The process is a dynamic entity. |
| The program does not have any resource requirement, it only requires memory space for storing the instructions. | The process has a high resource requirement, it needs resources like CPU, memory address, and I/O during its lifetime. |
| The program does not have any control block. | Process has its control block called Process Control Block. |
| The program has two logical components: code and data. | In addition to program data, a process also requires additional information required for the management and execution. |
| Program contains instructions | The process is a sequence of instruction execution. |