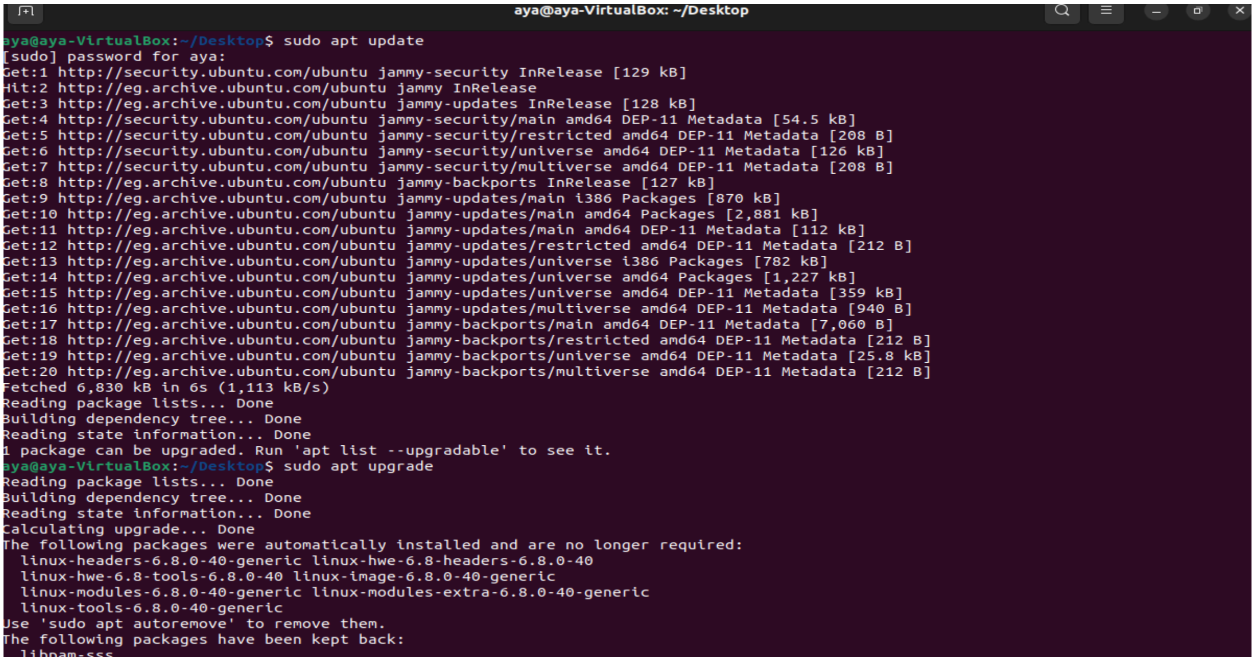
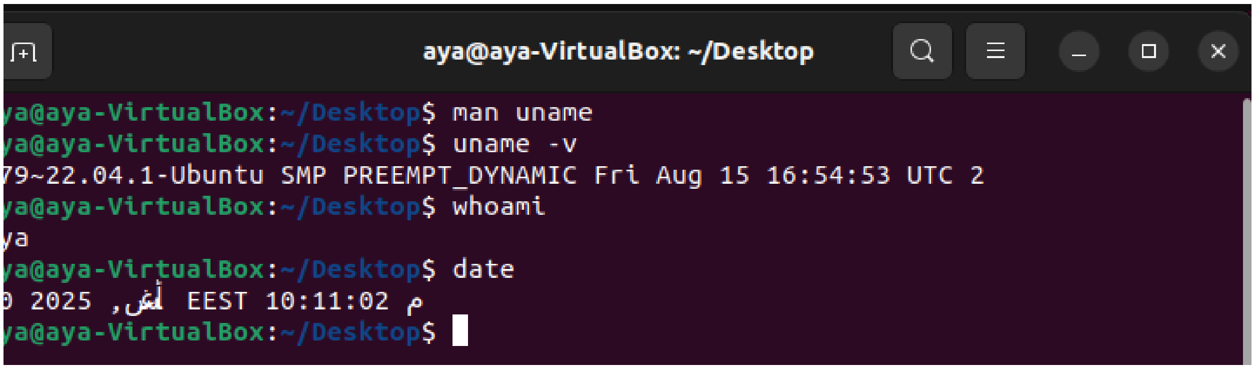
Day 1 - System Update & Directory Setup

1)



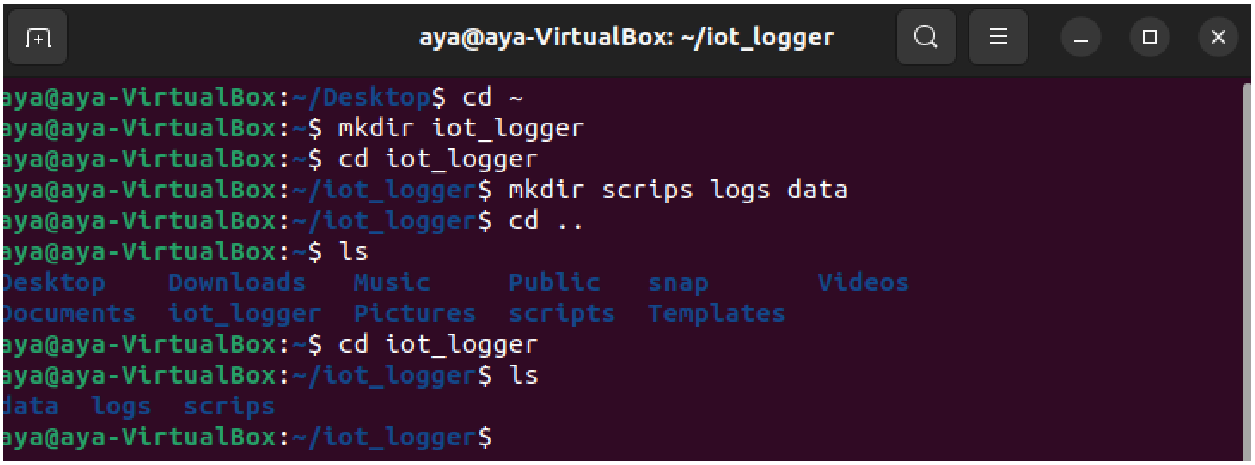
```
aya@aya-VirtualBox:~/Desktop$ sudo apt update
[sudo] password for aya:
Get:1 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Hit:2 http://eg.archive.ubuntu.com/ubuntu jammy InRelease
Get:3 http://eg.archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security/main amd64 DEP-11 Metadata [54.5 kB]
Get:5 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 DEP-11 Metadata [208 B]
Get:6 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 DEP-11 Metadata [126 kB]
Get:7 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 DEP-11 Metadata [208 B]
Get:8 http://eg.archive.ubuntu.com/ubuntu jammy-backports InRelease [127 kB]
Get:9 http://eg.archive.ubuntu.com/ubuntu jammy-updates/main i386 Packages [870 kB]
Get:10 http://eg.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [2,881 kB]
Get:11 http://eg.archive.ubuntu.com/ubuntu jammy-updates/main amd64 DEP-11 Metadata [112 kB]
Get:12 http://eg.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 DEP-11 Metadata [212 B]
Get:13 http://eg.archive.ubuntu.com/ubuntu jammy-updates/universe i386 Packages [782 kB]
Get:14 http://eg.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1,227 kB]
Get:15 http://eg.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 DEP-11 Metadata [359 kB]
Get:16 http://eg.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 DEP-11 Metadata [940 B]
Get:17 http://eg.archive.ubuntu.com/ubuntu jammy-backports/main amd64 DEP-11 Metadata [7,060 B]
Get:18 http://eg.archive.ubuntu.com/ubuntu jammy-backports/restricted amd64 DEP-11 Metadata [212 B]
Get:19 http://eg.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 DEP-11 Metadata [25.8 kB]
Get:20 http://eg.archive.ubuntu.com/ubuntu jammy-backports/multiverse amd64 DEP-11 Metadata [212 B]
Fetched 6,830 kB in 6s (1,113 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
1 package can be upgraded. Run 'apt list --upgradable' to see it.
aya@aya-VirtualBox:~/Desktop$ sudo apt upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following packages were automatically installed and are no longer required:
  linux-headers-6.8.0-40-generic linux-hwe-6.8-headers-6.8.0-40
  linux-hwe-6.8-tools-6.8.0-40 linux-image-6.8.0-40-generic
  linux-modules-6.8.0-40-generic linux-modules-extra-6.8.0-40-generic
  linux-tools-6.8.0-40-generic
Use 'sudo apt autoremove' to remove them.
The following packages have been kept back:
  libpam-sss
```

2)



```
ya@aya-VirtualBox:~/Desktop$ man uname
ya@aya-VirtualBox:~/Desktop$ uname -v
79~22.04.1-Ubuntu SMP PREEMPT_DYNAMIC Fri Aug 15 16:54:53 UTC 2
ya@aya-VirtualBox:~/Desktop$ whoami
ya
ya@aya-VirtualBox:~/Desktop$ date
0 2025 ,أغس EEST 10:11:02 م
ya@aya-VirtualBox:~/Desktop$ 
```

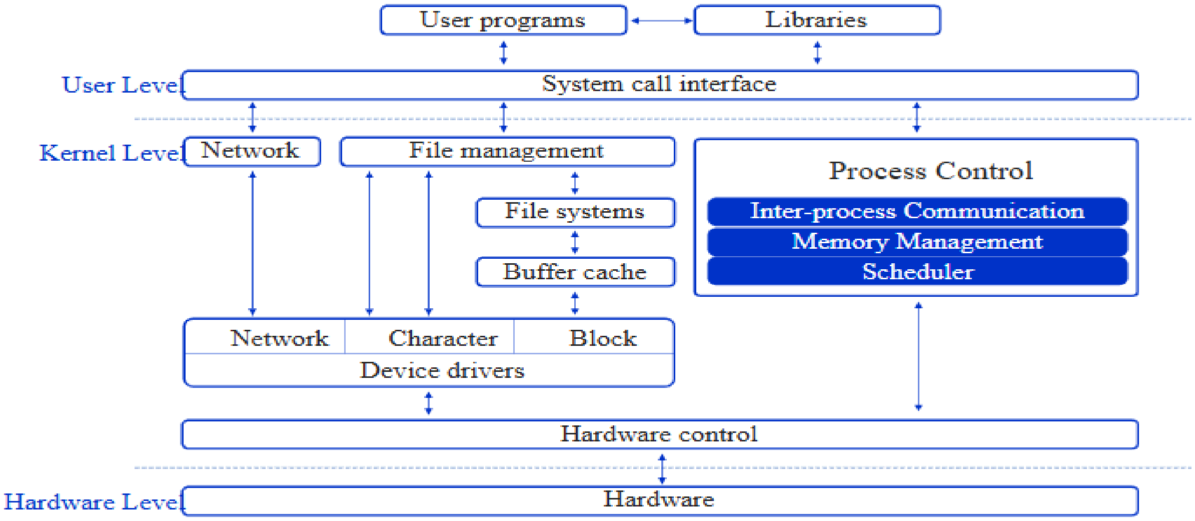
3)



```
aya@aya-VirtualBox:~/Desktop$ cd ~
aya@aya-VirtualBox:~$ mkdir iot_logger
aya@aya-VirtualBox:~$ cd iot_logger
aya@aya-VirtualBox:~/iot_logger$ mkdir scrips logs data
aya@aya-VirtualBox:~/iot_logger$ cd ..
aya@aya-VirtualBox:~$ ls
Desktop    Downloads   Music     Public   snap       Videos
Documents  iot_logger  Pictures  scripts  Templates
aya@aya-VirtualBox:~$ cd iot_logger
aya@aya-VirtualBox:~/iot_logger$ ls
data  logs  scrips
aya@aya-VirtualBox:~/iot_logger$
```

Open_ended Questions

1) This is the structure of linux where there are three main parts: shell, kernel, and hardware. For the shell, it is (the “قشره”) where the user writes commands to interface with the kernel either through libraries then system calls or system calls directly. Regarding the kernel, it is the layer that translates the commands and deals with the hardware (e.g. manage signals and processes). It consists mainly of three management systems: network, process control, and file management in addition to that it communicates with the hardware through the ABI (application binary interface). The system call (c functions like fork()) interface it tells the kernel what to do and is able to use kernel features.



OS Structure

2) The purpose of “/” is the root directory that contains all the other important subdirectories such as bin, etc, and usr. Additionally, it is always where all the absolute paths start from since it is the top-level directory.
“/bin” contains the binary commands (programs) that are needed for the basic system operations such as ls, rm, and cp.
“/sbin” it is the system binaries which is the essential system administration binaries (not the user binaries as the /bin) used by the root (superuser) for system management such as ifconfig (prints the network interfaces)
“/usr” does not stand for user; however, it is Unix System Resources that has the installed programs (but not the basic ones also not the programs and data that are required for booting).
“/etc” contains the system configuration files such as /etc/passwd having the user accounts (it is small text files that can be modified by administrators).
“/var” has the variables (the data that changes frequently) such as /var/log for system logs and /var/cache for the cached package files.

3) Linux treats everything as a file (we do not need to write any file extension; it is just for the user, other than it does not differ) because it simplifies the system design, unifies access methods (using the same file interface), and provides the users and developers with a consistent, flexible way to interact with both software and hardware (abstraction from hardware).

   The program is any program (before running it) once it runs it becomes called a process such as if we have a python script saved it will be considered as a program, and when we just click run, it automatically will be a process that has a process ID (PID).