

Day 1 – Phase 1: System Update & Directory Setup

Boss's Request: Make sure the Raspberry Pi OS is secure and organized.

Tasks:

- Refresh package lists and upgrade the system.

```
sudo apt update  
sudo apt upgrade
```

- Verify system details: kernel version, user, time.

```
# kernel version  
uname -r # or  
cat /proc/version  
  
# user  
whoami  
  
# time  
date # or  
date +"%H:%M:%S"
```

```
mohamed@iot ~> uname -r; whoami; date +"%H:%M:%S"  
6.8.0-65-generic  
mohamed  
19:18:58
```

- Create `/home/<username>/iot_logger` with subdirectories: logs, scripts, data.

```
mkdir -p ~/iot_logger/{logs,scripts,data} #or  
mkdir -p /home/$USER/iot_logger/{logs,scripts,data}
```

```
mohamed@iot ~/i/linux> mkdir -p ~/iot_logger/{logs,scripts,data}  
mohamed@iot ~/i/linux> ls /home/mohamed/iot_logger/  
data  logs  scripts
```

Open-Ended Questions:

- Draw or describe the Linux architecture layers (hardware → kernel → shell → user space). Where do system calls fit?

Hardware	CPU, memory, disk.
Kernel	It manages hardware resources (disk, memory, etc.), but it needs programs to execute commands. These commands usually come from user input, which the shell interprets and passes to the kernel.
Shell	It is a program that provides an interface between the user and the operating system. It interprets commands typed in the terminal and executes programs.
User space	Applications and user processes.
System calls	It sit between user applications and the kernel

- Explain the purpose of these directories: `/`, `/bin`, `/sbin`, `/usr`, `/etc`, `/var`.

Directory	Purpose
<code>/</code>	root of filesystem
<code>/bin</code>	contains basic commands for all users (like <code>ls</code> , <code>cp</code> , <code>mv</code>). (user)
<code>/sbin</code>	contains system administration commands (like <code>mount</code> , <code>reboot</code> , <code>shutdown</code>), usually only run by the root user . (admin)
<code>/usr</code>	Installed software, libraries, and read-only program data
<code>/etc</code>	Configuration files
<code>/var</code>	Variable data specific to the system. Files that dynamically change (database, cache directory, logs,...).

- Why does Linux treat everything as a file? Explain the difference between a program and a process.
 - Linux treat everything as a file for:
 - It gives a simple and unified way to interact with different resources.
 - Whether it's text on a disk, a device like a printer, or even memory, you can read/write to it using the same file operations.
 - This makes programming and system management easier, since the same tools (like `cat`, `ls`, `echo`) can work with files, devices, and processes in the same way.
 - **Program vs Process**
 - **Program** → static file (instructions stored on disk).
 - **Process** → running instance of a program in memory (with PID).