

# **Lab 1: Introduction to Linux and OS main components**

---

**Ahmed Baha Eddine Alimi**

**Assignment Report**

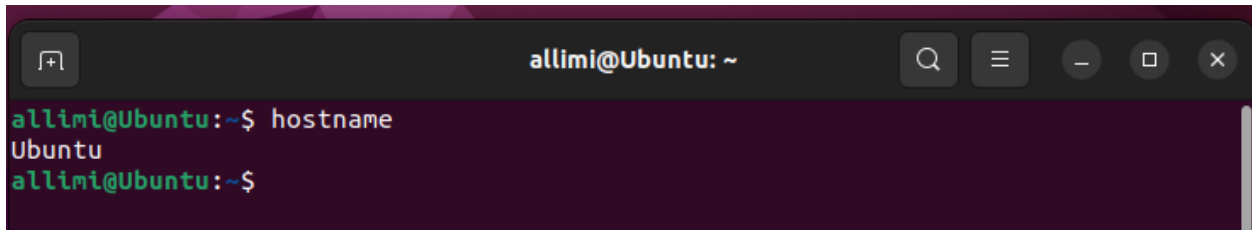
**SD-01**

# I. Introduction to Linux:

## 1. What is your machine hostname? How did you check it?

my machine host name is Ubuntu

I checked through **hostname** command:

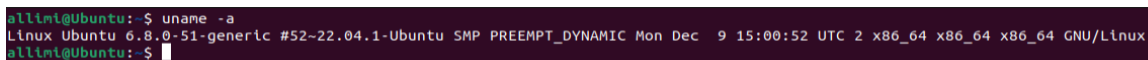
A terminal window with a dark purple background. The title bar shows 'allimi@Ubuntu: ~'. The prompt is 'allimi@Ubuntu:~\$'. The command 'hostname' has been entered, and the output 'Ubuntu' is displayed on the next line. The prompt 'allimi@Ubuntu:~\$' is shown again on the third line.

```
allimi@Ubuntu:~$ hostname
Ubuntu
allimi@Ubuntu:~$
```

## 2. What is the difference between /bin/bash and /bin/sh?

The /bin/bash (Bourne Again Shell) is a feature-rich shell widely used for its advanced scripting capabilities, including support for functions, arrays, and brace expansion. In contrast, /bin/sh (Bourne Shell) is a simpler shell that provides basic scripting functionality and adheres to the POSIX standard. Often, /bin/sh is a symbolic link to another shell like dash or bash, but when invoked as /bin/sh, it runs in a compatibility mode that ensures POSIX compliance and limits some of the more advanced features found in /bin/bash.

## 3. Explain all the details of the output from the command **uname -a**.

A terminal window with a dark purple background. The title bar shows 'allimi@Ubuntu: ~'. The prompt is 'allimi@Ubuntu:~\$'. The command 'uname -a' has been entered, and the output is displayed on the next line. The prompt 'allimi@Ubuntu:~\$' is shown again on the third line.

```
allimi@Ubuntu:~$ uname -a
Linux Ubuntu 6.8.0-51-generic #52-22.04.1-Ubuntu SMP PREEMPT_DYNAMIC Mon Dec 9 15:00:52 UTC 2 x86_64 x86_64 x86_64 GNU/Linux
allimi@Ubuntu:~$
```

The **uname -a** command provides system information:

Kernel name (**Linux**): The operating system kernel.

Host name (**Ubuntu**): The hostname of the system.

Kernel release (**6.8.0-51-generic**): The kernel version.

Kernel version (**#52-Ubuntu SMP PREEMPT\_DYNAMIC Mon Dec 9 15:00:52 UTC 2**):  
Detailed build version and timestamp.

Machine architecture (**x86\_64**): Hardware architecture.

Processor (**x86\_64**): The type of processor (often the same as the machine architecture)

Hardware platform: (**x86\_64**) : The hardware platform (often redundant and matching the architecture)

Operating system (**GNU/Linux**): OS type.

## 4. What command typically shows you the manual for POSIX compliant tools on the Linux operating system?

The **man** command displays the manual pages for tools

## II. GPT:

### 1. What is **fdisk** utility used for?

The **fdisk** utility is used to manipulate disk partition tables. It allows you to view partition information, Create, delete, or resize partitions and Change partition types.

### 2. Show the bootable device(s) on your machine, and identify which partition(s) are bootable.

```
allimi@ubuntu:~$ sudo fdisk -l
Disk /dev/loop0: 4 KiB, 4096 bytes, 8 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/loop1: 74,27 MiB, 77881344 bytes, 152112 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/loop2: 271,24 MiB, 284413952 bytes, 555496 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/loop3: 505,09 MiB, 529625088 bytes, 1034424 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/loop4: 91,69 MiB, 96141312 bytes, 187776 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/loop5: 12,93 MiB, 13553664 bytes, 26472 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/loop6: 38,83 MiB, 40714240 bytes, 79520 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/loop7: 500 KiB, 512000 bytes, 1000 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/sda: 25 GiB, 26843545600 bytes, 52428800 sectors
Disk model: VBOX HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: 9A2B4AC7-8142-4A5F-8827-4A0AF8BBF3C8

Device      Start      End  Sectors  Size Type
/dev/sda1    2048      4095     2048    1M BIOS boot
/dev/sda2    4096   1054719   1050624   513M EFI System
/dev/sda3  1054720 52426751 51372032 24,5G Linux filesystem
allimi@ubuntu:~$
```

Bootable Partitions: **/dev/sda1**, **/dev/sda2**.

**Bootable Device:** The disk `/dev/sda` is the bootable device, it contains:  
A BIOS boot partition (`/dev/sda1`) for compatibility with legacy booting.  
An EFI partition (`/dev/sda2`) for UEFI booting.

### 3. What is logical block address?

Logical Block Address (LBA) is a method used to address sectors on a storage device. It assigns a unique integer to each sector, starting from 0, making it easier to locate data regardless of the physical structure of the disk.

### 4. Why did we specify the `count`, the `bs`, and the `skip` options when using `dd`?

These options allow precise control over the data being manipulated:

`count`: Specifies how many blocks to read/write.

`bs`: Defines the block size for reading/writing.

`skip`: Skips a certain number of blocks before starting the operation.

### 5. Why does a GPT formatted disk have the MBR?

GPT disks include a "protective MBR" to prevent older, non-GPT-aware tools from misinterpreting the disk as unallocated. This MBR occupies the first sector and ensures backward compatibility.

### 6. Name two differences between primary and logical partitions in an MBR partitioning scheme.

Primary Partitions, a maximum of 4 can be created and can directly store data and boot OS, while Logical Partitions reside within an extended partition, and unlimited logical partitions can be created.

## III. UEFI Booting:

### 1. Why is Shim used to load the GRUB bootloader?

Shim is a small bootloader signed by a trusted certificate. It bridges the gap between UEFI Secure Boot and GRUB, ensuring GRUB can be loaded even with Secure Boot enabled.

### 2. Can you locate your grub configuration file? Show the path.\*

The GRUB configuration file is typically located at: **/etc/default/grub**

The compiled configuration file is: **/boot/grub/grub.cfg**

```
allimi@Ubuntu:~$ cat /etc/default/grub
# If you change this file, run 'update-grub' afterwards to update
# /boot/grub/grub.cfg.
# For full documentation of the options in this file, see:
#   info -f grub -n 'Simple configuration'

GRUB_DEFAULT=0
GRUB_TIMEOUT_STYLE=hidden
GRUB_TIMEOUT=0
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash"
GRUB_CMDLINE_LINUX=""

# Uncomment to enable BadRAM filtering, modify to suit your needs
# This works with Linux (no patch required) and with any kernel that obtains
# the memory map information from GRUB (GNU Mach, kernel of FreeBSD ...)
#GRUB_BADRAM="0x01234567,0xfefefefefe,0x89abcdef,0xefefefef"

# Uncomment to disable graphical terminal (grub-pc only)
#GRUB_TERMINAL=console

# The resolution used on graphical terminal
# note that you can use only modes which your graphic card supports via VBE
# you can see them in real GRUB with the command `vbeinfo'
#GRUB_GFXMODE=640x480

# Uncomment if you don't want GRUB to pass "root=UUID=xxx" parameter to Linux
#GRUB_DISABLE_LINUX_UUID=true

# Uncomment to disable generation of recovery mode menu entries
#GRUB_DISABLE_RECOVERY="true"

# Uncomment to get a beep at grub start
#GRUB_INIT TUNE="480 440 1"
```

3. According to the boot order, what is the third boot device on your computer? How did you check this?\*

```
allimi@Ubuntu:~$ sudo efibootmgr -v
BootCurrent: 0004
Timeout: 0 seconds
BootOrder: 0004,0000,0001,0002,0003
Boot0000* UiApp FvVol(7cb8bdc9-f8eb-4f34-aaea-3ee4af6516a1)/FvFile(462caa21-7614-4503-836e-8ab6f4662331)
Boot0001* UEFI VBOX CD-ROM VB2-01700376 PciRoot(0x0)/Pci(0x1,0x1)/Ata(1,0,0)N....YM....R,Y.
Boot0002* UEFI VBOX HARDISK VBbed73ed2-e4b2f344 PciRoot(0x0)/Pci(0xd,0x0)/Sata(0,65535,0)N....YM....R,Y.
Boot0003 EFI Internal Shell FvVol(7cb8bdc9-f8eb-4f34-aaea-3ee4af6516a1)/FvFile(7c04a583-9e3e-4f1c-ad65-e05268d0b4d1)
Boot0004* ubuntu HD(2,GPT,0db50176-06de-4009-be2b-29a247d59050,0x1000,0x100800)/File(\EFI\ubuntu\shimx64.efi)
allimi@Ubuntu:~$
```

The third boot device in the boot order is **Boot0001**, which corresponds to **UEFI VBOX CD-ROM VB2-01700376**.  
I used **efibootmgr** command.