Operating System

LAB - 09

Objectives:

1. Understanding the concept of HDD geometry, CHS and LBA addressing
2. Importance of partitions, and different types of partition tables
3. Details of MBR partitioning scheme
4. File systems and formatting a partition

Resources:

- Video Lecture 16 (Disk Geometry): https://www.youtube.com/watch?v=JNKhmSynRVM
- Video Lecture 17 (Disk Partitioning): https://www.youtube.com/watch?v=N-rrmeP1O2g
- Video Lecture 18 (Disk Formatting): https://www.youtube.com/watch?v=2w_kdcdVRMo

Task 1:

- **Write down the difference between a spinning disk and a solid state disk?**

Aspect	Spinning Disk (HDD)	Solid State Disk (SSD)
Mechanism	Uses spinning magnetic platters and a mechanical read/write head.	Uses flash memory with no moving parts.
Speed	Slower due to mechanical latency (seek time and rotational delay).	Faster because there are no moving parts (direct data access).
Durability	More prone to physical damage due to moving parts.	More durable and resistant to shocks and vibrations.
Cost	Cheaper per GB of storage.	More expensive per GB of storage.
Lifespan	Longer lifespan with regular usage but may degrade mechanically.	Limited write cycles but better endurance for read operations.

Noise	Generates noise due to spinning platters.	Completely silent.

- **Define Logical Block addressing? What is the maximum disk size support on a 24 bit LBA and on a 28 bit LBA?**

  Logical Block Addressing (LBA) is a method of accessing data on storage devices where each block is assigned a unique logical address. It abstracts the physical geometry of the disk (CHS) and simplifies data access.

Maximum Disk Size on 24-bit LBA:

- 2^24 × block size

Maximum Disk Size on 28-bit LBA:

- 2^28 × block size

- **What do you mean by a hard disk interface? Mention features of some of important HDD interfaces.**

A **hard disk interface** is a standard that defines how a hard drive communicates with the computer. It specifies the data transfer protocol and speed.

- **Important HDD Interfaces and Their Features:**

Interface	Features
IDE/PATA	Parallel interface; supports up to 133 MB/s; older standard.
SATA	Serial interface; faster (up to 6 Gb/s); supports hot-swapping.
SCSI	High-performance parallel interface; used in servers and enterprise setups.
NVMe	Ultra-fast interface for SSDs; communicates directly with the PCIe bus.
USB	External drives; supports plug-and-play; slower than SATA and NVMe.

- **Explain how reading and writing of a CHS disk is performed and differentiate between seek time and rotational delay? Describe how the mapping of CHS to LBA reduces seek time.**

Reading and Writing on a CHS Disk:

- **C** (Cylinder): Determines the track on the platter.
- **H** (Head): Identifies the read/write head for the specific surface.
- **S** (Sector): Locates the specific block on the track.
- Process:
    1. The disk head seeks the correct cylinder (seek time).
    2. It waits for the disk to rotate to the correct sector (rotational delay).
    3. The read/write operation is performed.

Seek Time vs. Rotational Delay:

Term	Definition
Seek Time	Time taken by the read/write head to move to the correct cylinder.
Rotational Delay	Time taken for the disk to rotate and align the desired sector with the head.

Mapping CHS to LBA and Reducing Seek Time

- **CHS to LBA Mapping:**
    - Logical Block Addressing (LBA) simplifies addressing by treating the disk as a continuous array of blocks rather than separate cylinders, heads, and sectors.
    - Formula: LBA=(C×HPC+H)×SPT+S−1\text{LBA} = (\text{C} \times \text{HPC} + \text{H}) \times \text{SPT} + \text{S} - 1LBA=(C×HPC+H)×SPT+S−1
        - **C**: Cylinder number
        - **HPC**: Heads per cylinder
        - **H**: Head number
        - **SPT**: Sectors per track
        - **S**: Sector number
- **Reducing Seek Time:**
    - By abstracting CHS geometry into a single LBA addressing scheme, the drive minimizes mechanical head movements, thus reducing seek time.
    - LBA allows the operating system to optimize read/write operations with

logical block arrangements, improving efficiency.

Task 2:

- **Write down five advantages of partitioning your hard disk.**

1. **Data Organization**: Allows better management of files by separating system, application, and personal data.
2. **Improved Performance**: Reduces file fragmentation and improves read/write efficiency by isolating frequently accessed files.
3. **Multiple Operating Systems**: Enables installation and management of multiple operating systems on the same disk.
4. **Enhanced Security**: Isolates sensitive data from the system partition, reducing the risk of accidental deletion or corruption.
5. **Simplified Backups**: Makes it easier to back up specific partitions without including the entire disk.

- **Differentiate between primary partition and logical partition.**

Aspect	Primary Partition	Logical Partition
Definition	A partition directly defined in the partition table.	A subpartition within an extended partition.
Limit	A maximum of 4 primary partitions per disk.	Unlimited logical partitions can exist within one extended partition.
Bootable	Can be made bootable and used for system installations.	Cannot be bootable; only used for data storage.
Placement	Defined at the disk's root level.	Exists within an extended partition.
Usage	Typically used for OS or critical system files.	Suitable for organizing user data or additional file storage.

- **What do you mean by a partition table? Draw a detailed schematic view of the partition table of your hard disk.**

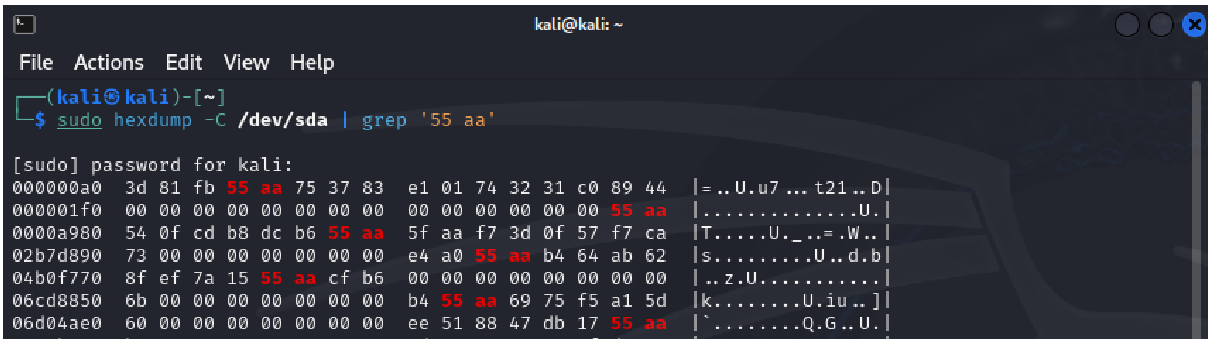
A partition table is a data structure on a hard disk that describes the division of the disk into partitions. It contains information about:

- Partition type
- Start and end addresses
- Size of each partition
- Boot flag

Schematic View of a Partition Table (MBR):

Byte Offset	Field	Description
0x1BE	Partition 1 Entry	Contains type, starting sector, and size.
0x1CE	Partition 2 Entry	Similar structure for second partition.
0x1DE	Partition 3 Entry	Similar structure for third partition.
0x1EE	Partition 4 Entry	Similar structure for fourth partition.
0x1FE	Boot Signature	Indicates if the disk is bootable (0x55AA).

- **Give a shell command that displays boot signature of your hard disk.**



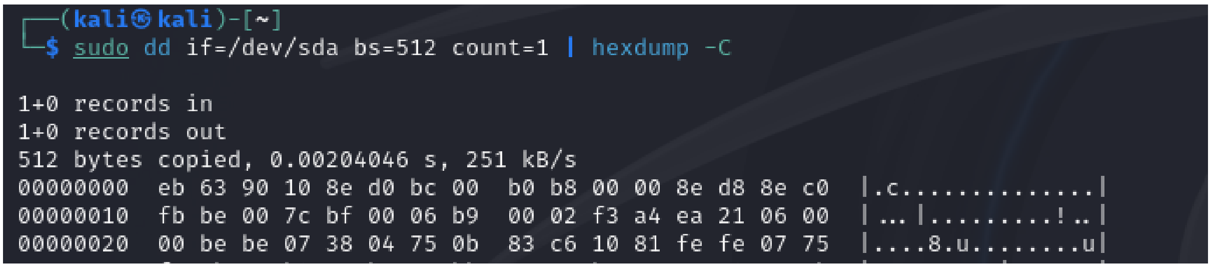
```
┌──(kali㉿kali)-[~]
└─$ sudo hexdump -C /dev/sda | grep '55 aa'

[sudo] password for kali:
000000a0  3d 81 fb 55 aa 75 37 83  e1 01 74 32 31 c0 89 44  |=..U.u7...t21..D|
000001f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 55 aa  |..............U.|
0000a980  54 0f cd b8 dc b6 55 aa  5f aa f7 3d 0f 57 f7 ca  |T.....U._..=.W..|
02b7d890  73 00 00 00 00 00 00 00  e4 a0 55 aa b4 64 ab 62  |s.........U..d.b|
04b0f770  8f ef 7a 15 55 aa cf b6  00 00 00 00 00 00 00 00  |..z.U...........|
06cd8850  6b 00 00 00 00 00 00 00  b4 55 aa 69 75 f5 a1 5d  |k........U.iu..]|
06d04ae0  60 00 00 00 00 00 00 00  ee 51 88 47 db 17 55 aa  |`.......Q.G..U.|
```

The command `sudo hexdump -C /dev/sda | grep '55 aa'` is inspecting the **raw contents of your disk** (`/dev/sda`) in hexadecimal format and searching for the **boot signature**, which is a marker (`55 AA`) indicating that the disk is bootable.

- **Give a shell command that displays the stage 1 boot loader program on your hard disk.**

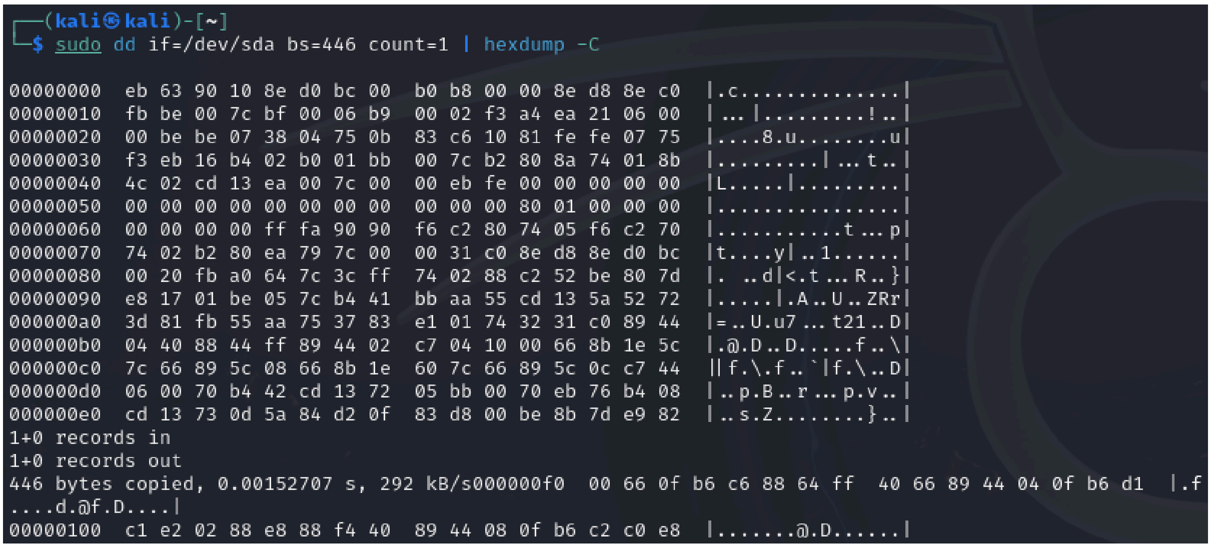
The command reads the first 512 bytes (one sector) from the hard disk device /dev/sda, which is the Master Boot Record (MBR), and displays it in a hexadecimal and ASCII format using hexdump.



```
┌──(kali㉿kali)-[~]
└─$ sudo dd if=/dev/sda bs=512 count=1 | hexdump -C

1+0 records in
1+0 records out
512 bytes copied, 0.00204046 s, 251 kB/s
00000000  eb 63 90 10 8e d0 bc 00  b0 b8 00 00 8e d8 8e c0  |.c..............|
00000010  fb be 00 7c bf 00 06 b9  00 02 f3 a4 ea 21 06 00  |...|.........!..|
00000020  00 be be 07 38 04 75 0b  83 c6 10 81 fe fe 07 75  |....8.u........u|
```

To specifically extract just the Stage 1 bootloader code, which resides in the first 446 bytes of the MBR, you can use this command:

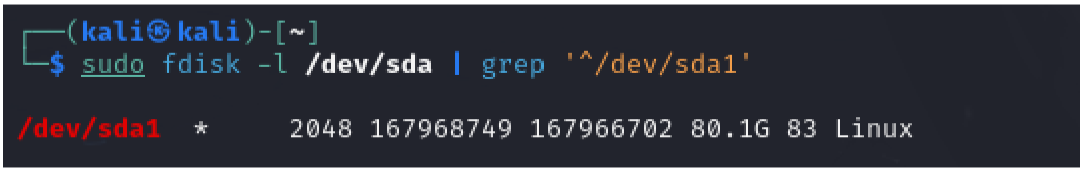


```
┌──(kali㉿kali)-[~]
└─$ sudo dd if=/dev/sda bs=446 count=1 | hexdump -C

00000000  eb 63 90 10 8e d0 bc 00  b0 b8 00 00 8e d8 8e c0  |.c..............|
00000010  fb be 00 7c bf 00 06 b9  00 02 f3 a4 ea 21 06 00  |...|.........!..|
00000020  00 be be 07 38 04 75 0b  83 c6 10 81 fe fe 07 75  |....8.u........u|
00000030  f3 eb 16 b4 02 b0 01 bb  00 7c b2 80 8a 74 01 8b  |.........|...t..|
00000040  4c 02 cd 13 ea 00 7c 00  00 eb fe 00 00 00 00 00  |L.....|.........|
00000050  00 00 00 00 00 00 00 00  00 00 00 80 01 00 00 00  |................|
00000060  00 00 00 00 ff fa 90 90  f6 c2 80 74 05 f6 c2 70  |...........t...p|
00000070  74 02 b2 80 ea 79 7c 00  00 31 c0 8e d8 8e d0 bc  |t....y|..1......|
00000080  00 20 fb a0 64 7c 3c ff  74 02 88 c2 52 be 80 7d  |. ..d|<.t...R..}|
00000090  e8 17 01 be 05 7c b4 41  bb aa 55 cd 13 5a 52 72  |.....|.A..U..ZRr|
000000a0  3d 81 fb 55 aa 75 37 83  e1 01 74 32 31 c0 89 44  |=..U.u7...t21..D|
000000b0  04 40 88 44 ff 89 44 02  c7 04 10 00 66 8b 1e 5c  |.@.D..D.....f..\|
000000c0  7c 66 89 5c 08 66 8b 1e  60 7c 66 89 5c 0c c7 44  ||f.\.f..`|f.\..D|
000000d0  06 00 70 b4 42 cd 13 72  05 bb 00 70 eb 76 b4 08  |..p.B..r...p.v..|
000000e0  cd 13 73 0d 5a 84 d2 0f  83 d8 00 be 8b 7d e9 82  |..s.Z........}..|
1+0 records in
1+0 records out
446 bytes copied, 0.00152707 s, 292 kB/s000000f0  00 66 0f b6 c6 88 64 ff  40 66 89 44 04 0f b6 d1  |.f
....d.@f.D....|
00000100  c1 e2 02 88 e8 88 f4 40  89 44 08 0f b6 c2 c0 e8  |.......@.D......|
```

The Stage 1 bootloader code initializes hardware and loads the next stage of the boot process (like GRUB or another bootloader) into memory.

- **Give a shell command that displays the partition type of the first partition of your hard disk.**



```
┌──(kali㉿kali)-[~]
└─$ sudo fdisk -l /dev/sda | grep '^/dev/sda1'

/dev/sda1  *      2048 167968749 167966702 80.1G 83 Linux
```

- **Mention at least five different partition types along with their numbers that your system supports.**

Partition Type	Number
Linux (ext4)	**83**
Linux Swap	**82**
FAT16	**6**

FAT32	**0x0B**
NTFS/HPFS	**7**
EFI System Partition	**EF**
Linux LVM	**8E**

- **Use fdisk command to create two primary and six logical partitions on your system, with appropriate sizes and mount points. Justify during viva.**

Open `fdisk` for your target disk:

```
sudo fdisk /dev/sda
```

1. **Create the First Primary Partition**:
   - Press `n` (for new partition).
   - Select `p` (for primary).
   - Choose partition number `1`.
   - Specify the starting sector (press Enter to accept the default).
   - Specify the size (e.g., `+10G` for a 10 GB partition).
2. **Create the Second Primary Partition**:
   - Press `n` (for new partition).
   - Select `p` (for primary).
   - Choose partition number `2`.
   - Specify the starting sector.
   - Specify the size (e.g., `+20G` for a 20 GB partition).
3. **Create an Extended Partition**:
   - Press `n` (for new partition).
   - Select `e` (for extended).
   - It will take the remaining disk space automatically.
   - Specify the starting sector and press Enter.
4. **Create Logical Partitions Within the Extended Partition**:
   - Press `n` (for new partition).
   - Select `l` (for logical).
   - Specify the starting sector and size (e.g., `+5G` for a 5 GB partition).
   - Repeat this step to create five more logical partitions (adjust sizes as needed).
5. **Verify Partition Table**:
   - Press `p` to print the current partition table and verify all partitions.
6. **Write Changes to Disk**:

- Press w to write the changes to the disk.

Mount Points Setup

Format the Partitions: Format each partition with a file system (e.g., ext4):

```
sudo mkfs.ext4 /dev/sda1  # First primary partition
sudo mkfs.ext4 /dev/sda2  # Second primary partition
sudo mkfs.ext4 /dev/sda5  # First logical partition
sudo mkfs.ext4 /dev/sda6  # Second logical partition
```

1. Create Mount Points: Create directories where partitions will be mounted:

```
sudo mkdir -p /mnt/part1 /mnt/part2 /mnt/part3 /mnt/part4
/mnt/part5 /mnt/part6 /mnt/part7 /mnt/part8
```

Mount the Partitions: Mount each partition to its respective directory:

```
sudo mount /dev/sda1 /mnt/part1
sudo mount /dev/sda2 /mnt/part2
sudo mount /dev/sda5 /mnt/part3
sudo mount /dev/sda6 /mnt/part4
sudo mount /dev/sda7 /mnt/part5
sudo mount /dev/sda8 /mnt/part6
sudo mount /dev/sda9 /mnt/part7
sudo mount /dev/sda10 /mnt/part8
```

Persist the Mount Points: Add the partitions to `/etc/fstab` for automatic mounting:

```
sudo nano /etc/fstab
```

Add lines like:

```
/dev/sda1   /mnt/part1   ext4   defaults   0 0
/dev/sda2   /mnt/part2   ext4   defaults   0 0
/dev/sda5   /mnt/part3   ext4   defaults   0 0
```

Commands to Validate the Partitions

1. **List Partitions**:
   `sudo fdisk -l`

2. **Check Mounted Partitions**:
   `lsblk`

3. **Test Mount Points**:
   `df -h`

Task 3:

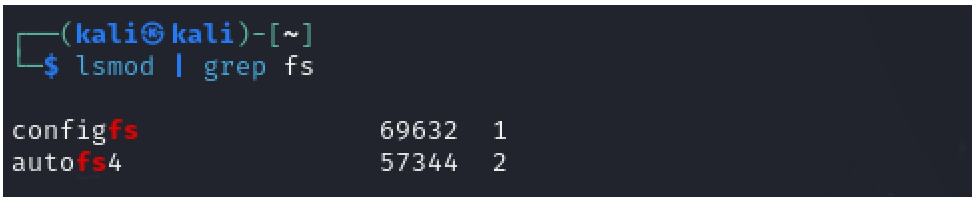
- **Define file system and journaling filesystem. Name the functionalities that a good file system must offer.**

File System:
A file system is a method of organizing, storing, and retrieving data on a storage device. It defines how data is structured on a disk (e.g., files, directories, metadata) and provides a means of accessing it.
Journaling File System:
A journaling file system logs changes in a dedicated area called a journal before committing them to the main file system. This ensures integrity and allows recovery in case of system crashes. Examples include ext3, ext4, and NTFS.

Functionalities a Good File System Must Offer

- Efficient data organization.
- Support for large file sizes and partitions.
- Security features (e.g., access permissions).

- **Give command that displays the list of currently loaded file system drivers, and write the output.**
  Use the *lsmod* command to display loaded kernel modules, including file system drivers:



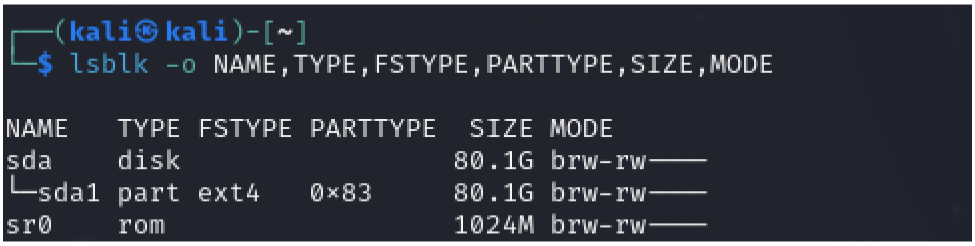
```
┌──(kali㉿kali)-[~]
└─$ lsmod | grep fs

configfs               69632  1
autofs4                57344  2
```

- Mention the max file size support and maximum partition size support of **ext3**, **ext4**, **vfat**, **ntfs**, and **zfs.**

File System	Max File Size	Max Partition Size
ext3	2 TB	16 TB
ext4	16 TB	1 Exabyte
vfat	4 GB	2 TB
NTFS	16 EB	16 EB
ZFS	16 EB	256 Quadrillion Zettabytes

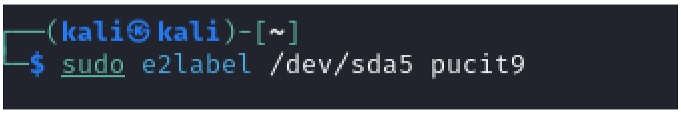
- **Give a shell command to display the name, type, fstype, parttype, size, and mode of the hard disk attached with your system. Write down a snap shot of the output and describe to TAs.**



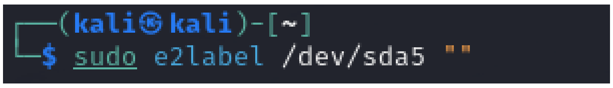
```
┌──(kali㉿kali)-[~]
└─$ lsblk -o NAME,TYPE,FSTYPE,PARTTYPE,SIZE,MODE

NAME   TYPE FSTYPE PARTTYPE  SIZE MODE
sda    disk                 80.1G brw-rw----
└─sda1 part ext4   0×83     80.1G brw-rw----
sr0    rom                  1024M brw-rw----
```

- **Give a command to assign a label “pucit9” to the first logical partition of your only attached scsi hard disk, and later give a command to undo it.**

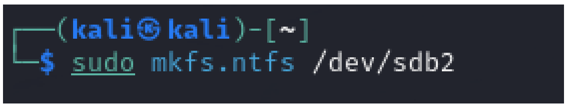


```
┌──(kali㉿kali)-[~]
└─$ sudo e2label /dev/sda5 pucit9
```

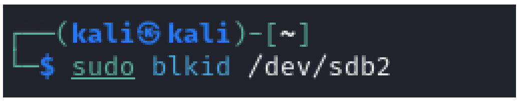


```
┌──(kali㉿kali)-[~]
└─$ sudo e2label /dev/sda5 ""
```

- **Give a command to format the second partition of the second scsci hard disk attached with your system to ntfs. Later give a command to confirm.**



```
┌──(kali㉿kali)-[~]
└─$ sudo mkfs.ntfs /dev/sdb2
```



```
┌──(kali㉿kali)-[~]
└─$ sudo blkid /dev/sdb2
```

The blkid command is used to display block device attributes such as UUID (Universally Unique Identifier), label, and filesystem type. It provides information about the specified partition or disk.