

Lab 7 Manage Local Storage

Part I- Check your file system

Step 1. Execute the `ls /dev/sd*` command to see the current hard disk devices.

Explanation: I use the command **sudo ls /dev/sd*** required by the lab to list my block stage devices, including hard disks & partitions, in my system.

Screenshot:

```
(kali@kali)-[~]
└─$ sudo ls /dev/sd*
[sudo] password for kali:
/dev/sda /dev/sda1
```

Step 2. Execute the `fdisk -l` command to list the current hard disk partitions.

Explanation: Still using **sudo** command as required by the lab, followed by command **fdisk -l** to display in detail info about my partitions on the storage devices in my system that are findable at this point.

Screenshot:

```
(kali@kali)-[~]
└─$ sudo fdisk -l
Disk /dev/sda: 80.09 GiB, 86000000000 bytes, 167968750 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: dos
Disk identifier: 0x3d57c577

Device      Boot Start      End  Sectors  Size Id Type
/dev/sda1   *    2048 167968749 167966702  80.1G 83 Linux
```

Step 3. Execute the `parted -l` command to list the current hard disk partition table.

Explanation: Still using **sudo** command as required by the lab, followed by command **parted -l** to output the partition tables and their structure for the connected drives.

Screenshot:

```
(kali@kali)-[~]
└─$ sudo parted -l
Model: ATA VBOX HARDDISK (scsi)
Disk /dev/sda: 86.0GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

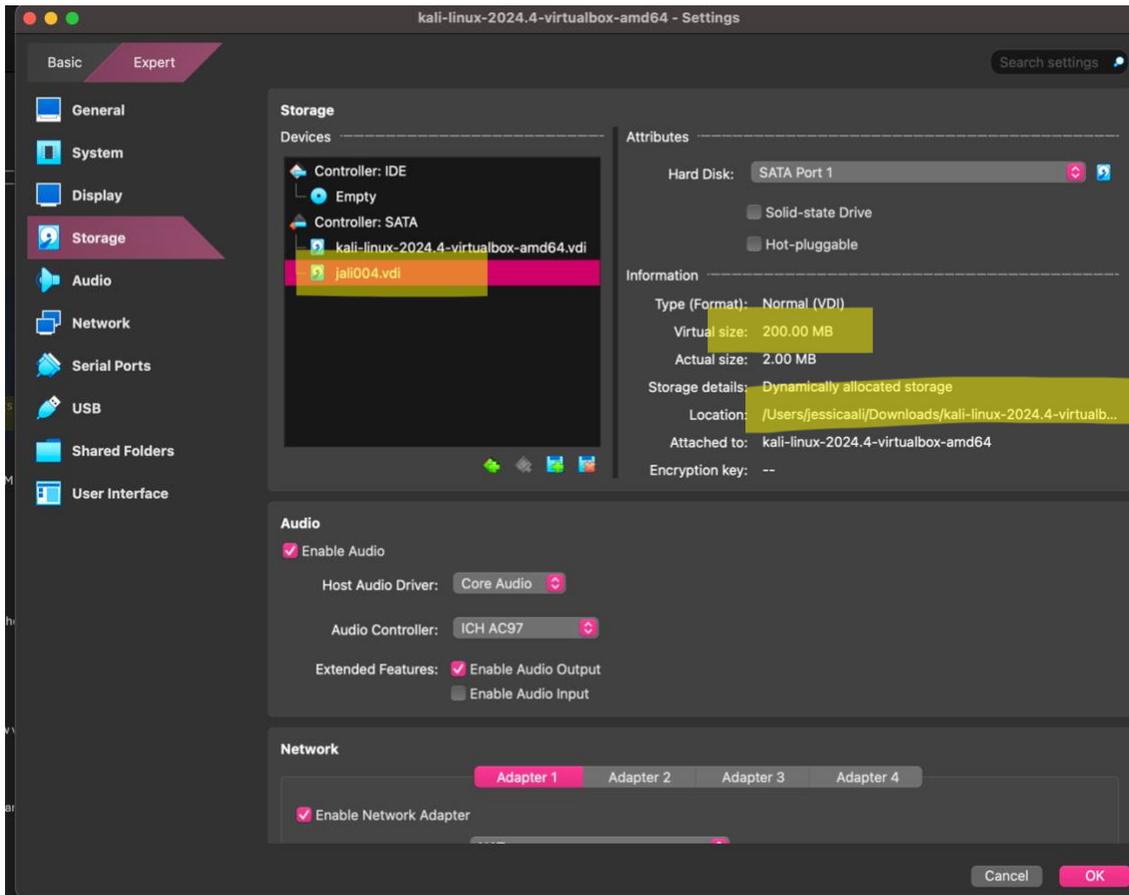
Number   Start   End     Size    Type     File system  Flags
  1       1049kB  86.0GB  86.0GB  primary  ext4         boot
```

Part II- Create a new virtual disk

Step 1. In the VM setting, attach a new virtual hard disk with the size of 200 MB to our current Linux VM. Name it as “your_midas.vdi”

Explanation: I add a new 200MB v/m from my Virtual Box GUI by setting the size, name (I use **jali004** here as required by the lab), path, etc.

Screenshot:



Step 2. Load this virtual hard disk to your virtual machine.

Explanation: I use command **sudo** followed by **ls /dev/sd*** to check if my new disk is recognized. It is, notice the 3rd output, **/dev/sdb**.

Screenshot:



```
File Actions Edit View Help
(kali㉿kali)-[~]
└─$ sudo ls /dev/sd*
[sudo] password for kali:
/dev/sda  /dev/sda1  /dev/sdb
```

Step 3. Repeat the steps in Part I and highlight the differences after adding the new virtual hard disk.

Explanation: I use commands as follows **sudo ls dev/ sd*** , **sudo fdisk -l** , and end with **sudo parted -l**. To summarize, there is an **/dev/sdb**, a new disk entry of 200 Mib w/o partitions, and lastly the inclusion of a new **/dev/sdb** with an unknown partition table. Which means it is ready to be setup. You can find the screenshot on the next page for a visual summary required by the lab. It is important to note moving forward the new device name is **/dev/sdb** and will not display as the filename **jali004.vdi**. The file name was just what VirtualBox uses to store the virtual disk on my Mac. Screenshot can be found on the next page.

Screenshot:

```
(kali㉿kali)-[~]
└─$ sudo ls /dev/sd*
/dev/sda /dev/sda1 /dev/sdb

(kali㉿kali)-[~]
└─$ sudo fdisk -l
Disk /dev/sdb: 200 MiB, 209715200 bytes, 409600 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

Disk /dev/sda: 80.09 GiB, 86000000000 bytes, 167968750 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: dos
Disk identifier: 0x3d57c577

Device      Boot Start          End      Sectors  Size Id Type
/dev/sda1   *      2048 167968749 167966702  80.1G 83 Linux

(kali㉿kali)-[~]
└─$ sudo parted -l
Model: ATA VBOX HARDDISK (scsi)
Disk /dev/sda: 86.0GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number  Start   End     Size    Type    File system  Flags
  1      1049kB 86.0GB 86.0GB  primary ext4          boot

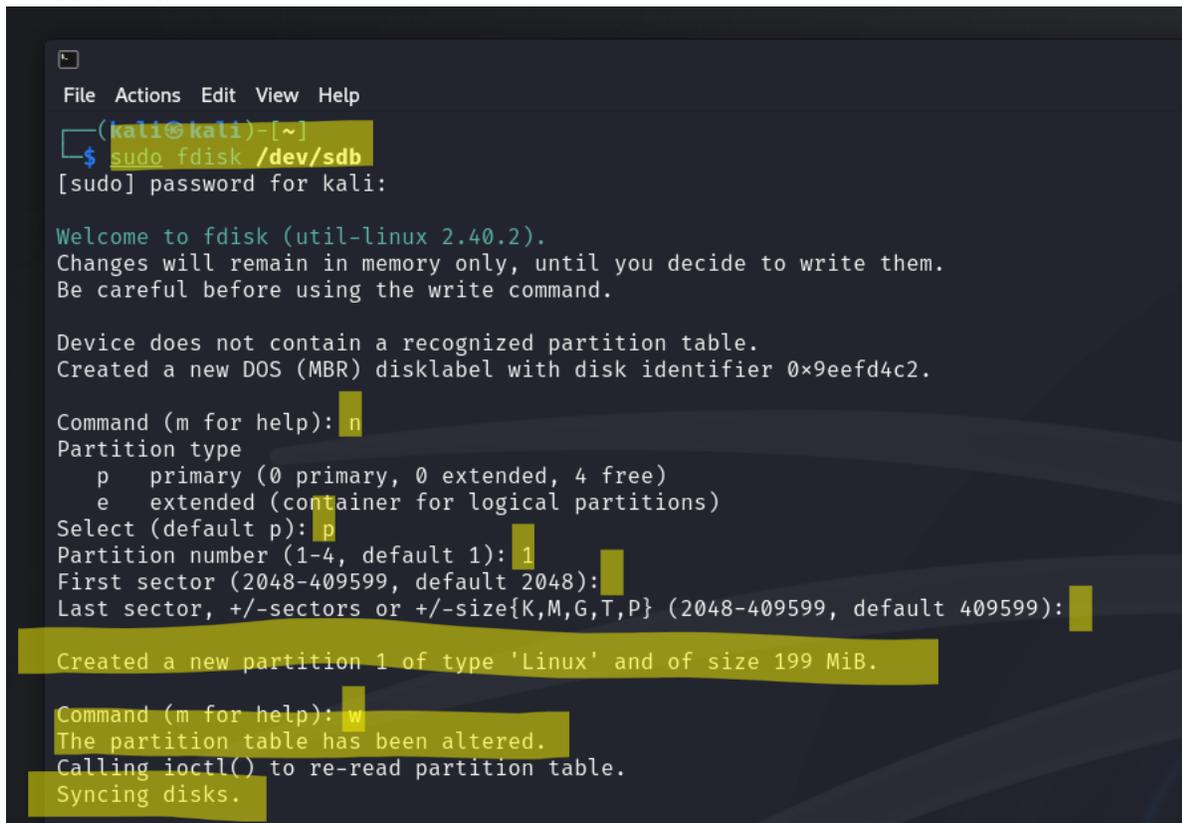
Error: /dev/sdb: unrecognised disk label
Model: ATA VBOX HARDDISK (scsi)
Disk /dev/sdb: 210MB
Sector size (logical/physical): 512B/512B
Partition Table: unknown
Disk Flags:
```

Part III– Creating Partitions and Filesystems

Step 1. Use the `fdisk` command to create a new primary partition on the new virtual hard disk attached in Part II.

Explanation: I use command **sudo fdisk /dev/sdb** to enter into the new virtual hard disk and create a partition. Once inside `fdisk`, I press **n** to create a new partition, followed by **p** to make it primary, followed by **1** to allocate a partition number, I press **enter** to accept the default first sector, I press **enter** again to accept the default last sector, then press **enter** again to create the new partition followed by **w** to confirm the write changes.

Screenshot:



```
File Actions Edit View Help
(kali@kali)-[~]
└─$ sudo fdisk /dev/sdb
[sudo] password for kali:

Welcome to fdisk (util-linux 2.40.2).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table.
Created a new DOS (MBR) disklabel with disk identifier 0x9eefd4c2.

Command (m for help): n
Partition type
  p   primary (0 primary, 0 extended, 4 free)
  e   extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1): 1
First sector (2048-409599, default 2048):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-409599, default 409599):

Created a new partition 1 of type 'Linux' and of size 199 MiB.

Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.
```

Step 2. Use the correct command to create an `ext4` filesystem on the new partition.

Explanation: It is important to note that I am working with `/dev/sdb1` because I am referring to the first partition on `/dev/sdb`. I use command **sudo mkfs.ext4** followed by what I am altering **/dev/sdb1**. Screenshot can be found on the next page.

Screenshot:

```
(kali@kali)-[~]
└─$ sudo mkfs.ext4 /dev/sdb1
mke2fs 1.47.1 (20-May-2024)
Creating filesystem with 203776 1k blocks and 51000 inodes
Filesystem UUID: 65a75e54-cd15-41ff-ad99-ff1a1975fb90
Superblock backups stored on blocks:
    8193, 24577, 40961, 57345, 73729

Allocating group tables: done
Writing inode tables: done
Creating journal (4096 blocks): done
Writing superblocks and filesystem accounting information: done
```

Step 3. Repeat the steps in Part I and highlight the differences.

Explanation: I use command **sudo fdisk -l** and then **sudo parted -l** to list out in detail so I can compare. Before there were no partitions listed under my /dev/sdb but now there are. The partition size now contains around 200 MB. I also see in my -l output that there is a partition listed as 1 under my /dev/sdb. Revealing a formatted ext4 partition on my disk. Screenshot can be found on the next page.

Screenshot:

```
(kali@kali)-[~]
└─$ sudo fdisk -l
Disk /dev/sdb: 200 MiB, 209715200 bytes, 409600 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: dos
Disk identifier: 0x9eefd4c2

Device            Boot Start      End  Sectors  Size Id Type
/dev/sdb1          2048 409599   407552   199M 83 Linux

Disk /dev/sda: 80.09 GiB, 86000000000 bytes, 167968750 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: dos
Disk identifier: 0x3d57c577

Device            Boot Start      End  Sectors  Size Id Type
/dev/sda1          *    2048 167968749 167966702 80.1G 83 Linux

(kali@kali)-[~]
└─$ sudo parted -l
Model: ATA VBOX HARDDISK (scsi)
Disk /dev/sda: 86.0GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number  Start   End     Size    Type    File system  Flags
  1      1049kB 86.0GB 86.0GB  primary ext4          boot

Model: ATA VBOX HARDDISK (scsi)
Disk /dev/sdb: 210MB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number  Start   End     Size    Type    File system  Flags
  1      1049kB 210MB   209MB  primary ext4
```

Step 4. Make a new directory named /cyse. And mount the new partition under this directory.

Explanation: I use command **sudo mkdir** followed by what is required by the lab **/cyse**. I use the command **sudo mount /dev/sdb1 /cyse** to allow accessibility to the partition through the /cyse directory. Screenshot is found on the following page.

Screenshot:

```
(kali@kali)-[~/home/kali]
└─$ sudo mkdir /cyse

(kali@kali)-[~/home/kali]
└─$ sudo mount /dev/sdb1 /cyse
```

Step 5. Use the df command to check the mounting point of the new partition.

Explanation: I use command **df -h** to display that I correctly executed step 4. I did as seen in the last line of the screen shot below. My file system and where it is mounted are all correct.

Screenshot:

```
(kali@kali)-[~/home/kali]
└─$ df -h

```

Filesystem	Size	Used	Avail	Use%	Mounted on
udev	926M	0	926M	0%	/dev
tmpfs	198M	960K	197M	1%	/run
/dev/sda1	79G	16G	60G	21%	/
tmpfs	988M	4.0K	988M	1%	/dev/shm
tmpfs	5.0M	0	5.0M	0%	/run/lock
tmpfs	1.0M	0	1.0M	0%	/run/credentials/systemd-tmpfiles-setup-dev-early.service
tmpfs	1.0M	0	1.0M	0%	/run/credentials/systemd-udev-load-credentials.service
tmpfs	1.0M	0	1.0M	0%	/run/credentials/systemd-journald.service
tmpfs	1.0M	0	1.0M	0%	/run/credentials/systemd-sysctl.service
tmpfs	1.0M	0	1.0M	0%	/run/credentials/systemd-tmpfiles-setup-dev.service
tmpfs	988M	12K	988M	1%	/tmp
tmpfs	1.0M	0	1.0M	0%	/run/credentials/systemd-tmpfiles-setup.service
tmpfs	1.0M	0	1.0M	0%	/run/credentials/getty@tty1.service
tmpfs	198M	120K	198M	1%	/run/user/1000
/dev/sdb1	181M	63K	167M	1%	/cyse

Step 6. Create a new file named for YourMIDAS.txt (replace YourMIDAS with your MIDAS ID) in the directory /cyse and put your name in that file.

Explanation: I use the command **echo** followed by my name for an output, then a pipe, followed by **sudo tee** to create the file and define where it will be. I do this to demonstrate that the partition created for the lab to verify it is writable.

Screenshot:

```
(kali@kali)-[~/home/kali]
└─$ echo "Jess" | sudo tee /cyse/001166237.txt
Jess
```

Step 7. Unmount /cyse directory.

Explanation: I use command **sudo umount** followed by the directory **cyse** because it is required by the lab. I am unmounting the partition from the mount point and safely disconnecting it from my file system.

Screenshot:

```
(kali@kali)-[~/home/kali]
└─$ sudo umount /cyse
```

Step 8. Check the contents in /cyse directory. What do you find?

Explanation: I use command **ls /cyse** to confirm the directory is empty because the partition was unmounted. I do not have access to this directory anymore and I can not see the file I previously created within directory cyse.

Screenshot:

```
(kali@kali)-[~/home/kali]
└─$ ls /cyse
```