

# Dummy Disk Creation from Hardware, Kernel to User Level for Testing

## Purpose Linux

**Author:** Aashish Chaudhary

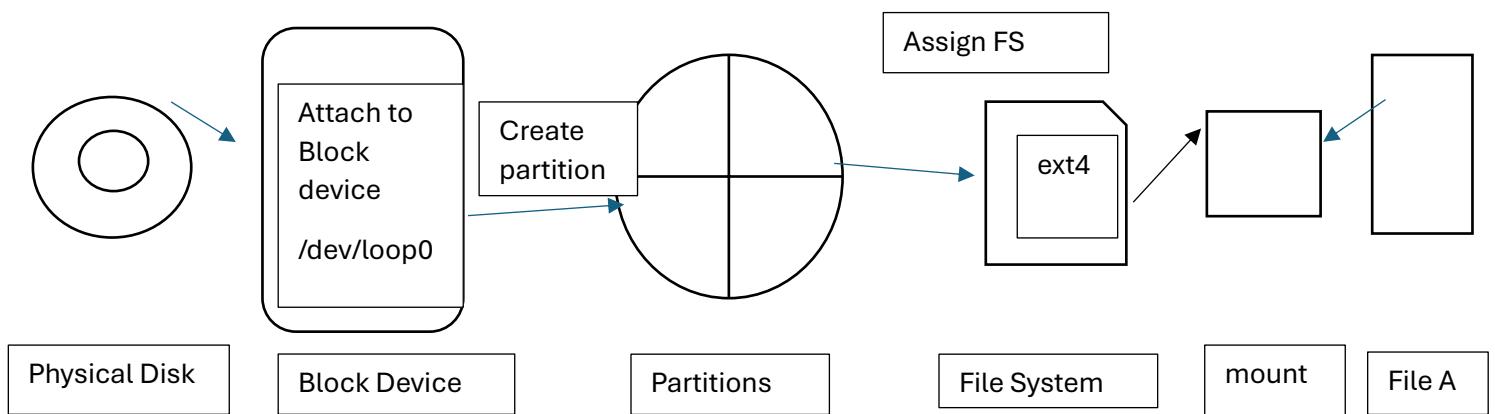
Concepts: Linux Storage creation on different layers using dd, df, lsblk, mkfs, losetup, mount, umount, fdisk, and others.

### Overview:

In this project, I have created a dummy disk from a file that uses the space from the root filesystem. And then attached it to the block-device. After the attachment, I created partition (in my case primary partition), assigned that partition an ext4 filesystem. Then that partition is mounted to the mount point. After that, we can write data according to what filesystem we assigned to that partition. After this, I reversed same process to successfully remove the attached device from the system. This way, the data doesn't get lost, and the device can be removed successfully without hurting or losing the existing Filesystem and Data. This is very useful for anyone who wants to use any physical storage (i.e. HDD, SSD, USB, or any Optical Medias) in Linux System.

### Mental Model: How does Files gets stored in Linux under the hood?

-→



## **Fig: Mental Model of from Disk to File Creation**

In Linux, in low level, Physical Disk gets attached to the block device files then we create Partitions on to it. On the partitions, we create or assign FileSystems as what kind of data we want to store on. After Creating FileSystems, we mount it to the mount point, and we can simply write data into it.

Note: If we want to remove the files all the way to the Hardware Level, we simply reverse this process. In simple explanation, we delete files on directory, then unmount the mount points, detach the physical device from the block-device and finally remove the hardware.

### **Steps:**

1. File Creation from the root filesystem.

Command:

```
dd if=/dev/zero of=/root/storage_lab.img bs=1M count=1024
```

-→Creates a 1Gig file filled with zero that we can use it as Disk

2. Attach the file to the block device

```
losetup -fP /root/storage_lab.img
```

-→Attach the file to the block device

3. Create Partition to the newly created Block-device

```
fdisk /dev/loop0
```

-→ creates the partition to the /dev/loop0. In my case its one primary. So its /dev/loop0p1

4. Assign FileSystem to the partition

```
-→ mkfs.ext4 /dev/loop0p1
```

Creates the ext4 filesystem to the partition

5. Create a Mount Point and mount the partition to it

**mkdir /mnt/storage\_lab** → creates mount point

**mount /dev/loop0p1 /mnt/storage\_lab** → mount device to the mount point

6. Navigate to the mounted point and write data to it.

**cd /mnt/storage\_lab** → navigate to the directory

**touch myfile.txt** → create the empty file and write to it

**Note: Don't forget to use sudo privileges while carrying out all these tasks.**

As I already mentioned, to remove the device, simply reverse the steps. If we are using the file (like in my case) we need to delete the created file as well. That's an additional step to it.

Troubleshooting:

1. If the device reboots the device attached from loopdriver loses. I had to redo everything from the beginning. And running fdisk changed the partition blocks. So there was mismatch superblock error on the drive. Running the following command and remounting the drive works perfectly fine.

Command: **sudo fsck /dev/loop0p1** and hit 'y'