Hardware

- Check hard drive for bad sectors in Linux
- Zero a hard drive in Linux
- Formatting a drive in linux

Check hard drive for bad sectors in Linux

Check for bad sectors with status

```
sudo badblocks -v -s /dev/sdX
```

- -v: Verbose output, lists bad sectors.
- -s: Shows progress as a percentage of completion.
- For a non-destructive test, use -n instead: sudo badblocks -n -v /dev/sdX.
- To save bad sector list: sudo badblocks -v -o badsectors.txt /dev/sdX.

If you need larger block sizes use this:

```
sudo badblocks -v -s -b 4096 /dev/sdX
```

This will give you 3KB block sizes.

Zero a hard drive in Linux

You might want to check the drive to see if it is still good in the first place. Go here first: Check hard drive for bad sectors

1. Unmount the drive: (if you haven't already)

```
sudo umount /dev/sdX*
```

   - Check with lsblk to confirm unmounting

2. Write zeros with progress:

```
sudo dd if=/dev/zero of=/dev/sdX bs=4M status=progress
```

- status=progress displays transfer progress (e.g., bytes written).

Verify drive identifier:

Ensure sdX matches your drive to avoid data loss.

Formatting a drive in linux

To partition and format your drive in Linux
Open the parted utility and then you will work inside (parted) that.

To partition and format your drive in Linux after zeroing, follow these steps. Replace `sdX` with your drive's identifier (check with `lsblk`). Ensure the drive is unmounted (`sudo umount /dev/sdX*`) and data is backed up, as these steps will erase existing structures.

Before you start: unmount your drive if its mounted. "lsblk" to find out.

1. Create a Partition Table and Partition:

```
sudo fdisk /dev/sdX
```

- Press `g` to create a new GPT partition table. (MBR is for booting drive)
- Press `n` to create a new partition:
  - Select default partition number (e.g., 1). If you want all space of the drive (one partition)
  - Select default first sector.
  - Select default last sector (or specify size, e.g., `+100G` for 100GB).
- Press `w` to write changes and exit.

2. Format the Partition:

```
sudo mkfs.ext4 /dev/sdX1
```

- This formats the first partition (`sdX1`) as ext4. Replace `ext4` with `vfat` for FAT32, `ntfs` for NTFS, etc.
- Progress is shown during formatting.

3. Verify:

```
lsblk -f
```

- Confirms the new partition and filesystem.

Notes:
- Check drive with `lsblk` to confirm `sdX`.

- For multiple partitions, repeat `n` in `fdisk` for each partition, then format each (e.g., `mkfs.ext4 /dev/sdX2`).
- If errors occur, ensure the drive is not mounted (`umount /dev/sdX*`).

For a more complex formatting, you can use "parted" to do that.

This is a simple example:
`sudo parted /dev/sdb`

Now you are in "parted"

Create a partition
- For MBR
`mklabel msdos`
- For GPT
`mklabel gpt`

The basic syntax for partition command
-`mkpart 'partition type' 'file system' start end`
If you want all space of the drive (one partition)
`mkpart primary ext4 1MiB 100%`
If you want more than one partition
`mkpart primary ext4 1MiB 2GB`
If you want to create another partition, use
`mkpart primary ext4 2GB 3GB`

To see the partitions
`print`

? Help?
`isblk`