Creating Virtual Disks Using Linux Command Line

[1]

Linux is indeed a great system with excellent tools[2] at our disposal. There are lots of things that can be achieved using the terminal. One such activity is creating virtual hard drives. Your Linux system should already have the tools required to do this without the need for virtual machine software.

NOTE: this tutorial only covers creating fixed-size disk images whose partitions can be mounted using Linux. Virtual machine programs like VirtualBox allow you to create dynamically expanding virtual drives that increase in size whenever necessary. If your goal is to create disks for programs such as VirtualBox, you need to use its tools.

Tools Requirements

The following commands that you will need are:

- ‘dd’ for creating the file. You can also bximage (part of the Bochs PC Emulator) if you wish.
- ‘fdisk’ for creating partitions, or which ever partition program you like.
- ‘mkfs’ for formatting the partitions.
- ‘losetup’ for setting up the loop devices for each partition.

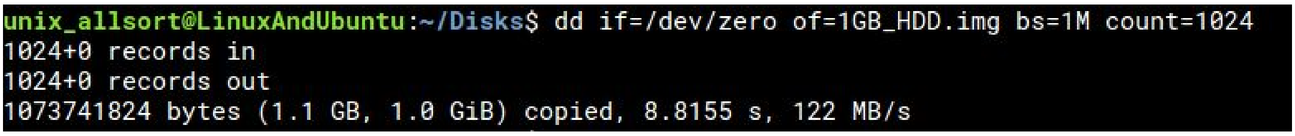
You may use whatever tool you are comfortable with to achieve the same goal, of course.

Create The Image and Format Partitions

Creating the image is simple using ‘dd’. All this will do is write zeros to a file of a specified size. For example, let’s create a 1GB image:

```
$ dd if=/dev/zero of=1GB_HDD.img bs=1M count=1024
```

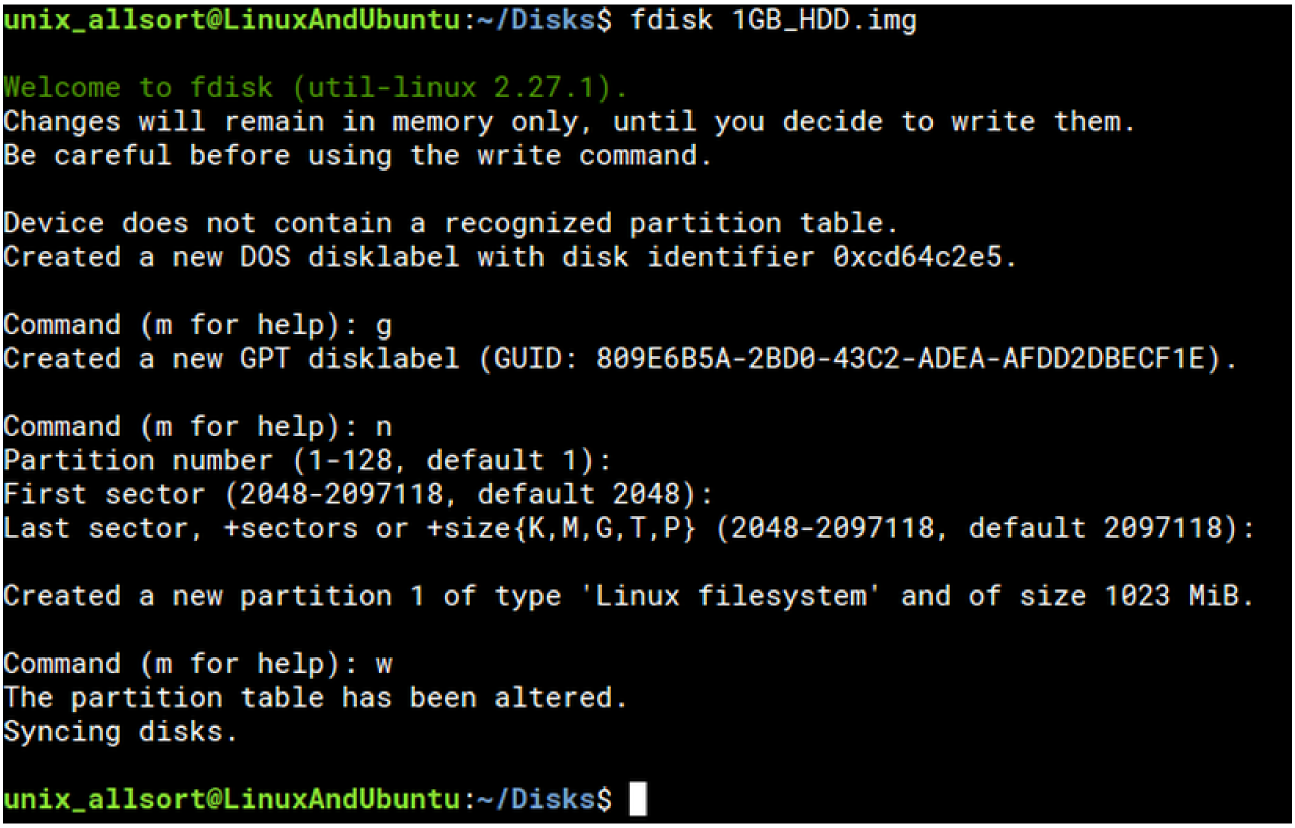
This will take a little time. You may choose a smaller or larger size if you wish.



```
unix_allsort@LinuxAndUbuntu:~/Disks$ dd if=/dev/zero of=1GB_HDD.img bs=1M count=1024
1024+0 records in
1024+0 records out
1073741824 bytes (1.1 GB, 1.0 GiB) copied, 8.8155 s, 122 MB/s
```

[3]

Once completed, a partition should be created using fdisk. Because there is no partition table, one will be created. This is the DOS partition table. Let's switch to a GPT table by entering 'g' into the prompt to create one. Now, create a partition by entering 'n'. Accept all of the defaults. The partition created will be in a native Linux format that can be either ext2, ext3, or ext4. Then write the changes to the image by entering 'w'.

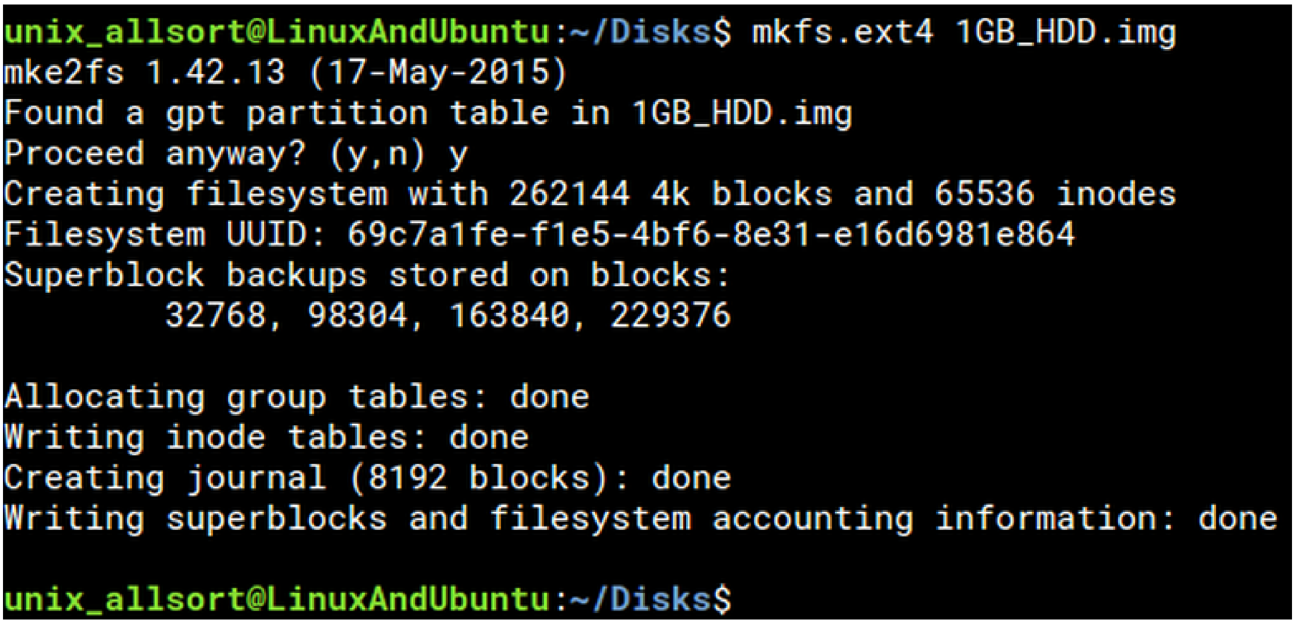


```
unix_allsort@LinuxAndUbuntu:~/Disks$ fdisk 1GB_HDD.img

Welcome to fdisk (util-linux 2.27.1).
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 disklabel with disk identifier 0xcd64c2e5.

Command (m for help): g
Created a new GPT disklabel (GUID: 809E6B5A-2BD0-43C2-ADEA-AFDD2DBECF1E).

Command (m for help): n
Partition number (1-128, default 1):
First sector (2048-2097118, default 2048):
Last sector, +sectors or +size{K,M,G,T,P} (2048-2097118, default 2097118):

Created a new partition 1 of type 'Linux filesystem' and of size 1023 MiB.

Command (m for help): w
The partition table has been altered.
Syncing disks.

unix_allsort@LinuxAndUbuntu:~/Disks$ █
```

[4]

After the changes are written and fdisk closes, all that is required to do is format the partition running 'mkfs.ext4' on the image file itself to create a ext4 partition. It may ask you if you wish to proceed anyway if a GPT partition is found. If so, say yes.

$ mkfs.ext4 1GB_HDD.img

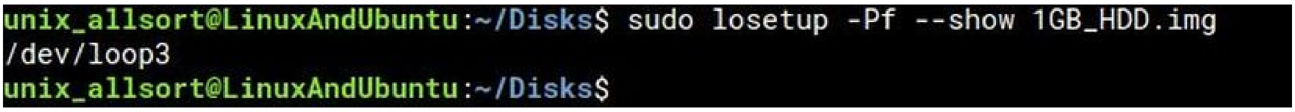


```
unix_allsort@LinuxAndUbuntu:~/Disks$ mkfs.ext4 1GB_HDD.img
mke2fs 1.42.13 (17-May-2015)
Found a gpt partition table in 1GB_HDD.img
Proceed anyway? (y,n) y
Creating filesystem with 262144 4k blocks and 65536 inodes
Filesystem UUID: 69c7a1fe-f1e5-4bf6-8e31-e16d6981e864
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376

Allocating group tables: done
Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done

unix_allsort@LinuxAndUbuntu:~/Disks$
```

[5]

If all went well, you can then proceed to setup a loop device for your image. This requires the use of 'losetup' (that is, loop setup). The command we wish to run will assign an available loop device (-f parameter to find one) to the partition on the image, and show the name of said loop device (--show parameter):

$ sudo losetup -Pf --show 1GB_HDD.img



```
unix_allsort@LinuxAndUbuntu:~/Disks$ sudo losetup -Pf --show 1GB_HDD.img
/dev/loop3
unix_allsort@LinuxAndUbuntu:~/Disks$
```

[6]

If successful, you should be able to access the partition by either using 'mount' or through your file manager.

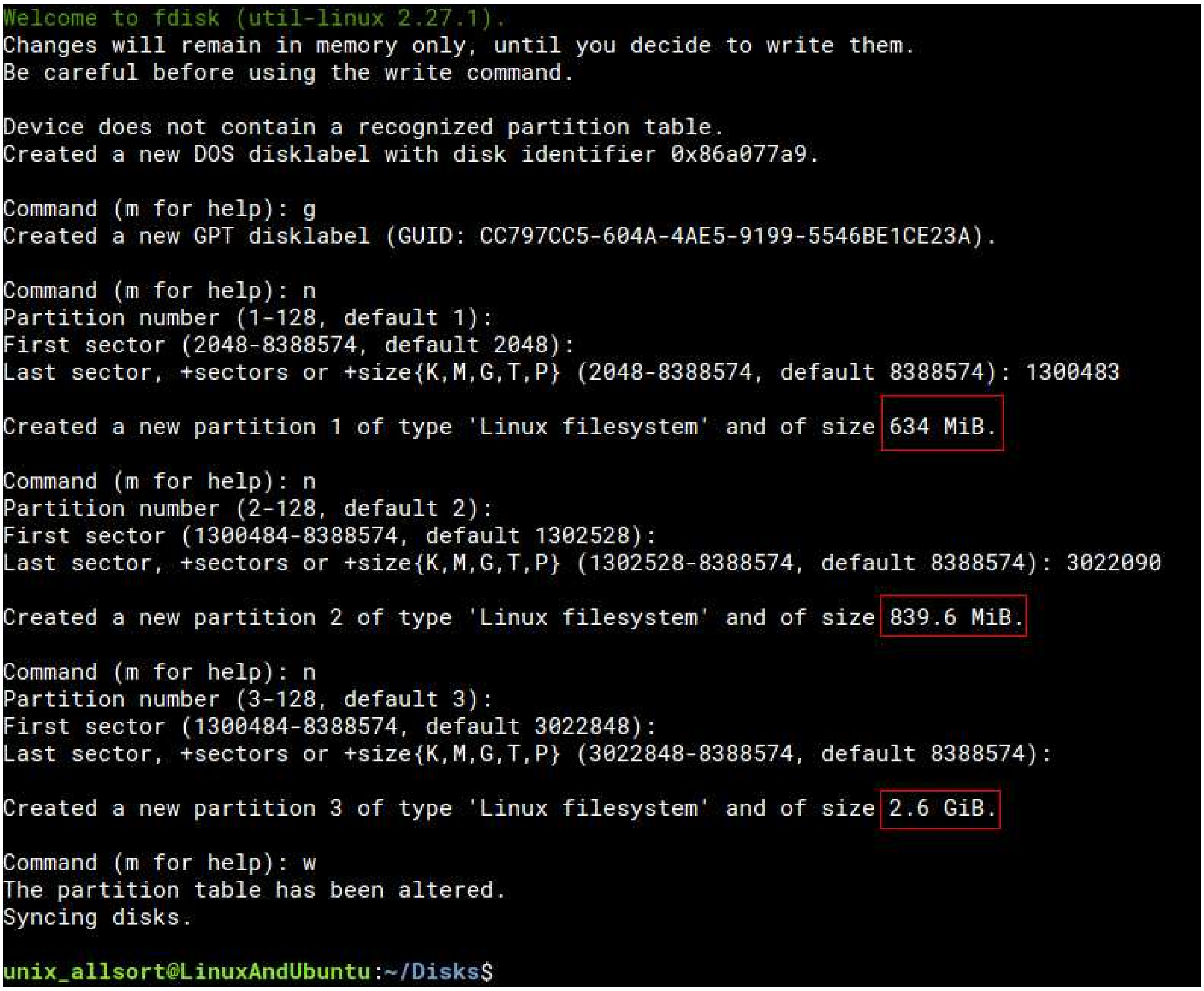
Images With Multiple Partitions

That was how you create virtual drives with a single partition. What about images with two or more partitions? There are a few extra steps necessary, but once you know what to do, it should still be quite simple.

Begin by creating a 4GB image:

$ dd if=/dev/zero of=4GB_HDD.img bs=1M count=4096

Use fdisk to create three Linux partitions with a GPT partition table. I chose the size of my partitions randomly. Feel free to choose the size of each partition yourself.

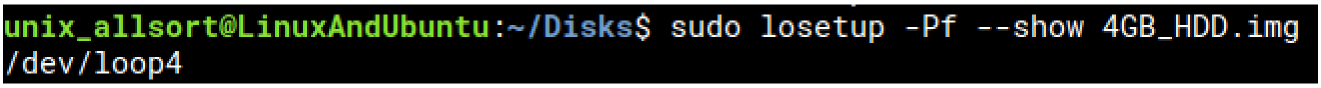


```
Welcome to fdisk (util-linux 2.27.1).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table.
Created a new DOS disklabel with disk identifier 0x86a077a9.

Command (m for help): g
Created a new GPT disklabel (GUID: CC797CC5-604A-4AE5-9199-5546BE1CE23A).

Command (m for help): n
Partition number (1-128, default 1):
First sector (2048-8388574, default 2048):
Last sector, +sectors or +size{K,M,G,T,P} (2048-8388574, default 8388574): 1300483

Created a new partition 1 of type 'Linux filesystem' and of size 634 MiB.

Command (m for help): n
Partition number (2-128, default 2):
First sector (1300484-8388574, default 1302528):
Last sector, +sectors or +size{K,M,G,T,P} (1302528-8388574, default 8388574): 3022090

Created a new partition 2 of type 'Linux filesystem' and of size 839.6 MiB.

Command (m for help): n
Partition number (3-128, default 3):
First sector (1300484-8388574, default 3022848):
Last sector, +sectors or +size{K,M,G,T,P} (3022848-8388574, default 8388574):

Created a new partition 3 of type 'Linux filesystem' and of size 2.6 GiB.

Command (m for help): w
The partition table has been altered.
Syncing disks.

unix_allsort@LinuxAndUbuntu:~/Disks$
```

[7]

Now we need to run ‘losetup’ to gain access to each partition by assigning loop devices to each one.

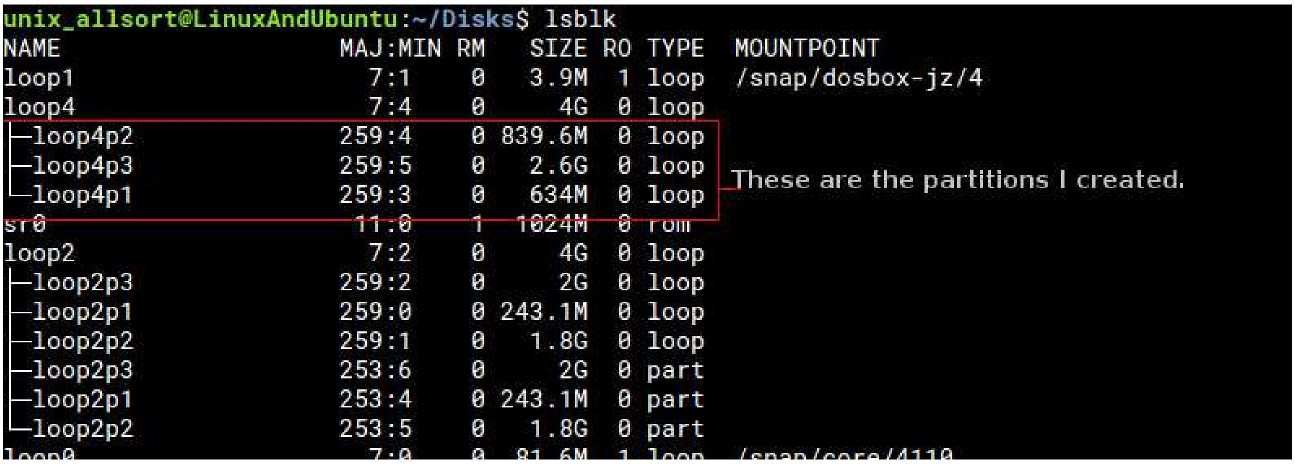
$ sudo losetup -Pf --show 4GB_HDD.img



```
unix_allsort@LinuxAndUbuntu:~/Disks$ sudo losetup -Pf --show 4GB_HDD.img
/dev/loop4
```

[8]

As before, we wanted to see what loop device was chosen. However, this time, the ‘-P’ parameter was useful in this case because it tells ‘losetup’ to scan the image for any partitions to create loop devices for. When the loop interfaces are created, have a look at ‘lsblk’ to see the created devices.

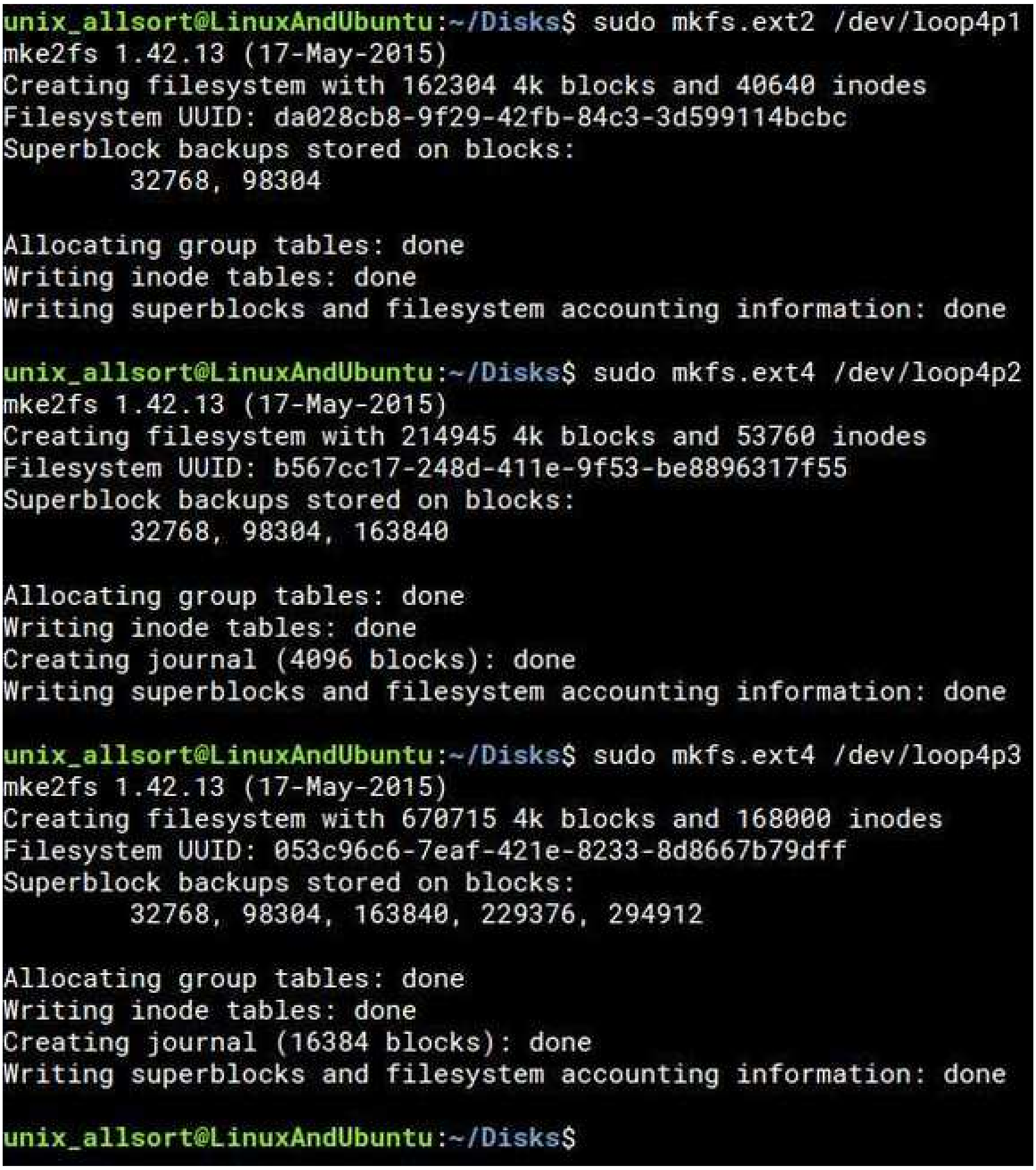


```
unix_allsort@LinuxAndUbuntu:~/Disks$ lsblk
NAME                MAJ:MIN RM   SIZE RO TYPE  MOUNTPOINT
loop1                 7:1    0   3.9M  1 loop  /snap/dosbox-jz/4
loop4                 7:4    0     4G  0 loop
├─loop4p2           259:4    0 839.6M  0 loop
├─loop4p3           259:5    0   2.6G  0 loop
└─loop4p1           259:3    0   634M  0 loop
sr0                  11:0    1  1024M  0 rom
loop2                 7:2    0     4G  0 loop
├─loop2p3           259:2    0     2G  0 loop
├─loop2p1           259:0    0 243.1M  0 loop
├─loop2p2           259:1    0   1.8G  0 loop
├─loop2p3           253:6    0     2G  0 part
├─loop2p1           253:4    0 243.1M  0 part
└─loop2p2           253:5    0   1.8G  0 part
loop0                 7:0    0  81.6M  1 loop  /snap/core/4110
```

These are the partitions I created.

[9]

After that, each partition needs to be formatted before use, so run ‘mkfs’ to create them. Try running ‘mkfs.ext2’ on the first partition to create an ext2 filesystem. Then run ‘mkfs.ext4’ on the other two to create ext4 filesystems on the image. Once they’re formatted, you should be able to mount them via the command line or a file manager.

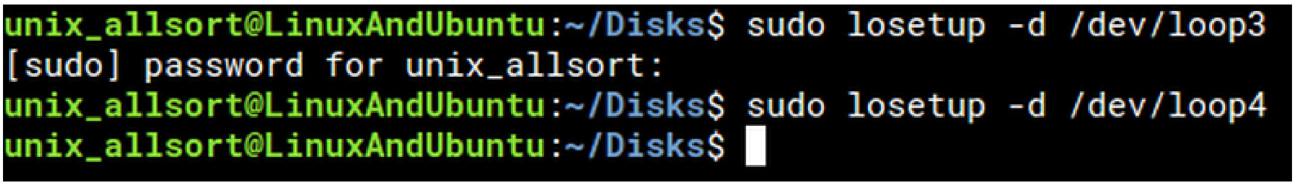


```
unix_allsort@LinuxAndUbuntu:~/Disks$ sudo mkfs.ext2 /dev/loop4p1
mke2fs 1.42.13 (17-May-2015)
Creating filesystem with 162304 4k blocks and 40640 inodes
Filesystem UUID: da028cb8-9f29-42fb-84c3-3d599114bcbc
Superblock backups stored on blocks:
        32768, 98304

Allocating group tables: done
Writing inode tables: done
Writing superblocks and filesystem accounting information: done

unix_allsort@LinuxAndUbuntu:~/Disks$ sudo mkfs.ext4 /dev/loop4p2
mke2fs 1.42.13 (17-May-2015)
Creating filesystem with 214945 4k blocks and 53760 inodes
Filesystem UUID: b567cc17-248d-411e-9f53-be8896317f55
Superblock backups stored on blocks:
        32768, 98304, 163840

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

unix_allsort@LinuxAndUbuntu:~/Disks$ sudo mkfs.ext4 /dev/loop4p3
mke2fs 1.42.13 (17-May-2015)
Creating filesystem with 670715 4k blocks and 168000 inodes
Filesystem UUID: 053c96c6-7eaf-421e-8233-8d8667b79dff
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done

unix_allsort@LinuxAndUbuntu:~/Disks$
```

[10]

Finish partitions

If you are finished with the partitions, simply run ‘losetup’ to remove the loop device you wish.



```
unix_allsort@LinuxAndUbuntu:~/Disks$ sudo losetup -d /dev/loop3
[sudo] password for unix_allsort:
unix_allsort@LinuxAndUbuntu:~/Disks$ sudo losetup -d /dev/loop4
unix_allsort@LinuxAndUbuntu:~/Disks$
```

[11]

Conclusion

To create a virtual drive with partitions on Linux is a very simple process. If you run into any trouble, do let me know in the comment section below this article. I'll to respond as soon as possible.

Links

1. http://www.linuxandubuntu.com/home/creating-virtual-disks-using-linux-command-line
2. http://www.linuxandubuntu.com/home/category/linux-tools
3. http://www.linuxandubuntu.com/uploads/2/1/1/5/21152474/create-image-and-format-partitions_orig.jpg
4. http://www.linuxandubuntu.com/uploads/2/1/1/5/21152474/fdisk-linux-partitioning-tool_orig.png
5. http://www.linuxandubuntu.com/uploads/2/1/1/5/21152474/mkfs-linux-tool_orig.png
6. http://www.linuxandubuntu.com/uploads/2/1/1/5/21152474/losetup-virtual-hard-disk-space_orig.jpg
7. http://www.linuxandubuntu.com/uploads/2/1/1/5/21152474/fdisk-linux-utility-tool_orig.jpg
8. http://www.linuxandubuntu.com/uploads/2/1/1/5/21152474/losetup-4gb-hdd_orig.png
9. http://www.linuxandubuntu.com/uploads/2/1/1/5/21152474/lsblk-linux-tool_orig.jpg
10. http://www.linuxandubuntu.com/uploads/2/1/1/5/21152474/mkfs-linux-utility_orig.jpg
11. http://www.linuxandubuntu.com/uploads/2/1/1/5/21152474/losetup-linux-partitioning-tool_orig.png