Lab 3

1. Use `fdisk -l` to locate information about the partition sizes.
2. Use `fdisk` to add a new logical partition that is 2GB in size.
   Fdisk /dev/sdb
3. Did the kernel feel the changes? Display the content of /proc/partitions file? What did you notice? How to overcome that?
4. Make a new `ext4` file system on the new logical partition you just created.

   Bonus: Try creating the `ext4` filesystem with 2k blocks and one inode per every 4k (two blocks) of filesystem.

5. Create a directory, name it `/data`.
6. Add a label to the new filesystem, name it `data`.
7. Add a new entry to `/etc/fstab` for the new filesystem using the label you just create.
8. Mount the new filesystem
9. Display your swap size.
10. Create a swap file of size 512MB.
11. Add the swap file to the virtual memory of the system.
12. Display the swap size.
13. Use the `fdisk` command to create 2 Linux LVM (0x8e) partitions using "unpartitioned" space on your hard disk. These partitions should all be the same size; to speed up the lab, do not make them larger than 300 MB each. Make sure to write the changes to disk by using the w command to exit the fdisk utility. Run the partprobe command after exiting the fdisk utility.

14. Initialize your Linux LVM partitions as physical volumes with the pvcreate command. You can use the pvdisplay command to verify that the partitions have been initialized as physical volumes.
15. Using only one of your physical volumes, create a volume group called test0. Use the vgdisplay command to verify that the volume group was created.

16. Create a small logical volume (LV) called data that uses about 30 percent of the available space of the test0 volume group. Look for VG Size and Free PE/Size in the output of the vgdisplay command to assist you with this. Use the lvdisplay command to verify your work.

17. Create an xfs filesystem on your new LV.
18. Make a new directory called /data and then mount the new LV under the /data directory. Create a "large file" in this volume.

19. Enlarge the LV that you created in Sequence 1 (/dev/test0/data) by using approximately 25 percent of the remaining free space in the test0 volume group. Then, enlarge the filesystem of the LV.

20. Verify that the file /data/bigfile still exists in the LV. Run the df command and check to verify that more free disk space is now available on the LV.

21. Use the remaining extents in the test0 volume group to create a second LV called docs.

22. Run the vgdisplay command to verify that there are no free extents left in the test0 volume group.

23. Create an xfs filesystem on the new LV, make a mount point called /docs and mount the docs LV using this mount point.

24. Add all of the remaining unused physical volumes that you created in Sequence 1 to the test0 volume group.

25. If you run vgdisplay again, there now should be free extents (provided by the new physical volumes) in the test0 volume group. Extend the docs LV and underlying filesystem to make use of all of the free extents of the test0 volume group. Verify your actions.