Hands-on Projects

These projects should be completed in the order given. The hands-on projects presented in this chapter should take a total of three hours to complete. The requirements for this lab include:

* A computer with Fedora Linux installed according to Hands-on Project 2-1.

Project 5-1

In this hands-on project, you view and create device files.

1. Boot your Fedora Linux virtual machine. Login to your chosen desktop environment as **user1** using password **LNXrocks!** and open up a terminal window.
2. At the terminal, become **root** by typing **su -** and press enter and provide **LNXrocks!** as the password.
3. At the command prompt, do a long listing of the **/dev/tty6** file. Recall that tty are the different terminals that you can access using **Ctrl+Alt+F#** keys. You should notice that tty is a character device. This makes sense since we input data to the terminal one character at a time. Observe the major and minor numbers.
4. At the command prompt, remove the **/dev/tty6** character device. Next, provide a long listing of the **/dev/tty6** file. The file should have been removed.
5. Switch to the command-line terminal (tty6) by pressing **Ctrl+Alt+F6** and attempt to log in to the terminal. Notice that since you deleted this terminal, it no longer will allow you to authenticate to the system under this tty. Switch back to the graphical display (tty2) by pressing **Ctrl+Alt+F2**, type the command **mknod /dev/tty6 c 4 6** at the command prompt, and press Enter. Next, provide a long listing of the **/dev/tty6** file. You should see the recreated device file.
6. **Provide screenshot(s) of steps 3 through 5.**
7. Switch to a command-line terminal (tty6) by pressing **Ctrl+Alt+F6** and log in to the terminal using the user name of **root** and the password of **LNXrocks!**. You should be able to use this terminal after recreating the file. This may take a moment or two depending on how quickly you have switched back to the tty.
8. Switch back to the graphical display (tty2) by pressing **Ctrl+Alt+F2.**
9. At the command prompt, provide a long listing of the **/dev/tty?** file. Review the different terminal devices and their associated major and minor numbers.
10. At the command prompt, type **ll /dev | less** and press Enter to list all the filenames underneath the /dev directory.
11. At the command prompt, cat the contents of **/proc/devices** to the terminal. You can determine major numbers from this file and correlate this to the information provided in step 10.
12. **Provide screenshot(s) of steps 7 through 11.**

## Project 5-2

In this hands-on project, you mount DVDs to the directory tree and view their contents.

1. In VirtualBox settings for your VM, attach the ISO image for Fedora Linux to the virtual CDROM drive as you did when we first installed Fedora. You can do this while your Fedora system is running.
2. Your desktop environment should prompt you that the ISO was inserted and you can open this up with the Files application. If you are not prompted, go to your activities screen and search for Files.
3. Take a few moments to explore the contents of the DVD within the Files application.
4. Open up a terminal and change to the **root** user with password of **LNXrocks!**.
5. At the command prompt, type **mount** and press Enter. Note the line at the bottom that lists your mounted DVD. Notice that it has been mounted to the expected mount point /run/media/user1/Fedora-WS-Live-26-1-5.
6. Type **umount /sr0** at the command prompt and press Enter to unmount the Fedora ISO image.
7. At the command prompt, create the **/mymount** directory to create the directory we will utilize for mounting the installation disk.
8. Next, create some files and directories within the **/mymount** directory. Type **mount -r /dev/cdrom /mymount** at the command prompt, and press Enter. We have mounted the ISO file as read only using the provided option to the /mymount directory.
9. At the command prompt, provide a long listing of the **/mymount** directory. You should see 4 directories that are provided from the Fedora ISO image.
10. At the command prompt, type **mount** and press Enter. Note the line at the bottom that lists your mounted DVD. Now pipe the output of the **mount** command to **grep** and grep for the string: **sr0**
11. Next, type **umount /mymount** at the command prompt and press Enter. Type the **mount** command at a command prompt utilizing the **grep** command for string: **sr0**.
12. At the command prompt, provide a long listing of the **/mymount** file. You will now see the files and directories you previously created in step 6. Recall that these files and directories are inaccessible when mounted over. Ideally you utilize an empty directory such as /mnt or /media.
13. In your virtualization software, detach the DVD ISO image for Fedora Linux to the DVD drive for the virtual machine. This performs the same action as ejecting the DVD from the physical DVD drive.
14. **Provide screenshot(s) of steps 2 through 13.**

## Project 5-3

In this hands-on project, you work with **standard** hard disk partitions. You will first create a hard disk partition using the fdisk utility. Next, you create an ext4 filesystem on the partition and mount it to the directory tree. Finally, you use the /etc/fstab file to automatically mount the partition at boot time.

1. At the terminal, become **root** by typing **su -** and press enter and provide **LNXrocks!** as the password.
2. Your hard drive should be referenced by **/dev/sda** but to double-check type **fdisk -l** and press Enter in a terminal. You should see /dev/sda as the disk and 2 partitions /dev/sda1 and /dev/sda2. /dev/sda2 should be Linux LVM.
3. At the command prompt, type **fdisk /dev/sda** and press Enter. You are now in the fdisk utility. Type **m** and press Enter to view the options of the fdisk utility. We should have at least 5GB of disk space free so we want to create 2 new partitions with this free space.
4. At the command prompt, type **n** and press Enter. You should have 2 free primary partitions to use. The default should say **p** so hit Enter to take the default. Otherwise, type **p** and press Enter. Your partition number should default to **3** press Enter to proceed. **Press** **Enter** to take the default **first sector**. Type **+4G** for the **last sector** and press Enter. You should see that output that you created your 3rd partition.
5. Now we will create the last partition we need. At the command prompt, type **n** and press Enter. You should have 1 free primary partitions to use. Type **p** and press Enter. Your partition number should default to **4** press Enter to proceed. **Press Enter** to take the default **first sector**. Type **+1G** for the **last sector** and press Enter. You should see that output that you created your 4th partition.
6. Type p to print the partitions. You should have sda1 through 4 created. Type **w** to write the changes we just made and to exit the fdisk utility. Note the output stating the kernel is not aware of these new partitions.
7. At the command prompt, type **partprobe** and press Enter to inform the kernel of the partition table changes.
8. We now have our partitions created. Next step is to create a filesystem on the partition. At the command prompt, type **mkfs –t ext4 /dev/sda4** and press Enter.
9. Create a directory named **/newmount** for use as a mount point directory underneath the / directory for mounting the new filesystem we created.
10. Copy the **/etc/hosts** to the **/newmount** directory. Provide a long listing of the **/newmount** directory to confirm the file we just copied.
11. At the command prompt, type **mount -t ext4 /dev/sda4 /newmount** and press Enter. Type the **mount** command at a command prompt utilizing the **grep** command for string: **sda4** and you will see that the new filesystem is mounted to /newmount mount point directory.
12. Provide a long listing of the **/newmount** directory and observe that you no longer see the hosts file. This is because we mounted /dev/sda4 over the /newmount directory and now see the contents of /dev/sda4. The hosts file still exists and we would have to unmount this directory to access it again.
13. In order to have this filesystem mount to the /newmount directory upon reboot of the system, we need to add this to the /etc/fstab file. Utilize the **vim** command to edit the **/etc/fstab** file. Add the following entry to the bottom of the file and write the contents and exit vim.

/dev/sda4 /newmount ext4 defaults 0 0

1. Utilize the **mount** command and the **grep** command for the string **sda4** and press Enter. You should see that /dev/sda4 is mounted. If it is not, review the previous steps to determine why.
2. At the command prompt, type **umount /newmount** and press Enter. Provide a long listing of the **/newmount** directory and you should see the hosts file again.
3. At the command prompt, type **mount -a** and press Enter. Next run the previous command from step 18. The partition should be mounted since the **-a** option reads the /etc/fstab file and mounts all of the directories defined.
4. **Provide screenshot(s) of steps 7 through 20.**

## Project 5-4

In this hands-on project, you configure the **LVM** to host two logical volumes using the space within. Next, you will format these logical volumes and mount them to the directory tree, as well as edit the /etc/fstab file to ensure that they are mounted at boot time.

1. At the terminal, become **root** by typing **su -** and press enter and provide **LNXrocks!** as the password.
2. At the command prompt, type **pvdisplay** and press Enter. You should see /dev/sda2 physical volume and the associated fedora VG (volume group).
3. At the command prompt, type **pvcreate /dev/sda3** and press Enter. We just added a new PV (physical volume) for use with LVM (Logical Volume Manager).
4. At the command prompt, type **pvdisplay** and press Enter. You should see /dev/sda3 added as a new physical volume. Note the PV size and that it is not associated with any VG.
5. At the command prompt, type **vgdisplay** and press Enter. You will see the fedora VG that was associated with /dev/sda2 in step 2.
6. At the command prompt, type **vgcreate vg00 /dev/sda3** and press Enter. We have now created a new volume group named vg00. We could have also added /dev/sda3 to the existing fedora VG by using the vgextend command.
7. At the command prompt, type **vgdisplay** and press Enter. You should see the new vg00 with 4GB of space available.
8. At the command prompt, type **lvdisplay** and press Enter. You will see the output for /, /home/ and swap LV (logical volume).
9. At the command prompt, type **lvcreate -L 2GB -n volume1 vg00** and press Enter to create a 2GB logical volume called volume1 from the vg00 volume group.
10. At the command prompt, type **lvcreate -l 100%FREE -n volume2 vg00** and press Enter to create a logical volume called volume2 from the vg00 volume group with the rest of the available space.
11. At the command prompt, type **lvdisplay** and press Enter. You should see the new volumes created in the vg00 volume group.
12. **Provide screenshot(s) of steps 2 through 11.**
13. At the command prompt, type **mkfs.ext4 /dev/vg00/volume1** and press Enter to format the volume1 logical volume using the ext4 filesystem. Next, type **mkfs.ext4 /dev/vg00/volume2** and press Enter to format the volume2 logical volume using the ext4 filesystem.
14. Create a directory named **/volume1** to create a mount point for the volume1 logical volume. Next, create the directory **/volume2** to create a mount point for the volume2 logical volume.
15. At the command prompt, type **mount -t ext4 /dev/vg00/volume1 /volume1** and press Enter to mount the volume1 logical volume to the /volume1 mount point directory. Next, type **mount -t ext4 /dev/vg00/volume2 /volume2** and press Enter to mount the volume2 logical volume to the /volume2 mount point directory.
16. Verify the directories are mounted using the **mount** command and **grep** for string: **volume**. Verify both volumes are mounted.
17. **Provide screenshot(s) of steps 13 through 16.**
18. Use **vim** to modify **/etc/fstab**. Add the following lines to the bottom of the file, as shown below, to ensure that the volume1 and volume2 logical volumes are mounted at boot time:

/dev/vg00/volume1 /volume1 ext4 defaults 0 0

/dev/vg00/volume2 /volume2 ext4 defaults 0 0

1. Save your changes and quit the vim editor.
2. At the command prompt, type **umount /volume1 /volume2** and press Enter. Next type **mount -a** to read the /etc/fstab file and mount any directories that are not mounted (volume1 and volume2).
3. Verify the directories are mounted using the **mount** command and **grep** for string: **volume**. Verify both volumes are mounted.
4. **Provide screenshot of step 14 through 20.**

## Project 5-5

In this hands-on project, you utilize features of **LVM** to reduce a logical volume and add that space to another logical volume while the system is in use. You will utilize the df and du commands to see storage availability and directory sizes. Lastly, you will update the /etc/fstab file to ensure we remove the necessary logical volume.

1. At the terminal, become **root** by typing **su -** and press enter and provide **LNXrocks!** as the password.
2. At the command prompt, type **lvdisplay** and press Enter. You should see the volume1 and volume2 LVs. Lets say you wish you utilized all 4GB for /volume1 rather than splitting it with volume2. We can remove this LV and add the remaining space to /volume1.
3. At the command prompt, type **df -h** to see the mounted filesystems along with the relevant space output. Review the full size of the volume, the space used, and the remaining space left, and each mount point directory.
4. At the command prompt, type **umount /volume2** and press Enter. Next type **lvremove /dev/vg00/volume2** which is the LV path from step 2 (this would destroy anything on this LV). Next type **vgdisplay** and review the allocated PE size and the Free PE size. We now have 2GB freed up to add to /volume1.
5. At the command prompt, type **lvextend -l +100%FREE /dev/vg00/volume1** to extend the /volume1 LV with the 2GB we have free. Next type **df -h /volume1** to review the size of the filesystem.Notice that the size is still 2GB. In order to utilize the added 2GB, we need to inform the partition that the ext4 filesystem grew. We do this with the following command: **resize2fs /dev/mapper/vg00-volume1** and then run **df -h** again to verify the new size of the filesystem is 4GB.
6. Verify the VG output and the LV output with the **vgdisplay** and **lvdisplay** commands.
7. Use **vim** to modify **/etc/fstab**. Remove the line referring to /volume2 or comment it out with a #.
8. To review the available inodes on a filesystem (we can run out of inodes just as we can run out of free space on a filesystem) run **df -i** and review the output of max number, used, and free inodes.
9. To review the file size of all the top level directories under / run the following: **du -hs /\*** and review the output.
10. **Provide screenshot of step 2 through 9.**