COMP 150 Lab 6 FileSystems

Types of files. There are 7 types of files:

1. Regular
2. Directory
3. Soft link
4. Character
5. Block
6. Named pipe (FIFO)
7. Socket

Find examples of each (you may need to make them) and **take a snapshot** (use the snipping tool) and paste the entries here. Include a) your prompt showing your name and b) the full path to the file.

Links: Hard Links, Soft Links.

In the next section we will explore hard links and soft links. You will take screen shots (using the snipping tool) of each step.

Hard Links: First we explore **hard links**

1. Make a bin directory if you have not done so already: [~Home] $ **mkdir bin**
2. Put a file name hlink in the bin. You can do this by using **cp**, **echo, nano, or cat file >hlink**
3. cd to your home directory
   $ **cd**
4. Make a hard link to *yourusername/HOME/bin/hlink*
   Do a long list of your bin and **take a snapshot** of the link containing hlink. It should show a link number of 1
5. In your home directory type **$ ln bin/hlink hlink2**
6. This will make a hard link to the file in your bin.
   ls –l your home directory to verify the file is there.
   Take a snapshot of the listing in your home directory. Note that the number of links will go up by 1.
7. Make a new directory, called *cat.* Change to that directory and make a hard link with the file in your bin. <u>You will need to use the full path name</u>. Call that file hlink3
8. Do a list and note that the number of links is now 3. **Take a snapshot of the listing of hlink3 in the directory cat.**
9. Add a line to hlink2 in your home directory. Verify that the line appears in the file bin/hlink and in the file cat/hlink3.
10. Show the inode number of each of the original file and the two hard links (use ls –i)
    Take a snapshot of the three (identical?) inode values.

11. Delete bin/hlink and cat/hlink3 with **rm**

. **Take snapshots** of

  a. **ls of bin**
  b. **ls of cat**
  c. **ls of your home directory**
  d. **the contents of hlink2 in your home directory.**

Soft Links: Now we will explore **soft links**.

1. Make a file **slink** in your bin.
2. Make a soft link with that file in your home directory.
   ln –s bin/slink slink2
3. **ls** your home directory and your bin. **Take a snapshot of these listings.**
   Note that there is no hint of a link in the listing of your bin. In your home directory there is a '->' followed by the name and location of the linked file. Also note the size of the two files. slink2 is much smaller than bin/slink. This is because slink2 only points to slink.
   How does this compare to file sizes with the hard links?

4. Add a line to slink2. Look at bin/slink. Is the line there?
   Cat the two files and show a listing of their inodes. **Take a snapshot.**

5. Now remove the file bin/slink.
   $ cat slink2 and **take a snapshot**
   Now add a new file bin/slink by the following
   $ echo ' this is an completely new file using the old name' > bin/slink
   $ cat slink2 and **take a snapshot**

Chmod: Exploring chmod. Type you answers on the sheet with your screen shots.

1. Using the 7777 way of coding permission changes write the **chmod** command that will give
   a. User read write and execute permission
   b. Group write and execute permission
   c. Other no access

   ____

   Now suppose you want these changes to apply to the directory *network.analysis* and all the files in that directory. Write the full command that will accomplish this.

   ____

2. Do the same thing but use the u,g,o,a/+_= / r w x notation. It will probably be easier if you use several commands.

3. Assign the directory network.analysis the gid of one of netan, which we can assume is one of your groups.

Mount a new filesystem

To start this part of the lab, we are going to simulate adding a new disk to your VM. VMware allows you to add additional disks to your VM just as if you were adding a new physical drive to a real system:

1. Start VMware, but don't start your VM.
2. Highlight your VM, and then click the `Hard Disk` header on the right side of the window.
3. This opens a new window. Click on the Add button at the bottom of the screen
4. Select `Hard Disck`
5. Go through the wizard, and choose the following options:
    a. `Fixed-size storage`.
    b. Leave the `Location` alone, but set the size down to `1 GB`.
6. When it finishes creating the disk, select `OK`.
7. Now, power up your VM.

At this point, you have effectively attached the new disk to your system, but there is no filesystem for the new disk. This is same as if you had plugged in a new physical drive to an actual system. In this case, the new disk is completely unpartitioned and unformatted. We therefore have to partition the drive first. To do that we have to know which device on the system represents the new drive.

It is **VERY IMPORTANT** to be sure you are working with the correct device when doing any sort of disk manipulation. For example, if you repartition your system drive instead of the new one, your VM will cease to function. So, be sure to take a snapshot of your VM before proceeding (just in case).

Recall from lab 5 that you can use `dmesg` to find information about devices and drives on your system. In particular, you have already used `dmesg` to find out exactly the information we need. Go back to the previous lab if you need help remembering how drives are represented and what you need to `grep` for in the `dmesg` output to locate the drives. Then:

1. Run `dmesg` and pipe the output to `grep` to find the lines that tell you about drives.
2. Notice that you get more lines of output than in the previous lab because there is a new device listed in the output.
3. In particular, find the line that talks about a drive that is around 1GB in size (around 1000 MB), and that is your new drive.
4. **TAKE A SCREENSHOT** of your `dmesg`/`grep` command including the command and the output that include lines describing the new drive.

You should have been able to identify the new disk as device "`sdb`". All device files are kept in the `/dev` directory, so the full path to the new disk file is `/dev/sdb`. Now we actually need to partition the drive. For our purposes we will simply create one partition on the drive that uses all the space. We will use the `fdisk` (format disk) program to do so:

1. Run: `sudo /sbin/fdisk /dev/sdb`
2. This starts the `fdisk` program in interactive mode to format the `/dev/sdb` disk.
3. In `fdisk`, you can enter an "`m`" to print the help menu. There are quite a few options, but we'll just do the most basic partitioning here.
4. Enter an "`n`" to create a new partition.
5. You will be prompted to enter an "`e`" to add an extended partition or a "`p`" to add a primary partition. We are only creating one partition, so enter "`p`" for a primary partition.
6. You will be prompted for a partition number. Enter "`1`" (one) to create the first partition.
7. Next, you will be prompted for the `First cylinder`. Just hit enter to use the default value, and then do so again to accept the default value for `Last cylinder`.
8. Now, enter "`p`" to print the partition information for your drive.
9. **TAKE A SCREENSHOT** of the output including the information about the disk and the single partition on the disk. Note: the output should start with "`Disk /dev/sdb`". If it doesn't then **STOP WHAT YOU ARE DOING**, enter "`q`" to quit `fdisk`, and rerun the `fdisk` command, being sure you specify `/dev/sdb` as the argument.
10. Finally, enter "`w`" to write the new partition table to the drive. After a few moments you will be put back into your shell.

So far, we've added the drive to your VM, found the device that represents that drive, and partitioned the drive. The next step is to create a filesystem on the new drive so that you can use it within your VM as part of the normal tree structure. There are a number of different types of filesystems that you can use. One of the most common is the ext3 filesystem. Each partition on each drive must have a filesystem on it. Individual partitions are therefore also represented inside your VM by adding a digit to the end of the disk device. The disk is `/dev/sdb`, so the first partition is `/dev/sdb1`. To create an ext3 filesystem on your new partition:

1. Run: `sudo /sbin/mke2fs -j /dev/sdb1`
2. `mke2fs` is the program to make ext2 and ext3 filesystems, and the `-j` option indicates that it should use a journaling structure which is the difference between ext2 and ext3.

The new disk is now finally ready to be used. Every partition on every disk is *mounted* somewhere in the file hierarchy. For example, on your VM `/dev/sda1` is the boot partition and it is mounted inside your VM on the `/boot` directory. You can use the `df` command to print out the current filesystems, where they are mounted, and how much space each filesystem has:

1. Run: `df -h`

2. The `-h` option makes the numbers more human-readable.

So, we need to add our new `/dev/sdb1` filesystem to our system somewhere. Let's use the new filesystem to store all our information for this lab assignment. We need to make a directory and then mount the new filesystem on that directory:

1. Run: `sudo mkdir /lab6`
2. This creates the `/lab6` directory at the top of the file hierarchy.
3. Then, tell the OS to use `/dev/sdb1` as the filesystem for `/lab6`. Run: `sudo mount /dev/sdb1 /lab6`
4. To verify that it worked, run: `df -h`
5. **TAKE A SCREENSHOT** of the output of `df -h`. You should see that `/dev/sdb1` is mounted on `/lab6.`

At this point, the new disk partition is finally usable on the system. Any files and directories placed in the `/lab6` directory will be stored on `/dev/sdb1`, not your original disk. Unfortunately, this is not permanent. Running the mount command by hand only mounts the partition once. At the end of this lab you will learn how to ensure that it will be mounted automatically every time the system boots.

Because we created the `/lab6` directory as root, it is owned and group-owned by root. We don't want to have to do everything with `sudo` in that directory, so it would be more convenient if it were owned by our normal user. Use the `chown` and `chgrp` commands to change ownership:

1. To change the owner, run: `sudo chown USERNAME /lab6`
2. Obviously, replace `USERNAME` with your actual username.
3. To change the group, run: `sudo chgrp USERNAME /lab6`
4. Again, replace `USERNAME` with your username (which is also your default group).
5. To verify it worked, run: `ls -ld /lab6`
6. You should see that `/lab6` has owner and group set to your account.

It would be nice to see this directory in our home directory. The solution is to use a soft link on our home area to make it appear as if we have a dedicated lab6 directory when in fact we are simply going to point our `lab6` directory to the `/lab6` one. To do that:

1. Make sure you are in your home area directory (`/home/USERNAME`).
2. Run: `ln -s /lab6 lab6`
3. This creates a soft link named `lab6` in your home directory that points to the `/lab6` partition.
4. To verify that it worked, run: `ls -l lab6`
5. You should see that `lab6` is soft link (file type "`l`" (ell)) that points to `/lab6`.

6. **TAKE A SCREENSHOT** of the output of the above `ls` command, including your command prompt with the command.

Finally to guarantee that your `/lab6` partition will be mounted automatically every time your VM boots up, we need to edit a configuration file that stores information about every partition. This file is `/etc/fstab` (file system table). Each line of this file tells the OS about one filesystem. Each line has 6 fields separated by whitespace (usually tabs, but it doesn't matter). Here are the descriptions of each field, left-to-right:

1. Device name.
2. Location to mount the device.
3. Filesystem type.
4. Mounting options.
5. The last two fields deal with when the filesystem is checked during system boot.

You can get more information by looking at the man page for `fstab` (`man fstab`). You need edit the file and add a new line so that your system knows where to mount the new partition. Look at the other lines in the file, the man page, and other resources online to determine exactly how to do that. Once you think you've got it:

1. Reboot your VM.
2. When it comes back up, log in and check to see if `/lab6` was mounted automatically (you can check with the `df` command again).
3. If it didn't work keep trying until you get it.
4. Then once it does work, run: `cat /etc/fstab`
5. **TAKE A SCREENSHOT** of the output of the cat command including your prompt and all like normal.