This is Worksheet and Assignment 12

This is a combined Worksheet and Assignment.. Quizzes and tests may refer to work done in this Worksheet and Assignment; save your answers. You will use a checking program at the end of the assignment to verify the correctness of your work. You must upload the check program results before the due date.

Before you get started - REMEMBER TO READ ALL THE WORDS

You must have your own Fedora 12 virtual machine (with **root** permissions) running to do this lab. You cannot do this assignment on the Course Linux Server because you do not have **root** permissions on that machine.

Disks, Partitions, and File Systems

1 Commands, topics, and features covered

Use the on-line help (**man** command) for the commands listed below for more information.

- **`df`** – show mounted partitions and amount of used/free space
- **`du`** – recursively display disk usage in directories
- **`eject`** – unmount and eject a CDROM
- **`fdisk`** – to display, create, delete, and manage partitions; option **`-l`** is very useful
- **`file`** – determine what kind of thing a pathname is. Can show partition file system types using option **`-s`** and will follow (dereference) symbolic links using option **`-L`** (upper case)
- **`mkfs`** – create a file system on a device, usually a hard disk partition.
- **`mkswap`** – initialize a partition for use as a Linux swap partition.
- **`mount`** – mount a file system existing on some device into the main file system tree.
- **`swapon`** – tell the Linux kernel to use an initialized swap partition.
- **`umount`** – detach (unmount) a mounted file system (e.g. that was mounted with **`mount`**).

2 Correct user, command lines, and command output

- For this lab, you will need to obtain a **root** (super-user) prompt so that you'll have the required **privilege** level to run the **file system maintenance** commands. The **root** account is the only account with sufficient **permissions** to use these file system utilities. All of this lab is done as the **root** user. Your shell prompt will tell you if you are the **root** user by including a **#** character.
- Some answer blanks require you to enter **command lines**. Do **not** include the shell **prompt** with your command lines. Give only the part of the command line that you would type yourself.
- Make sure you know the difference between a command **line** (which is what you type into the shell) and command **output** (which is what the command displays on your screen).

3 Backup and Recovery

1. Take a snapshot of your virtual machine before you begin this lab (and perhaps before you begin each new section) so that you can recover if needed.

4 Creating a new VMware virtual hard disk

1. **Shut down** your Fedora machine so it is **powered off** and not just suspended.
2. With your Fedora machine still powered off, add to it a VMware virtual **1GB** hard drive as shown in the Course Notes ("***Create a New VMware Virtual Hard Disk***"). Create it exactly **1 GB** in size.
3. Reboot (power on) your Fedora virtual machine. Check - the new disk should appear as **`/dev/sdb`**:
   a) **`ls -l /dev/sd*`** *(make sure the **sdb** disk is visible)*
   b) **`cat /proc/partitions`** *(make sure the **sdb** disk is visible)*

4. Make sure you **only** change things on this new **`sdb`** disk in this lab! The **`sda`** disk is your Linux **ROOT** disk; if you damage it you will need to recover back to your snapshot. You do have a **snapshot**, right?

5 Viewing Existing Partitions using fdisk

1. First, you must have added a new 1 GB hard drive in **VMware** and rebooted, as described above. **DO NOT USE `/dev/sda` WHICH IS YOUR MAIN FEDORA INSTALLATION DRIVE!**
2. Make sure you have **root (**super-user) privileges.
3. At the **root** shell prompt, type "**`cat /proc/partitions`**" and verify that you have an **`sdb`** partition of 1GB (approx **1048576** blocks) and that you do not have any **`sdb1`** or **`sdb2`** partitions. If you have any **`sdb1`** or **`sdb2`** or other **`sdbN`** partitions, call over your instructor before you continue!
4. To manage partitions, use the **`fdisk`** command. The syntax of the **`fdisk`** command is:
   - **`fdisk [`*`options`*`] `*`device_name`*
     A useful option (try it!) is "**`-l`**", e.g.: **`fdisk -l ; fdisk -l /dev/sdb`**
5. At the **root** shell prompt, type (make sure you use the disk device **`sdb`**!): **`fdisk /dev/sdb`**
   Make sure you use **`sdb`**. You will get several warning messages about a "**new DOS disklabel**" and an **invalid flag**. This is **normal**, since the new virtual disk has not been **initialized** yet. Continue:
6. The **`fdisk`** utility is now prompting you for input with a different **prompt**: **`Command (m for help):`**
   This prompt is the **`fdisk`** utility **prompt**. As it says, type **`m`** for a list of menu options for **`fdisk`**.
7. Read the list of **Command action** commands and record the (one-character) **`fdisk`** command to:

   a) display/list/**print** the **table** of all **partitions**: _____
   b) create/**add** a **new partition**: _____
   c) remove/**delete** a **partition**: _____
   d) show/display/**list partition types** (system ids): _____
   e) **change** a **partition's** type (**system id**): _____
   f) save/**write** partition **table to disk** (**and exit**): _____
   g) exit/**quit `fdisk` without saving changes**: _____

8. Select the **`fdisk`** option that **lists** the **partition types**. (Partition types are also called "system identifiers".) Record the type (system id) number of the following partition types, making sure you read them correctly:

   a) "**`Linux`**": _____
   b) "**`Linux swap / So`**": _____ *(short for* **`Linux swap / Solaris`***)*

6 Creating Partitions using fdisk

Use the correct commands in the **`fdisk`** utility to create the following **six** new partitions on your **`sdb`** disk. Note that **`fdisk`** will adjust the size of each partition slightly to fit the DOS partition table disk geometry; don't be alarmed that the size that **`fdisk`** creates and displays to you isn't exactly the size you asked for. Display the partition table **after each change** to confirm that things are working.

1. Create a **primary** partition of **200 MB**. The type (system id) will default to type "**Linux**".
2. Create another **primary** partition of **100 MB**. Leave the type (Linux) as default.
3. Create an **extended** partition large enough to host the following three **logical** partitions that total **600MB**. You must make the extended partition large enough to hold **all three** logical partitions described in the next step. NOTE: You cannot create it *exactly* **600MB**. You need to make it a bit **larger** to accommodate the logical partition information. Experiment to see how much "a bit" means. The end of the extended partition must be **less than** cylinder 130. (130 is the end of the disk.)
4. **Create** these three **logical** partitions inside the **extended** partition that you created in the previous step:
   a) The size of the **first logical** partition is **200MB**. Leave the partition type set as "**Linux**".
   b) The size of the **second logical** partition is **100 MB**. Change the partition type to "**Linux swap**".
   c) The size of the **third logical** partition is **300MB**. Change the partition type to "**HPFS/NTFS**".

If you **run out of space** creating the logical partitions inside the extended partition, you can **delete the partitions** and start over as many times as needed. (You can also start over by exiting **`fdisk`** without saving/writing any of your partition changes.) Make the extended partition just big enough, no bigger.

5. Did you remember to set the correct partition **types** (system id) on each of the six partitions?
6. **Save** your changes (six partitions) to disk and exit **`fdisk`**. You will return to your **root** shell prompt.
7. Verify the creation of six new **`sdb`** partitions using both command lines "**`ls -l /dev/sd*`**" and "**`cat /proc/partitions`**" again. You should have exactly **six** partitions on this second disk.
8. Use **`fdisk`** to show the partition table, listing the **six** new partitions that you just created (**`sdb1, sdb2, sdb3, sdb5, sdb6, sdb7`**) and answer these questions from the **`fdisk`** output:

a) Give the **full** *device name, blocks, a*nd *ID* of the two **Primary** Partitions on the new 1.0GB Drive:	_____ , _____ , ___ _____ , _____ , ___
b) Give the **full** *device name, blocks,* and *ID* of the one **Extended** partition on the new 1.0GB Drive:	_____ , _____ , ___
c) *Give* the **full** *device names, blocks,* and *ID* of the three **Logical** partitions on the new 1.0GB Drive:	_____ , _____ , ___ _____ , _____ , ___ _____ , _____ , ___
d) How many more **Primary** Partitions could you create on the new 1.0GB Drive, and explain **Why**:	_
e) If you had enough disk space, how many more **Logical** partitions could you create on the new 1.0GB Drive, and explain **Why**:	_

7 Deleting a logical Partition using `fdisk`

1. You have **six** partitions on your second hard disk. (Make sure this is true before continuing!) Create a **VMware** backup **snapshot** of this virtual machine, so you can return here if things go wrong.
2. Use **`fdisk`** to delete the first **logical** (not primary) partition that has size **200MB**.
3. How has the partition numbering changed for the five remaining partitions, after the deletion?

   _____
   _____
   _____

4. **Save** (write) the partition changes (now only **five** partitions) to disk and exit **`fdisk`**.
5. Re-enter **`fdisk`** and (re-)create a new logical partition on the second disk, but do ***NOT*** save your work. How does the partition numbering change when you recreate the partition you just deleted?

   _____

6. Are the three logical partitions again in ascending **Start** block order on the disk? _____
7. Switch to the Extra/Expert Mode menu. What Expert command letter **fixes** partition order? _____
8. Use the Expert Command to "**fix**" the partition order. Return to the main menu (non-Expert mode) and list the partitions now. Are the logical partitions now in ascending Start block order? _____
9. Exit **`fdisk`** without saving the sixth partition or the fixed partition order. Do not save your work.
10. You still have **five** remaining partitions, the same that you saved in step 4 above. Check these five partitions in the next step.

8 Use your 1 GB disk and five partitions from the previous steps

To continue with the next sections of this lab, you must have successfully created these **five** (remaining) partitions on the 1GB disk. Verify that they have exactly the same **Device** numbers, exactly the same **Id** and **System**, *approximately* the same **Start** and **End**, and *approximately* the same number of **Blocks**.

```
# fdisk -l /dev/sdb
   Device Boot   Start      End      Blocks   Id  System
/dev/sdb1            1       26      208813+  83  Linux
/dev/sdb2           27       40      112455   83  Linux
/dev/sdb3           41      119      634567+   5  Extended
/dev/sdb5           67       80      112423+  82  Linux swap / Solaris
/dev/sdb6           81      119      313236    7  HPFS/NTFS
```

Do not proceed until you have the above five partitions created. The sizes may vary *slightly*. The System ID must match. The **end** of the **Extended** partition must be **less than** cylinder 130. (130 is the end of the disk). All file system commands in the next part of this lab referring to a hard disk will **use one of the above partitions**.

1. Create a **back-up** of your **5**-partition system now using the VMware **snapshot** function. You can return to this back-up if anything goes wrong during this lab, or if you want to repeat the lab for practice.
2. Use **`fdisk`** to delete all the partitions and then re-create them again, **without** writing out your changes. Recreate the above five partitions, but don't save your work. Would you remember how to do this when asked to demonstrate it at a job interview? When installing a new disk? Practice!

9 The useful `file` command - what is that thing?

1. The Unix/Linux **`file`** command is very useful for identifying things in the file system, such as directories, programs, images, files, and special files such as disk partitions. Run the **`file`** command and give it as a single argument the full **absolute** path name of the **device file** of the **Extended** partition on your new 1.0 GB disk. Record the command line and the output here:

   _____
   _____

2. Repeat the above command line and add the option that makes **`file`** examine ***special files*** (e.g. block devices) and give the new command line and new extended (x86 boot sector) output here:

   _____
   _____
   _____
   _____

3. Use **`file`** (and the correct special file option) on the Linux **swap** partition device file and record the (not very useful) output here. (You will note that an empty new partition isn't very interesting.)

   _____

4. Use **`file`** (using the correct special file option) on the **first** partition of the **first** disk (**not** your new disk) and record the more useful information printed about the type of file system stored in the partition. Record that command line (using the absolute path of the device file of the first partition of the first disk) and record the output (about an **`ext4`** filesystem and journal) here:

   _____
   _____
   _____

5. Try using the **`file`** command on various other pathnames and see what it says. Try your **ODT** files, the password file, the **`/bin/ls`** program, the **`vmlinuz`** file under **`/boot`**. Lastly, use **`file`** on **`/etc/favicon.png`** and then add the option to **dereference** (follow) the symbolic link and give the full command line (use the full absolute pathname) and record the (PNG image) output here:

   _____
   _____

10 Creating file systems using `mkfs`

Make sure you have the five partitions from the previous steps available. To create a file system on an *existing* **partition** or device, use one of the **`mkfs`** commands. (The partition must already exist!) Syntax:

- **`mkfs -t filesystem_type device_name`**

  For example: **`mkfs -t ext3 /dev/sdb9`**
  The *device_name* is usually the name of an *existing* **partition**, ***not*** the name of the whole disk.

1. Find the device names on your 1GB disk of the **two** partitions that have partition type (System ID) of "**Linux**". Create a Linux type **ext3** file system on the *first* Linux partition. Record the exact command line you used, and the **output** of "**`file -s device_name`**" for the ***device name*** you used:

   _____
   _____

2. Create an **ext4** file system on the *second* partition that has type "**Linux**". Record the exact command line you used, and the **output** of "**`file -s device_name`**" for the ***device name*** you used:

   _____
   _____

3. Create an **ntfs** file system on the only "**HPFS/NTFS**" partition. Record the exact command line you used, and the **output** of "**`file -s device_name`**" for the ***device name*** you used:

   _____
   _____
   _____
   _____

4. Take a VMware back-up snapshot now and name it something like "**done_mkfs**".

11 Mounting & Unmounting a Linux File system using `mount`

Creating a **file system** does not automatically connect that file system to your Linux system. To connect an existing file system to your Linux file system so that you can access it, use the **`mount`** command. Syntax:

- **`mount [-t filesystem_type] device_name mount_point`**

  For example: **`mount -t ext4 /dev/sdb9 /mnt/foobar`**
  The *device_name* is usually the name of an existing **partition**, ***not*** the name of the whole disk.
  The *mount_point* must be an already existing directory. (The directory is usually empty, since the act of mounting on it will "cover up" or hide its contents until the file system is unmounted again.)
  Let me repeat that: **THE MOUNT POINT MUST ALREADY BE AN EXISTING DIRECTORY**. If the mount point directory doesn't exist yet, create it first.

- While completing this next section of the lab, use **`mount`** without any arguments frequently to confirm your actions and ensure that mounting and unmounting were successful. Use this command to see what is mounted and what type of thing each is. Confirm each action that you do!

1. List all the partitions on your system using: **`fdisk -l`**
   You may see two error messages about **`/dev/dm-0`** and **`/dev/dm-1`** that you can ignore.
   Record the sizes of Disk **`/dev/sda`** and Disk **`/dev/sdb`**: _____
2. List all the mounted file systems using: **`mount`**
   What **partition device name** is mounted on the **`/boot`** directory? _____
   What **type** of file system is the **`/boot`** file system? _____

11.1 Mount three file systems

1. Create three directories named **`/mnt/ext3`**, **`/mnt/ext4`**, and **`/mnt/ntfs`** to use as mount points for the three file systems you created at the end of Section 3. (You can reduce your typing by using one command with brace brackets to express the three pathnames.) After creating them, record the output of:

   `ls -id /mnt/ext3 /mnt/ext4 /mnt/ntfs`

   _____

   (You can also reduce your typing by using the brace brackets on the above command line!)
2. Use three **`mount`** commands to mount the three file systems you created previously, each mounted on its own directory from above. (Recall that each file system was created with a particular type. Match the file system type with the directory name.) Record the three **`mount`** commands you used to do this here:

   _____

   _____

   _____

   (Sorry, you can't use brace brackets here - you have to use three separate command lines.)
3. Use **`mount`** without any arguments to verify that you have three new mounted file systems. Each file system type should match the directory name on which it is mounted. Each file system should be mounted only **once**. (If you have duplicate entries, unmount them using the **`umount`** command.)
4. Record again the (different!) output of: **`ls -id /mnt/ext3 /mnt/ext4 /mnt/ntfs`**

   _____

   (You can again reduce your typing by using the brace brackets on the above command line!)
5. Leave the three file systems mounted. Take a VMware back-up Snapshot now and name it something like "**done_3mount**".

11.2 Disk Free - df

1. The **`df`** command shows information about mounted file systems, including the amount of disk space used and disk space still available. A useful option is **`-h`** that shows output in "human-readable" form. Give the human-readable amount of disk space **Used** and **Available** on the **`/boot`** file system:

   _____

2. If you add up the **Used** plus **Available** disk space on an **NTFS** file system, it equals the **Size** of the file system. If you add up **Used**+**Available** on a **Linux** file system, it is usually about 5% smaller than the **Size** of the file system. Why? (Hint: Search for: **`why linux reserved space`**)

   _____

   _____

   _____

11.3 Disk Usage - du

1. The **`du`** command walks the file system and recursively shows the disk usage in every directory under a directory. With the **`-s`** option, only the **summary** of the disk usage is shown. With the **`-h`** option, the output is given in "human-readable" form (similar to the same option to **`df`**). Use **`du`** to show a **summary** of the **human-readable** amount of disk space on the **`/`** (ROOT) file system and record the command line used and its output here (The command will take some time to finish!):

   _____

   _____

2. Compare the speed of running the above **`du`** command (which has to walk the entire directory tree) against the speed of running **`df`** in the previous section. Also compare the **Used** output of **`df`** with the summary output of **`du`**. Why does **`df`** say that more disk space is used than **`du`**?

   _____

   _____

12 Preparing a Swap Partition on a hard drive using `mkswap` and `swapon`

Linux systems usually have an entire partition devoted to **swap area** to enable **virtual memory**. The partition must be initialized with **`mkswap`** and then given to the Linux kernel using **`swapon`**. Syntax:

- **`mkswap`** ***`[options]`*** ***`device_name`***
- **`swapon`** ***`[options]`*** ***`[device_name]`***

The ***device_name*** is the **pathname** (usually **absolute**) of the existing swap partition that will be used.

1. Enter **three** commands: Initialize the **Linux swap** partition on your new 1 GB disk; tell the kernel to use it; display the active swap partitions. Record the **three** command lines used to do these **three** actions:

   _____
   _____
   _____

2. Record the **output** of "**`file -s`** ***`device_name`***" for the ***device name*** you used:

   _____

3. Leave the new swap area connected. Take a VMware back-up Snapshot now and name it something like "**done_swap**".

13 Creating a Linux File system on a virtual Floppy disk and mounting/unmounting/ejecting it

13.1 Creating a VMware virtual floppy disk image file

To do some exercises, you first need to create a virtual floppy disk image file and connect it to VMware:

1. In VMware, click **VM → Settings**
2. Select **Floppy** in the **Virtual Machine Settings** window
3. Select **Use a floppy image**
4. Select the **Create...** button to create a new floppy disk image file. Name the file: **`linux.flp`** and select **Open** to create the file and attach it to your Fedora virtual machine. (Remember where the file is on your disk, so that you can find it again later!)
5. Back in **Virtual Machine Settings**, ensure that the floppy is **Connected**: make sure the **Device Status Connected** checkbox is checked.
6. ***DO NOT*** check the **Connect at power on** box or your Fedora machine will not boot.
7. Click **Save** to save the new **Virtual Machine Settings**.
8. At your Fedora super-user command line, verify that the floppy disk image file is connected by using:

   ```
   fdisk -l /dev/fd0
   ```

   The first line of output should read: **`Disk /dev/fd0: 1 MB, 1474560 bytes`**

13.2 Creating a Linux file system on a virtual floppy disk and mounting/unmounting/ejecting it

- You must have a virtual floppy disk image file connected to your Fedora machine to do this section. Use the **`fdisk`** command to verify that the floppy disk image file is connected to **`/dev/fd0`**
- Floppy disks appear in Linux with device names such as **`/dev/fd0`** (floppy disk zero) and are used **without** any partition numbers. Do **not** use **`fdisk`** to create partitions on a floppy disk. You create file systems directly on a floppy disk using **`mkfs`** by inserting a floppy disk (or connecting a virtual disk image file) and then using its Linux device name with no partition name, e.g. **`/dev/fd0`**
- Floppy disks are small enough that the default **ext2** file system type is best; do not create any journalling file systems on a floppy disk. Always use the **ext2** file system type for a Linux-style floppy disk.

1. Create a type **ext2** (the default) file system on your floppy disk. Record the exact command line you used, and the **output** of "**`file -s device_name`**" for the ***device name*** you used:

   _____

   _____

2. Mount the floppy disk file system on **`/mnt/linux`** after creating it. Record the command used:

   _____

3. Use **`mount`** without any arguments to verify that you have a new **ext2** file system mounted and copy the single line of **output** related to the only floppy disk device here:

   _____

4. Copy a file to the floppy disk: **`cp -a /etc/passwd /mnt/linux/passwd`**
5. Use this command to see the Linux directory contents: **`ls -lia /mnt/linux`**
   Copy the two lines of output for the **`lost+found`** directory and the **`passwd`** file here:

   _____

   _____

6. To "eject" a virtual floppy disk, first **un-mount** it from Linux and then **un-check** (disconnect) the VMware **Device Status Connected** checkbox. Do **not** eject a floppy disk that is still **mounted** in your Fedora machine! Unmount it **first**, and only then disconnect it. Do this now: un-**mount** the floppy and then eject it. Verify that "**`fdisk -l device_name`**" gives no output for the unmounted floppy device and copy here the **output** of "**`file -s device_name`**" for the ***device name*** you used:

   _____

14 Creating a Microsoft FAT file system on a virtual floppy disk and mounting it

14.1 Creating a VMware virtual floppy disk image file

- Linux can mount many types of file systems, including Microsoft file systems. Microsoft machines can only mount Microsoft file systems. Let's create a Microsoft-format diskette and mount it in Linux.

1. **Create** and attach to Fedora a **new** virtual floppy disk image file named: **`windows.flp`**
   This new image file you create will **replace** your Linux floppy image on the **virtual** floppy disk drive. Make sure you have unmounted your Linux floppy image file before you create and switch images!
2. Verify that the floppy disk image file is connected as you did in an earlier section above.
   The first line of output should read: **`Disk /dev/fd0: 1 MB, 1474560 bytes`**

14.2 Creating a Microsoft FAT file system and mounting it

1. Create a type **vfat** file system on your Windows floppy disk. Record the exact command line you used, and the **output** of "**`file -s device_name`**" for the ***device name*** you used:

   _____

   _____

2. Mount the floppy disk file system on **`/mnt/vfat`** after creating it. Record the command used:

   _____

3. Use **`mount`** without any arguments to verify that you have a new **vfat** file system mounted and copy the single line of **output** related to the only floppy disk device here:

   _____

4. Copy a file to the diskette: **`cp -a /etc/passwd /mnt/vfat/passwd`**
5. Use this command to see the Linux directory contents: **`ls -lia /mnt/vfat`**
   Copy the one line of output for the new copy of the **`passwd`** file here:

   _____

6. Why is there no **`lost+found`** directory shown in the output above?

   _____

7. Leave the **vfat** floppy mounted. Take a VMware back-up Snapshot now and name it "**done_vfat**".

15 Mounting & unmounting a CDROM file system

- **IMPORTANT NOTE:** *All removable media, such as floppies, USB, external hard drives, and CD-ROMs, must not be removed from the drive, disconnected, or swapped with another disk while **mounted**. Removing or changing media that is **mounted** causes the data in memory and the data on the device to be out of synchronization. This can result in data loss and other severe errors. Whenever you want to switch or remove a floppy, USB, or CD-ROM, **unmount** it first, before removing or changing the medium. Always unmount first!*

- You can do this section using either **Method A** a virtual CDROM (an **`*.iso`** file) or **Method B** a real, physical, data (not audio!) CDROM (if you have one). Any data CDROM will do, even a Windows install CDROM. Choose either **Method A** or **B** below (or try both, for practice):

1. **Method A - Virtual CDROM:** In the VMware settings for your Fedora VM, under **CD/DVD (IDE)** under **Device Status**, select the checkbox beside "**Connected**", and in the same window, under "**Connection**", select "**Use ISO image**". Use the **Browse** button and navigate to any directory containing **`*.iso`** files. Even Windows **`*.iso`** files will work. (You can find **`*.iso`** files in your VMware software directory, e.g. **`/usr/lib/vmware/isoimages/`**) Pick any one of the **`*.iso`** files you find. **Save** the settings.
1. **Method B - Physical CDROM:** Connecting the physical CD/DVD drive to Linux through VMware is much like selecting a floppy disk image file. In the VMware settings for your Fedora VM, under **CD/DVD (IDE)** under **Device Status**, select the checkbox beside "**Connected**", and in the same window, under "**Connection**", select "**Use a physical drive**", with **Device** set to "**Auto Detect**" (or try the other setting, too). **Save** the settings. Now put a **data** CD-ROM disk in the CDROM drive (e.g. your Fedora disk, or a lab boot disk, or a Windows Install disk). Note: an Audio CD will **not** work. Use a **data** disk.
2. Fedora is a desktop (not a server) O/S and it may "automount" some disks (especially CDROMs) for you. You will need to **unmount** them before you can mount them manually using shell commands below. Use the **`mount`** command to see if the CDROM device **`/dev/sr0`** has been mounted by Fedora and **unmount** the device if it has been automounted. (Hint: Rather than searching with your eyes in the output of **`mount`** for the **`sr0`** device, use a pipeline to search for it. Unmount it if you see it.)
3. Make sure no CD is mounted on your Fedora desktop before you continue; unmount **`/dev/sr0`**
4. Linux often creates many symbolic links in the **`/dev`** directory that point to the **`sr0`** device name. Use a two-command pipeline to show **all** these symbolic links (more than one) and record the **pipeline** here (Hint: One command to show everything under **`/dev`** piped into a command that finds strings matching the device name. Always make the computer do your searching work for you!):

   _____

5. Verify that Fedora can access your physical CD: **`fdisk -l *device_name*`**
   You should see some warning messages, disk size information, and partition information.
6. Give the **output** of "**`file -s *device_name*`**" for the ***device name*** of the CDROM. (If the output says only "writable, no read permission", you forgot to "Connect" the CDROM to the virtual machine.)

   _____

7. Mount the CDROM file system (using any of its device names) on the directory **`/mnt/cdrom`** after creating it. Record the command used:

   _____

8. Use **`mount`** without any arguments to verify that you have a new file system mounted from the CDROM device and copy the single line of **output** related to the only CDROM device here:

   _____

9. Try (as **root**) to copy a file to the CDROM: **`cp -a /etc/passwd /mnt/cdrom/passwd`**
   and record the error message here:

   _____

10. Leave the CDROM mounted. Take a VMware back-up Snapshot now and name it "**done_cdrom**".

16 Lab Check and Upload - `assignment12marks.txt`

This is the section that tests and marks the work you did above. The Assignment Check program below will do the checking to make sure you got things right. Do you have three partitions from the second disk, one floppy disk, and one CDROM mounted? Is the new swap area from the second disk configured and in use?

Following a method similar to previous labs, download and run the **`assignment12check`** program. You can run it over and over until you are happy with the output. When you are ready, rename the **`assignment12marks.txt`** file to be **`assignment12.txt`** and upload it from Fedora to Blackboard. Do not try to edit or print this file - it contains hundreds of lines of status output!

17 Command: `eject` *`device_name`*

The **`eject`** command will unmount a given CDROM file system if its mounted, and eject (disconnect) the media from the drive. On our Fedora systems, the default device is the **cdrom** device, so just "**`eject`**" with no arguments (with no device name) will also work.

18 Ejecting and unmounting everything

After marking your assignment using the above check program, give the command lines to do these operations (and do them!):

1. Eject the CDROM. _____
2. Unmount the VFAT file system. _____
3. Unmount the NTFS file system. _____
4. Unmount the EXT4 file system. _____
5. Unmount the EXT3 file system. _____

19 Practice these commands

Go back to your earlier snapshots and **REPEAT** these exercises as often as necessary, until you can do this without looking at the lab instructions. Without looking at the instructions, can you do the following:

- Create a new VMware virtual disk and connect it to Fedora (e.g. create **`/dev/sdc`**).
- Create primary, extended, and logical partitions on the new disk.
- Set partition types.
- Create any type of file system inside any partition.
- Mount and unmount any file system

Can you do all the above operations without reference to any help files? Practice! A job interview may involve having you partition a disk and install Linux. Try to look like you know what you are doing!