This is Lab Worksheet 13 - not an Assignment

This Lab Worksheet contains some practical examples that will prepare you to complete your Assignments. You do not have to hand in this Lab Worksheet. Make sure you complete the separate Assignments on time. Quizzes and tests may refer to work done in this Lab Worksheet; save your answers.

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.

Boot Process and GRUB

1 Commands, topics, and features covered

Use the on-line help (**man** command) for the commands listed below for more information.
The Class Notes also cover the use of these commands and the syntax of GRUB pathnames.

- **`df`** – display space available (optionally inodes available) on all mounted file systems
- **`grub`** – **GRand Unified Bootloader** (legacy version 0.9x)
  The man page is useless. See this instead: **http://www.dedoimedo.com/computers/grub.html**
- **`mount`** – display a list of all mounted file systems
- **`uname`** – display system name, kernel release and version number, machine, processor, and O/S type
- **`/etc/inittab`** – documentation on Run Levels; default Run Level
- **`/boot/grub/`** – directory containing GRUB configuration files

2 Use your host O/S to read this lab - do snapshot - use root - second disk

- Since this lab requires you to reboot your Linux virtual machine multiple times, you will find it easier to open this lab document in **Open Office** in your Windows host O/S, not inside Fedora where you will have to keep closing it to reboot (and can't read it while you are doing so). Use **OO** not **MSWord**!
- Take a **snapshot** of your virtual machine before you begin this lab so that you can **recover** if needed.
- Changing the boot process **requires** super-user (**root**) privileges. Use **root** throughout this lab.
- One question in this lab uses the second hard disk attached to your virtual machine in an earlier lab. If you do not have this second disk, you may go back to the previous lab and create one. Verify that the output of **`fdisk -l /dev/sdb`** shows the five partitions from the previous labs (1,2,3,5,6).

3 Boot Partition device

The Linux kernel and associated files are stored under directory **`/boot`** on Linux. The **`/boot`** directory may be a subdirectory of the **ROOT** file system, or it may be its own mounted **BOOT** partition and file system. Fedora 12 uses a **separate** BOOT partition, which means GRUB pathnames and Linux pathnames differ.

1. Give a **two-command bash shell pipeline** that will first display the list of all mounted file systems and then pipe it into a command that displays **only** the line(s) containing the string **`'/boot'`** :

   ____

2. Give the one line of **output** of the above two-command bash shell pipeline:

   ____

3. Give the equivalent GRUB partition name (in parentheses) of the above mounted **`/boot`** file system:

   ____

4 Kernel Version Number

Your Linux kernel has a **version** number, as in "What **version** of the kernel are you running?". Unfortunately, the command that *displays* the kernel **version** number calls it a kernel **release** number, because it uses the name **version** to stand for the kernel *compile date*. Know that **version** means **release** in the documentation for this command. When asked for the "kernel **version**", interpret that to mean the "kernel **release**" number.

1. Give the **command line** that displays **only** the version (release) of the Linux kernel are you running:
   ____
2. Record your kernel **version** number (about 26 characters; begins with the digit 2):
   ____

5 GRUB Shell Commands

GRUB is a shell-like utility that has many built-in commands. See the Class Notes for help.

1. Do GRUB pathnames include the `/boot` directory prefix on Fedora 12? __________
2. Why? ____

Entering and Leaving the GRUB Shell

3. Start the command-line GRUB shell by typing `grub` at the **root** (super-user) BASH shell command prompt. Type `help` at the `grub>` prompt to see a partial list of GRUB shell commands. What GRUB shell command exits the GRUB shell and returns you to the BASH prompt? __________

Finding pathnames

4. Start GRUB. Enter the following GRUB shell command to find on which disk partition your **kernel** pathname is located. Where you see `X`s, **replace** those with information from your kernel **version** number. If you get "*File not found*", check your typing and permissions and **try again** until you find it:

```
grub> find /vmlinuz-2.X.XX.X-XXX.fc12.i686.PAE
```

5. Record the actual pathname you used: ______________________________
6. Give the GRUB **partition** output of the above GRUB shell **find** command: ______________
7. Look for the pathname `/grub/grub.conf` and record its GRUB **partition**: ______________

Displaying Text Files

8. Use the GRUB shell `cat` command to display the contents of file `/grub/device.map` and record the **last** line of the file here: ______________________________
   Hint: You will need to **find** which **partition** the file is on, and **prefix** the file name with that partition.

Setting a default (root) disk and partition

9. The GRUB shell `root` command can set a disk **partition** device **prefix** that will be used in front of pathnames so that you don't have to type the partition name at the start of GRUB pathnames. Type the `root` command **without** any arguments and record the **current** default **device** name prefix (a floppy disk - not very useful): ______________________________
10. Use the `root` command to **set** the the default partition to be the **same** as the partition of the `grub.conf` and `device.map` files, above. Enter that `root` GRUB command line here:
    ____
11. Having set a `root` device partition prefix, now enter the **shortest** GRUB shell command line that will display the file `/grub/grub.conf`: ______________________________
12. Confirm that this command works without needing a partition prefix: `cat /grub/device.map`
    If it doesn't work, redo the `root` command with the correct partition name prefix.

Disk Information

13. What GRUB shell **command line** will display the **geometry** (partitions) of the **first** disk?
    ____

14. Paste the **four**-line **output** (showing two partitions) of the above GRUB shell **`geometry`** command:
    ____
    ____
    ____
    ____

15. What GRUB shell **command line** will display the **geometry** (partitions) of the **second** disk (the disk you added in a previous lab)? ____

16. Paste the **six**-line **output** (showing four partitions) of the above GRUB shell geometry command:
    ____
    ____
    ____
    ____
    ____
    ____

17. Exit the GRUB Shell and return to your BASH shell prompt. Command used: ________________

6 Configuring GRUB: grub.conf (or menu.lst)

The legacy GRUB configuration file is named **`/boot/grub/grub.conf`** on Fedora. It is named **`menu.lst`** on some other distributions of Linux, and a symbolic link in the same directory on Fedora gives it both names. Remember to snapshot your virtual machine and back up this file before you change it!

1. Use **`ls -li`** on the **absolute** pathname of the above **symbolic link** and give the output here:
   ____
   ____
2. Give the number of lines, words, characters in the **`grub.conf`** file: _________________________
3. Edit the **`grub.conf`** file to change the **`timeout`** from **zero** to **60** so that GRUB waits for you. Give the new number of lines, words, characters in the file (**one** more character): _____________________
4. Save all your work, close your programs, and then reboot your virtual machine. When the boot process begins, you should see a 60 second countdown in progress. Interrupt the countdown by pressing a key.
5. When the countdown is interrupted your system should display the one-entry GRUB menu. Now press just the single letter '**`a`**' to temporarily (for this boot only) alter the kernel command line arguments. Do not press **[Enter]** yet!
6. After pressing '**`a`**' you will see a line that ends with **`rhgb quiet`**. These are both kernel arguments: **`rhgb`** = *RedHat Graphical Boot* – gives a GUI mode booting screen with most of the information hidden while the user sees white and blue bands growing from left to right as Linux boots. **`quiet`** = hides the majority of boot messages before **`rhgb`** starts. You should now **backspace** over the two words **`rhgb quiet`** to erase those two options (for this boot only - the changes are not saved), and press **[Enter]** to continue booting using the new kernel option setting.
7. Watch the screen during the boot process and notice all the boot process information that is displayed on the screen. Note the huge difference made by hiding the graphics screen and enabling all system messages. You can diagnose many problems by viewing the full boot message list.
8. Does using '**`a`**' to change the kernel boot options make a change that affects **every** reboot? _________

7 Adding a boot image menu entry; preview configuration file

The first few lines of the GRUB configuration file give options that affect GRUB. In the bottom half of the file, every **`title`** keyword starts a paragraph of lines (four, in most cases) that defines a "**boot menu entry**". The Fedora 12 file only contains **one** menu entry to start. (Software updates may add more, but you were told not to install any software updates.) We will now add a second boot menu entry. (Where strings are quoted below, do not enter the quotes. The quotes are not part of the string.)

1. Give the number of lines, words, characters in the **`grub.conf`** file: ________________________
2. Back-up the **`grub.conf`** file somewhere safe, so that you can restore it if you make any errors.
3. What keyword begins a "**boot menu entry**"? ________________________
4. How many lines are in a "**boot menu entry**" in your own **`grub.conf`** file? ________________
5. Edit the **`grub.conf`** file using **`vi/vim`** to make these changes:
   a) On the long line that begins with **`kernel`**, remove the two words at the end of the line: "**`rhgb quiet`**". This will allow you to see the kernel boot messages, which are essential to diagnose boot-time problems. (Save the file and confirm that the saved file has two fewer words in it.)
   b) Using four or five command letters in **`vi/vim`**, **duplicate twice** the first and only four-line **boot menu entry** in the file. Duplicating it twice will add **eight** more lines to the file, **duplicating** the first (topmost) existing four-line boot menu section **two** times to create **three** identical sections of **four** lines. The four duplicated lines must each start with a **`title`** line. You must have three boot menu entries in total. (*Hint for vim*: Yank four lines and paste them twice.)
   c) Insert the words "**`single user`**" in front of "**`Fedora`**" in the **`title`** line of the *second* (middle) **boot menu entry**. Next, add the keyword to this second **boot menu entry** that will enable **single-user** mode.
   d) Insert the words "**`text only`**" in front of **`"Fedora"`** in the **`title`** line of the *third* (last) **boot menu entry**. Next, add the digit to this third **boot menu entry** that will enable the system to enter the full multi-user mode **Run Level without** X11 graphics. Record the Run Level __________
   e) Save the file and give the new number of lines, words, characters in the file (you should see exactly **eight** more lines compared with the unmodified file): ________________________
   f) Use **`grep`** to confirm that the new **`grub.conf`** file contains **three** different **`title`** lines and paste the three lines here:

   ____
   ____
   ____

Preview the nw GRUB configuration file using the command-line GRUB shell inside Linux

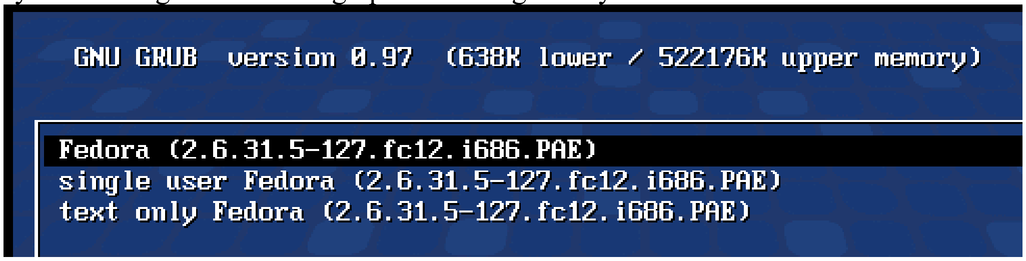
6. Preview the new configuration file using the command-line GRUB shell inside Linux as follows:
   a) At the Linux shell command line, start the GRUB shell (as **root**) and use the **`configfile`** command inside GRUB to load your changed configuration file. As a file name argument to **`configfile`** you will need to specify both the partition and the pathname to the GRUB configuration file you edited above. The errors *Invalid device requested* and/or *Cannot mount selected partition* mean you didn't get the partition name correct. The error *File not found* means you didn't get the pathname correct. The error *Selected disk does not exist* might mean you don't have enough permissions - read all the words. Keep trying until you get it right. Enter the **correct GRUB filename** you used to load the file: ____
   b) When you get the **`configfile`** command right, you will see the message: "**Press any key to enter the menu**". When you see that, use **[Enter]** to enter the menu (other keys will not work). (If you disabled the **`hiddenmenu`** command in GRUB, you will go directly to the menu without needing to push **[Enter]**.)
   c) Underneath the displayed banner "**`GNU GRUB version 0.97`**" you should see **three** menu entries: the original **`Fedora`** menu item and your two additional menu items. If this is not true, break out of the GRUB shell (see below), restore the GRUB configuration file from your saved backup copy, and redo the edits to make it true. You must see **three** menu entries before continuing:

```
0: Fedora (2.6.31.5-127.fc12.i686.PAE)
1: single user Fedora (2.6.31.5-127.fc12.i686.PAE)
2: text only Fedora (2.6.31.5-127.fc12.i686.PAE)
```

7. Break out of (interrupt) the GRUB shell using ^C (control-C) to return to the BASH prompt. (You cannot actually select and run either of these menu entries without rebooting. If you try, you will get a harmless **segmentation fault** error from the Linux kernel that you try to load.)

Run the new GRUB configuration file

8. When you see all three menu entries, as shown above, you are ready to try your new GRUB menu. Reboot the Virtual Machine and interrupt the GRUB countdown to enter the GRUB menu, as you did before. This time, the menu should have **three** entries. If the menu does not have **three** entries, go back and try the edits again. Here is a graphic showing what you should see:



```
GNU GRUB  version 0.97  (638K lower / 522176K upper memory)

Fedora (2.6.31.5-127.fc12.i686.PAE)
single user Fedora (2.6.31.5-127.fc12.i686.PAE)
text only Fedora (2.6.31.5-127.fc12.i686.PAE)
```

a) Try the **second** menu entry to boot **single-user** (*Maintenance Mode*). You should boot up to a black text-only screen with a **root** shell prompt. This is where you would reset the **root password**, if you forget it. Type "**`shutdown -r now`**" at the **root** prompt to reboot again. (If you **`exit`** the single-user **root** shell prompt, your system will continue booting to its default Run Level.)

b) As you reboot, enter the **GRUB** three-item menu again and try the **third GRUB** menu entry to boot **text-only** (no X11 graphics). You should boot up to a black text-only screen with a **`login`** prompt. Log in as the **root** account and type **`telinit`** followed by the Run Level number that will enable full X11 graphics. Your system will proceed to display the graphical login screen.

9. At the graphical login screen, log in as your userid, become the **root** user, and run **`telinit`** followed by the Run Level number that will **reboot** the system to the GRUB menu again.

a) In the GRUB menu, use the arrow keys to move down to select and highlight the **second** menu option (**single user**).

b) This time, use the single letter '**`e`**' key to **edit** the entire menu item. You will open another menu showing all **three** lines in the boot menu. You can view or edit any of the three lines before booting.

c) Use the arrow keys to move down to the **`kernel`** line and again type the single letter '**`e`**' key to **edit** that kernel line. You will be sent to the end of the kernel line, where you can confirm that the options "**`rhgb quiet`**" have been deleted and replaced with the **single-user** option keyword. (If this is not true, cancel the edit, continue the boot process, and redo this exercise to fix it.) Give the last option (the **single-user** option) visible on the kernel line: ______________________________

d) At the end of the kernel line, replace the **single-user** option keyword with the nonsense word "**`doghouse`**" and and then push **[Enter]** to accept the changes. You will return to the previous three-line menu. Follow the instructions on this menu to **boot** the system using your changed menu item. What letter boots the system from this menu? ______________________________

e) The system should boot multi-user with the usual graphical login screen.

10. At the graphical login, log in to the system and copy here the **output** of: **`cat /proc/cmdline`**

____

____

11. Describe what **`/proc/cmdline`** contains: ______________________________________

8 *Password-secured GRUB command line; TAB completion*

1. Take a **snapshot** of your working system (three-item menu) before you try to set passwords in GRUB.
2. Give the number of lines, words, characters in the **`grub.conf`** file: __________________________
3. Add the following GRUB option line near the beginning of the GRUB configuration file, anywhere **after** the comment lines and **before** any **`title`** sections: **`password sesame`**
4. Save the file and give the new number of lines, words, characters in the file (you should see exactly **one** more line and **two** more words): ____
5. In the command line GRUB shell, load and **preview** the modified configuration file (as you did before) using the Linux command-line version of GRUB (do not reboot!):

a) After loading the GRUB configuration file, press **[Enter]** when you see the message: **Press any key to enter the menu** (If you disabled the `hiddenmenu` command in GRUB, you will go directly to the menu without needing to push **[Enter]**.)

b) At the bottom of the screen starting with the banner "`GNU GRUB version 0.97`" you should see instructions on using a password with "`p`". (If this isn't true, you need to break out and fix the file to ask for a password, first.) You should see this paragraph:

```
 Use the ^ and v keys to select which entry is highlighted.
Press enter to boot the selected OS or 'p' to enter a
password to unlock the next set of features
```

c) Type the **seven** characters `psesame` (no spaces) and then push **[Enter]**. (You are typing the command "`p`" immediately followed by the password with no spaces between.)

d) The instructions should change to let you use all the GRUB edit commands. If you see "`Failed!`", you either typed the password command incorrectly (type `psesame`) or you have entered the wrong password in the GRUB configuration file. Push **[Enter]** and try the password again, or break out and fix things so that the password works before continuing. Make sure it works at the command line before you reboot the system!

6. Once you have verified that the password works using GRUB at the shell inside Linux, break out of (interrupt) the GRUB shell using ^C (control-C) to return to the BASH prompt.
7. **Reboot** your system and interrupt the GRUB countdown to display the three-item GRUB boot menu, as before. Note that the '`a`' and '`e`' commands do not work in this password-secured boot menu. Notice also the new instructions at the bottom of this menu, telling you how to enter a password to unlock more features. **Follow** the instructions to **unlock** the menu using the `sesame` password you specified, above.
8. The **unlocked** menu now shows **all** the GRUB commands again. Type the single letter '`c`' to open the GRUB shell. This is the **boot-time** version of the GRUB shell you used at the command line earlier. The difference now is that the machine is running **only** the GRUB shell. No operating system has been found and loaded yet. GRUB is running **stand-alone** without any operating system.

TAB completion in GRUB (stand-alone version)

9. This boot-time version of the GRUB shell has working **TAB** command, device, and file name completion. (The **command-line** version of GRUB also should do this, but is broken under Fedora 12.)

   a) Type the letter '`g`' and push **TAB** - GRUB will complete the command `geometry` for you.

   b) Type an open parenthesis and again push **TAB** - GRUB will tell you what possible disks you have.

   c) Continue and type `hd0` after the parenthesis and again push **TAB** - GRUB will add a comma.

   d) Continue and push **TAB** again (after the comma) and GRUB will list all the partitions on the disk `(hd0,`

   e) Type ^U (control-U) to erase the line, leaving only the GRUB prompt.

   f) Type `root` to display the default **device**. That **device** name is: ______________________________

   g) At the prompt type `ca` and push **TAB** - GRUB will complete the command name `cat` for you.

   h) Type a forward **slash** (the start of a pathname argument to `cat`) and push **TAB** - GRUB will list all the possible file name completions for you, starting at the ROOT on the default device.

   i) After the slash, type `grub/g` and push **TAB** - GRUB will complete the path `/grub/grub.conf` for you. Let the computer do your typing for you! Use the TAB key!

   j) Push the Escape (**Esc**) key to return to the GRUB boot menu, pick the first line to boot, and log in.

9 Encrypted password-secured GRUB

The password in the GRUB configuration file is visible as plain text. This is not very secure. You can use fully encrypted passwords, using the built-in GRUB shell command `md5crypt` to encrypt them:

1. From the command line, start the GRUB shell and run the GRUB command `md5crypt` (no options or arguments). Respond to the password prompt with a new six-character password: `foobar` and record the encrypted value here (cut and paste!): ____
2. In a second terminal window, **edit** the GRUB configuration file and **replace** the `password` line you added earlier with this **new** line: `password --md5` *encrypted_password*

Replace ***encrypted_password*** with the **encrypted** value you recorded, **above**. (Cut and paste!) Save the file. (Check the file; make sure you have the **same** number of lines in the file as before you started.)

3. In the command line GRUB shell, load and **preview** the modified configuration file (as you did before):
   a) After loading the configuration file, press **[Enter]** when you see the message: **Press any key to enter the menu** (If you disabled the **`hiddenmenu`** command in GRUB, you will go directly to the menu without needing to push **[Enter]**.)
   b) At the bottom of the screen starting with the banner "**`GNU GRUB version 0.97`**" you should see instructions on using a password with "**`p`**". (If this isn't true, you need to break out and fix the file to ask for a password, first.) Type the **seven** characters **`pfoobar`** (no spaces) and then push **[Enter]**. (You are typing the command "**`p`**" immediately followed by the password with no spaces between.)
   c) The instructions should change to let you use all the GRUB edit commands. If you see "**`Failed!`**", you either typed the password command incorrectly (type **`pfoobar`**) or you have the wrong encrypted password in the GRUB configuration file. Push **[Enter]** and try the password again, or break out and fix things so that the password works before continuing. Make sure it works at the command line before you reboot the system!
4. Once you have verified that the password works using GRUB at the shell inside Linux, break out of (interrupt) the GRUB shell using ^C (control-C) to return to the BASH prompt.
5. **Reboot** your system and interrupt the GRUB countdown to display the three-item GRUB boot menu, as before. As you did before, verify that the password works at boot time as well.

If you have problems with a forgotten GRUB password locking you out of rebooting, you will need to restore the working snapshot of your Virtual Machine, or alternatively, boot the Virtual Machine into rescue mode with a Linux Install CDROM, and edit the file "**`/sysimage/boot/grub/grub.conf`**" to remove the password.

10 Booting into single-user mode (changing forgotten root password)

To change a forgotten **root** password, you can boot your system in a restricted **single-user** mode that does not start many system daemons and goes directly into a **root** shell prompt. The system should not be left in single-user mode; many things are not started.

To go **single-user**, reboot the system, enter the GRUB menu, edit any **`kernel`** line and remove the **`rhgb quiet`** options and replace them with the single-user option word **`single`** on the end of the **`kernel`** line and boot that modified entry. The system will come up in black-screen text console mode with a **root** shell prompt. You can perform any **root** function, including changing passwords.

1. Reboot in single-user mode exactly as given above and record here the last two lines you see on the black terminal console screen. (The last line is the **root** prompt.) You will have to re-type the two lines; cut and paste will not work in console mode:
   ____
   ____
2. To leave single-user mode and start the system in multi-user mode, simply **`exit`** the console **root** shell. This will exit the single-user shell and allow the system to come up to the default Run Level.

11 Review

Look at the list of commands given in the Class Notes as **`900_unix_command_list.txt`**. You were asked to keep a notebook documenting how you used these commands. Do you know what every one of those commands does, and how to use it?

1. Rewrite **`/home/me/../you/../../etc/../home/./me/../you/../me/./foo`** as a simplified absolute path: ________________________________
2. In an empty directory, how many words are in file **`b`** after this: **`echo x >a ; ls >b`** ________

3. File **a** contains **3** lines. File **b** contains **4** lines. How many lines are in file **d** (not in **c**) after this command line: **ln a d ; ln d c ; cat a a b b d d >c** ____________________
4. What is the final link count of file **a** after all the commands in the above question: ____________
5. What does password **:x:** mean in **/etc/passwd**? ________________________________
   ____
6. If you are in **/home/me** and "**ls -l**" shows "**bar -> /foo**" then dereference **bar** as an absolute path with no symbolic links: ____
7. If you are in **/home/me** and "**ls -l**" shows "**bar -> foo**" then dereference **bar** as an absolute path with no symbolic links: ____
8. If you are in **/home/me** and "**ls -l**" shows "**bar -> dir/foo**" then dereference **bar** as an absolute path with no symbolic links: ____
9. If you are in **/home/me** and "**ls -l**" shows "**bar -> ../foo**" then dereference **bar** as an absolute path with no symbolic links:
   ____
10. If you are in **/home/me** and "**ls -l**" shows "**bar -> ../you/foo**" then dereference **bar** as an absolute path with no symbolic links: ____
11. What command will show the type of file system inside a partition? ______________________
12. What command will show the amount of free disk space in a partition? ____________________
13. What command will recursively show disk usage in directories? _________________________
14. Give a command line that will show lines containing the name **root** inside all *four* account files under **/etc**: ____
15. Give a command line that will *modify* and *move* the home directory of account **bob** to be **robert**:
    ____
16. Give a command that will recursively find *all* pathnames (anywhere) owned by **UID 1000**:
    ____
17. Give a command that shows all partition names and System IDs on the *third* disk:
    ____
18. **T**rue or **F**alse: The System ID of a partition restricts what kind of file system you can put into it: ___
19. Give the Linux and GRUB names of the *third* partition on the *second* disk:
    ____
20. Give a command that will change permissions on a directory to make the names in it readable by group members, but prevent group access to anything in the directory. *Do not change any other permissions*.
    ____
21. Name three types of partitions: ____
22. What option keyword do you add to boot a machine single-user? _______________________
23. What GRUB line do you modify to add the above keyword? __________________________
24. Which option to **ls** displays the directory itself and not its contents? __________________
25. What file records your default Run Level? ______________________________________
26. What command displays your numeric UID and GID? ______________________________
27. How do you execute the program **foo** in the current directory? ________________________
28. How do you display the value of the HOME environment variable? ______________________
29. Comment out the **splashimage** line from your GRUB configuration file and reboot. What is different? ____
30. How do you make your shell always search for commands in **/usr/local/bin**?
    ____
31. Mount the **ext4** file system on **/dev/sdb2** onto /**mnt/test** and try this command (as **root**):

    ```
    # ln /etc/passwd /mnt/test/foo
    ```

    Why does it fail? ____
    Hint: Links are made with inode numbers inside file systems. Different file systems have different sets of inode numbers. How could you make a link that would work:
    ____