1. Is each of the following an absolute pathname, a relative pathname, or a simple filename?

**a. milk\_co**

**b. correspond/business/milk\_co**

**c. /home/max**

**d. /home/max/literature/promo**

**e. ..**

**f. letter.0210**

A: a = Simple Filename

b = relative Pathname

c = absolute Pathname

d = absolute Pathname

e = Hidden Filename (is a relative path name as it means go up 1 level.)

f = Simple Filename

2. List the commands you can use to perform these operations:

a. Make your home directory the working directory

b. Identify the working directory

A: a. cd; (cd~)

b. pwd

3. If the working directory is **/home/max** with a subdirectory named **literature**, give three sets of commands you can use to create a subdirectory named **classics** under **literature**. Also give several sets of commands you can use to remove the **classics** directory and its contents.

A: a = mkdir /home/max/literature/classics,

mkdir ~max/literature/classics,

or mkdir ~/literature/classics

b = rmdir /home/max/literature/classics,

rmdir ~max/literature/classics,

rmdir ~/literature/classics,

or rm -r /home/max/literature/classics

4. The **df** utility displays all mounted filesystems along with information about each. Use the df utility with the **-h** (human-readable) option to answer the following questions.

a. How many filesystems are mounted on your Linux system?

b. Which filesystem stores your home directory?

c. Assuming your answer to exercise 4a is two or more, attempt to create a hard link to a file on another filesystem. What error message do you get? What happens when you attempt to create a symbolic link to the file instead?

A: a. six;

b. /dev/shm; (/dev/mapper/fedora-root)

c. Failed to create hard link `/home/Destop' => `draft': Permission denied;

Failed to create symbolic link `/home/Destop': Permission denied

5. Suppose you have a file that is linked to a file owned by another user. How can you ensure that changes to the file are no longer shared?

A: Make a local copy of the file, not linked to the copy owned by the other user.

6. You should have read permission for the **/etc/passwd** file. To answer the following questions, use cat or less to display **/etc/passwd**. Look at the fields of information in **/etc/passwd** for the users on the local system.

a. Which character is used to separate fields in **/etc/passwd**?

b. How many fields are used to describe each user?

c. How many users are on the local system?

d. How many different login shells are in use on your system? (\_Hint:\_ Look at the last field.)

e. The second field of **/etc/passwd** stores user passwords in encoded form. If the password field contains an **x**, your system uses shadow passwords and stores the encoded passwords elsewhere. Does your system use shadow passwords?

A: a. colon (:);

b. seven;

c. three: root, aplustudent, hongzhang;

d. five;

e. yes.

7. If **/home/zach/draft** and **/home/max/letter** are links to the same file and the following sequence of events occurs, what will be the date in the opening of the letter?

a. Max gives the command **vim letter**.

b. Zach gives the command **vim draft**.

c. Zach changes the date in the opening of the letter to January 31, writes the file, and exits from vim.

d. Max changes the date to February 1, writes the file, and exits from vim.

A: The date will be February 1 because Max is the last person who write the file.

8. Suppose a user belongs to a group that has all permissions on a file named **jobs\_list**, but the user, as the owner of the file, has no permissions. Describe which operations, if any, the user/owner can perform on **jobs\_list**. Which command can the user/owner give that will grant the user/owner all permissions on the file?

A: Using **ls** command can list this file.

The **chmod** command lets you change the file permission.

9. Does the root directory have any subdirectories you cannot search as an ordinary user? Does the root directory have any subdirectories you cannot read as a regular user? Explain.

A: Yes, as an ordinary user, you cannot search or read some subdirectories in the root directory, such as the executable files that are setuid and owned by root because they have root privileges when they run, even if they are not run by root. This type of program is very powerful because it can do anything that root can do (and that the program is designed to do). Because of the power they hold and their potential for destruction, many sites minimize the use of these programs on their systems and ordinary users cannot search or read them.

10. Assume you are given the directory structure shown in Figure 6-2 on page 177 and the following directory permissions:

d--x--x--- 3 zach pubs 512 2010-03-10 15:16 business

drwxr-xr-x 2 zach pubs 512 2010-03-10 15:16 business/milk\_co

For each category of permissions--owner, group, and other--what happens when you run each of the following commands? Assume the working directory is the parent of correspond and that the file cheese\_co is readable by everyone.

a. **cd correspond/business/milk\_co**

b. **ls –l correspond/business**

c. **cat correspond/business/cheese\_co**

A: a. owner: OK; group: OK; other: Permission denied

b. owner, group, and other: Permission denied

c. owner and group: Is a directory; other: Permission denied

11. What is an inode? What happens to the inode when you move a file within a filesystem?

A: An inode is the control structure for a file. Each directory entry associates a filename with an inode. Although a single file might have several filenames (one for each link), it has only one inode. If the two filenames have the same inode number, they share the same control structure and are links to the same file. Hence, when you move a file within a filesyste, the inode will not change.

12. What does the .. entry in a directory point to? What does this entry point to in the root (/) directory?

A: The .. entry is a link to the parent directory. If it is in the root directory, there is no parent, and the .. entry is a link to the root directory itself.

13. How can you create a file named -i? Which techniques do not work, and why do they not work? How can you remove the file named -i?

A: the command to create a file: mkdir ./-i

Because this file likes a command line argument, it does not be easy to remove.

The command to remove this file: rm ./-i

14. Suppose the working directory contains a single file named **andor**. What error message do you get when you run the following command line?

**$ mv andor and\/or**

Under what circumstances is it possible to run the command without producing an error?

A: The error message: **mv: cannot move 'andor' to 'and/or': No such file or directory**

The solution is you should create a file named “and”. The command will be:

**$ mkdir and**

**$ mv andor and\/or**

**$ ls and**

**or**

15. The **ls -i** command displays a filename preceded by the inode number of the file (page 206). Write a command to output inode/filename pairs for the files in the working directory, sorted by inode number. (\_Hint:\_ Use a pipeline.)

A: Command: **ls -i | sort –i**

16. Do you think the system administrator has access to a program that can decode user passwords? Why or why not? (See exercise 6.)

A: Normally, the system administrator cannot decode user passwords. The administrator can assign a new password to a user. Passwords are generally encrypted by a one-way hash so the system can tell when the correct password is entered, but it cannot regenerate the cleartext password. The system applies the hash algorithm to the entered password and checks whether the result matches the stored, encrypted password. A match means the correct password was entered. However, if a user has a weak password, the system administrator can use a program such as crack or John the Ripper to decode a password. I can download either of these utilities using yum. The packages are named crack and john.

17. Is it possible to distinguish a file from a hard link to a file? That is, given a filename, can you tell whether it was created using an **ln** command? Explain.

A: No. All links to a file are of equal value: The operating system cannot distinguish the order in which multiple links were created. When a file has two links, you can remove either one and still access the file through the remaining link. You can remove the link used to create the file, for example, and as long as one link remains, still access the file through that link.

18. Explain the error messages displayed in the following sequence of commands:

**$ ls -l**

total 1

drwxrwxr-x. 2 max pubs 1024 03-02 17:57 dirtmp

**$ ls dirtmp**

**$ rmdir dirtmp**

rmdir: dirtmp: Directory not empty

**$ rm dirtmp/\***

rm: No match.

A: There is a file whose name begins with a period (a hidden file) in the

dirtmp directory. Use ls with the –a option to list the file. Remove the file, and then you will be able to remove the directory.

19. Name and describe the 3 P's of linux problems?

A: All problems in Linux can be described by the 3 P’s which is Path, Permission and dependencies.