Chapter 3 Notes

I. The /etc/passwd file contains a list of all the system user accounts, along with some basic configuration information about each user.

II. The default prompt symbol for the bash shell is the dollar sign ($)

III. Entering the man command followed by a specific command name provides that utility's manual entry.

- A. You can page through the man pages by pressing the spacebar, or you can go line by line using the Enter key
- B. You can use the arrow keys to scroll forward and backward through the man page text
- C. When you are finished with the man pages, press the q key to quit
- D. You can search the man pages using keywords
    1. The syntax is $ man -k keyword
- E. To see the pages desired, you type $ man [section#] topic
    1. For the overview man pages in section 7, type $ man 7 hostname
- F. In addition, most commands accept the -help or --help option
    1. For example, you can type hostname -help to see a help screen

IV. The virtual directory contains file paths from all the storage devices installed on the computer, merged into a single directory structure

V. The Linux virtual directory structure contains a single base directory, called the root

VI. Linux uses a forward slash (/) instead of a backward slash (\) to denote directories in file paths

VII. The first hard drive installed in a Linux system is called the root drive

- A. The root drive contains the virtual directory core. Everything else builds from there
    1. Mount points are directories in the virtual directory where you can assign additional storage devices
    2. Often system files are physically stored on the root drive. User files are typically stored on a separate drive or drives

VIII. Common Linux Directory Names:

- A. /: Root of the virtual directory, where normally, no files are placed
- B. /bin: Binary directory, where many GNU user-level utilities are stored
- C. /boot: Boot directory, where boot files are stored
- D. /dev: Device directory, where Linux creates device nodes

E. /etc: System configuration files directory
F. /home: Home directory, where Linux creates user directories
G. /lib: Library directory, where system and application library files are stored
H. /media: Media directory, a common place for mount points used for removable media
I. /mnt: Mount directory, another common place for mount points used for removable media
J. /opt: Optional directory, often used to store third-party software packages and data files
K. /proc: Process directory, where current hardware and process information is stored
L. /root: Root home directory
M. /sbin: System binary directory, where many GNU admin-level utilities are stored
N. /run: Run directory, where runtime data is held during system operation
O. /srv: Service directory, where local services store their files
P. /sys: System directory, where system hardware information files are stored
Q. /tmp: Temporary directory, where temporary work files can be created and destroyed
R. /usr: User binary directory, where the bulk of GNU user-level utilities and data files are stored
S. /var: Variable directory, for files that change frequently, such as log files

IX. You use the *change directory command* (cd) to move your shell session to another directory in the Linux filesystem
   A. The cd command syntax is pretty simplistic: $ cd [destination]
   B. If you don't specify a destination on the cd command, it takes you to your home directory
   C. Using absolute directory references
      1. Defines exactly where the directory is in the virtual directory structure, starting at the root
         a) "Full name for a directory"
      2. Always begins with a forward slash (/)
   D. Using relative directory references
      1. Allow you to specify a destination directory reference relative to your current location

2. Doesn't start with a forward slash (/)
    a) Starts with either a directory name (if you're traversing to a directory under your current directory) or a special character
        (1) christine@server01:~$ cd Documents
3. You can also use a special character to indicate a relative directory location
    a) The single dot (.) to represent the current directory
    b) The double dot (..) to represent the parent directory
        (1) If you are in your home directory (/home/christine) and want to go to the /etc directory, you could type the following:
            (2) christine@server01:~$ cd ../../etc
            christine@server01:/etc$ pwd
            /etc
            christine@server01:/etc$

X. The tilde in the prompt indicates that your shell session is located in your home directory
    A. christine@server01:~$ [command]

XI. The pwd command displays the shell session's current directory location, which is called the *present working directory*

XII. To see what files are available on the system, use the *list* command (ls)
    A. Doesn't show hidden files
        1. Filenames start with a period (.)
        2. Use the -a parameter to display them
    B. The -F parameter flags the directories with a forward slash (/), to help identify them in the listing.
        1. Similarly, it flags executable files with an asterisk (*), to help you more easily find files that can be run on the system
    C. The recursive option (-R), shows files that are contained within subdirectories in the current directory
    D. Multiple options can be combined into a single line
        1. $ ls -FR
    E. The -l parameter produces a long listing format, providing more information about each file in the directory
        1. The long listing format lists each file and subdirectory on a single line

2. The first line in the output shows the total number of blocks contained within the directory. After that, each line contains the following information about each file (or directory):
    a) The file type — such as directory (d), file (-), linked file (l), character device (c), or block device (b)
    b) The file permissions
    c) The number of file hard links
    d) The file owner username
    e) The file primary group name
    f) The file byte size
    g) The last time the file was modified
    h) The filename or directory name
3. Uses a filter to determine which files or directories it should display in the output
    a) $ ls -l my_script
       -rwxrw-r-- 1 christine christine 54 May 21 11:26 my_script

4. Recognizes standard wildcard characters and uses them to match patterns within the filter:
    a) A question mark (?) to represent one character
    b) An asterisk (*) to represent any number of characters
        (1) $ ls -l my_scr?p
            t-rw-rw-r-- 1 christine christine 0 May 21 13:25 my_scrapt
            -rwxrw-r-- 1 christine christine 54 May 21 11:26 my_script
        (2) $ ls -l my*
            -rw-rw-r-- 1 christine christine 0 May 21 13:25 my_file
            -rw-rw-r-- 1 christine christine 0 May 21 13:25 my_scrapt
            -rwxrw-r-- 1 christine christine 54 May 21 11:26 my_script
    c) Using the asterisk and question mark in the filter is called file globbing
        (1) The processing of pattern matching using metacharacter wildcards
    d) You can also use brackets
        (1) $ ls -l my_scr[ai]pt

-rw-rw-r-- 1 christine christine 0 May 21 13:25 my_scrapt
-rwxrw-r-- 1 christine christine 54 May 21 11:26 my_script

(2) You can specify a range of characters, such as an alphabetic range [a - i]

(3) You can specify what should not be included in the pattern match by using the excla-mation point (!)

(a) $ ls -l f[!a]ll

F. To view a file or directory's inode number, add the -i parameter

1. The inode number of a file or directory is a unique identification number that the kernel assigns to each object in the file system

XIII. You can use the touch command to easily create an empty file

A. $ touch test_one

B. Assigns your username as the file owner

C. Can also be used to change the modification time

1. To change only the access time, use the -a parameter with the touch command

XIV. The cp command copies files and directories from one location in the filesystem to another

A. Uses two parameters — the source object and the destination object:

1. cp [source] [destination]

B. When both the source and destination parameters are file names, the cp command copies the source file to a new destination file

1. The new file acts like a brand new file, with an updated modification time

C. The -R parameter allows you to recursively copy the contents of an entire directory in one command

D. You can also use wildcard metacharacters in your cp commands

E. It is best to add the -i option to force the shell to ask whether you want to overwrite a file

XV. Tab auto-complete allows you to start typing a filename or directory name and then press the tab key to have the shell complete it for you

XVI. Two types of file links are available in Linux:

A. A symbolic link

1. A physical file that points to another file somewhere in the virtual directory structure

2. The two symbolically linked together files do not share the same contents
    a) Link tends to have a very small size compared to original file
3. Use the ln command with the -s option to create a symbolic link
    a) $ ln -s data_file sl_data_file
4. Has a different inode number from the original

B. A hard link
1. Creates a separate virtual file that contains information about the original file and where to locate it
2. Physically the same file
    a) File size is exactly the same
3. Use the ln command without any parameters to create a hard link
    a) $ ln code_file hl_code_file
4. Shares the same inode number with the original

XVII. The mv command is available to move both files and directories to another location or change its name

A. mv affects only a file's name
B. You can use the mv command to move a file's location and rename it, all in one easy step
1. $ mv /home/christine/Pictures/fzll /home/christine/fall

XVIII. The command to remove files in the bash shell is rm

A. After you remove a file, it's gone forever
1. Therefore, a good habit is to always tack on the -i parameter to the rm command
B. You can also use wildcard metacharacters to remove groups of files
C. If you're removing lots of files and don't want to be bothered with the prompt, use the -f parameter to force the removal
D. The -r option allows the command to descend into the directory, remove the files, and then remove the directory itself
1. Has the same function as -R
E. The rm -rf command gives no warnings

XIX. By default, the rmdir command works only for removing empty directories

A. Has no -i option to ask if you want to remove the directory

XX. Use the mkdir command to create a new directory

A. You can create several directories and subdirectories at the same time with the -p parameter
    1. $ mkdir -p New_Dir/Sub_Dir/Under_Dir
    2. Makes any missing parent directories as needed

XXI. The file command can peek inside of a file and determine just what kind of file it is
A. Can be used to find which file a symbolic link is linked to
B. Can determine the platform that a binary executable program was compiled for and what types of libraries it requires

XXII. The cat command is a handy tool for displaying all the data inside a text file
A. The -n parameter numbers all the lines for you
B. The -b parameter numbers only numbers the lines that have text in them
C. The -T parameter replaces any tabs in the text with the ^I character combination
D. For large files, the text in the file just quickly scrolls off the display without stopping

XXIII. The more command displays a text file, but stops after it displays each page of data
A. You can use more to navigate through a text file by pressing the spacebar or you can go forward line by line using the Enter key
B. When you are finished navigating through the file using more, press the q key to quit

XXIV. The less command name is an advanced version of the more command
A. It provides several very handy features for scrolling both forward and backward through a text file, as well as some pretty advanced searching capabilities
B. Can display a file's contents before it finishes reading the entire file
C. Supports the same command set as the more command, plus many more options

XXV. The tail command displays the last lines in a file
A. By default, it shows the last 10 lines in the file
B. You can change the number of lines shown using tail by including the -n parameter
    1. $ tail -n 2 log_file
       line19
       Last line - line20
C. The -f parameter allows you to peek inside a file as the file is being used by other processes

XXVI. The head command displays a file's first group of lines

A. By default, it displays the first 10 lines of text
B. Similar to the tail command, the head command supports the -n parameter so you can alter what's displayed
C. The head command doesn't support the -f parameter feature as the tail command does