CSI 402

What is the Shell?

The Shell

- The shell is a program that takes in keyboard commands, and then passes them to the operating system to execute.
- Almost all linux distros provide a shell program called bash.
- Bash stands for Bourne Again SHell.
- Bash is the replacement for sh, which was the original unix shell program written by Steven Bourne.
- We need a terminal emulator to interact with the shell.

- If the last character of the prompt is “#” rather than “$”, then we are in the state of one of two things: Either we are logged in the root user, or we selected a terminal emulator that provides superuser/admin privileges.
- The command line remembers up to 1000 previous commands in its command history.

Cursor Movement

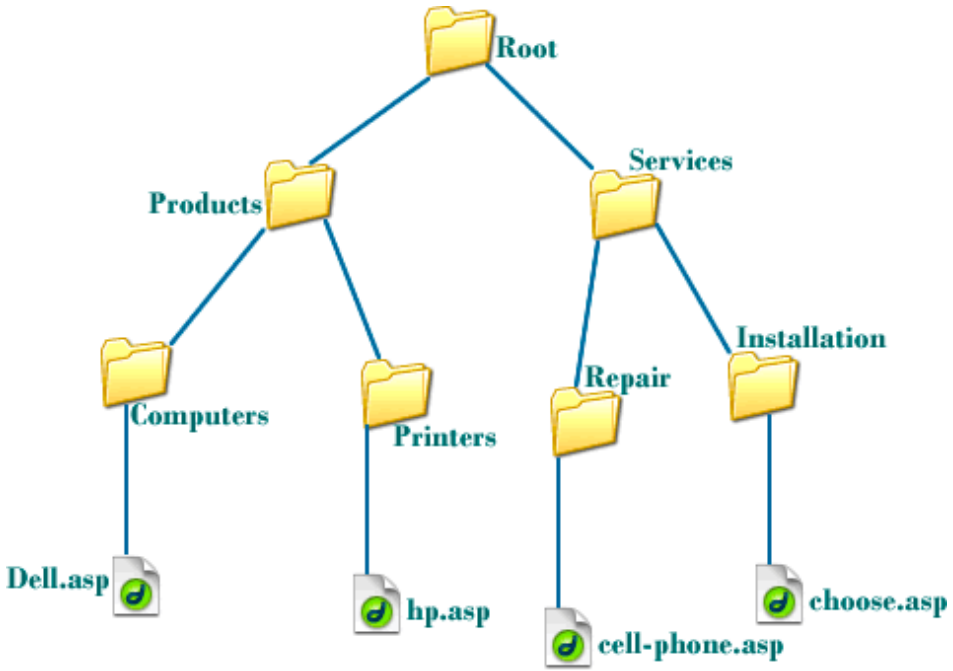
- You can use the left and right arrow keys to move the cursor on the terminal. This makes editing commands much easier.
- You can also highlight text on the terminal which will be maintained by X(X Window System). If you press the middle mouse button it will be pasted at the cursors' location.

Simple Commands

- “date” – displays the current time and date
- “cal” – displays a calendar of the current month
- “df” – displays the amount of current free space on your disk drives
- “free” – displays the amount of free memory
- “exit” – exit a terminal session

Chapter 2 - Navigation

- Unix-like operating systems such as Linux organizes its files in a hierarchical directory structure.



- They are organized in a tree-like pattern of directories.
- The first directory is the root.
- Windows has a separate file system tree for each storage device. While Unix-like systems, like Linux, have only one single file system tree.

Working Directories

- To print the current working directory, simply type “pwd”
- When you first log in to a terminal session, your current working directory is set to your home directory. The home directory is the only place a regular user is allowed to write files.
- To list the files and directories in the current working directory, type “ls”.
- You can also use “ls” to list the contents of any other directories, not just the current working directory.
- To change the current working directory, use the ”cd” command followed by the pathname of the desired working directory.

Absolute vs Relative Pathnames

- Absolute pathnames reference the root directory. Absolute paths contain the full path to the file. You can be in any directory on the system and be able to access the file using an absolute path.
- A relative path only works if you are in the current working directory.
- For example, I can provide you the full link to a website i.e: http://ansonfoong.com/documents/resume.html, this is an example of an absolute path.
- Imagine I was in the directory “documents”, I could access resume.html via relative path.
- The relative pathname starts from the current working directory.

Relative Pathnames

- The “.” notation refers to the working directory, and the “..” notation refers to the parent of the working directory.
- So for example, if you wanted to change the working directory to /usr/bin, you would provide the absolute pathname. i.e: cd /usr/bin
- We can change the working directory to usr in two ways.
- Absolute Path: cd usr (Refers to the current directory)
- Relative Path: cd .. (This will go back one directory, aka go to the parent of the cwd)

Shortcuts

- “cd ” – Changes the working directory to your home directory.
- ”cd - “ – Changes the working directory to the previous working directory.
- “cd ~user_name” - Changes the working directory to the home directory of user_name. For example, cd ~bob will change the directory to the home directory of user ~bob.

Files

- Files that begin with a dot ” . “ are hidden.
- File names and commands are case sensitive in Linux.
- Doing ls -a will list hidden files that begin with a dot ”.”
- Linux has no concept of ”file extension” i.e: .js, .java, .tar, etc.. You may name your files any way you like.

Chapter 3 – Exploring the System

- The “ls” command lists contents and directories of the current working directory.
- We can also do “ls (name of directory in the current working directory)” to view other contents and directories in the specified directory.
- For example, suppose your current working directory is the Desktop, and the Desktop contained these folders: “Projects, Documents, Pictures” you can do “ls projects” to list all of the contents and directories in *projects*.

Command, Option, Arguments

- Commands are often followed by one or more options that modify their behavior. I.e: “ls -l” produces a long format output for all of the files in the directory.

```
[me@linuxbox ~]$ ls -l
total 56
drwxrwxr-x 2 me me 4096 2007-10-26 17:20 Desktop
drwxrwxr-x 2 me me 4096 2007-10-26 17:20 Documents
drwxrwxr-x 2 me me 4096 2007-10-26 17:20 Music
drwxrwxr-x 2 me me 4096 2007-10-26 17:20 Pictures
drwxrwxr-x 2 me me 4096 2007-10-26 17:20 Public
drwxrwxr-x 2 me me 4096 2007-10-26 17:20 Templates
drwxrwxr-x 2 me me 4096 2007-10-26 17:20 Videos
```

Combining Options

- You can also combine options too.
- “ls -lt” will list all of the files in a long format output as well as the file’s modification time.

- You can also view multiple directories.
- “ls ~ /usr” – This will list all of the directories in the home directory (specified with the ~ character) and the directories in /usr.
- You can also do “ls –l” which will change the output to long format.
- You can use the following arguments with the ls command. (See next page).

Option	Long Option	Description
`-a`	`--all`	List all files, even those with names that begin with a period, which are normally not listed (i.e., hidden).
`-A`	`--almost-all`	Like the `-a` option above except it does not list `.` (current directory) and `..` (parent directory).
`-d`	`--directory`	Ordinarily, if a directory is specified, `ls` will list the contents of the directory, not the directory itself. Use this option in conjunction with the `-l` option to see details about the directory rather than its contents.
`-F`	`--classify`	This option will append an indicator character to the end of each listed name. For example, a "/" if the name is a directory.
`-h`	`--human-readable`	In long format listings, display file sizes in human readable format rather than in bytes.
`-l`		Display results in long format.
`-r`	`--reverse`	Display the results in reverse order. Normally, `ls` displays its results in ascending alphabetical order.
`-S`		Sort results by file size.
`-t`		Sort by modification time.

Long Format

- -rw-r--r-- 1 root root 213414 2018-02-17 12:15 sample.txt
- This is an example of long format output when you use the -l argument. We will look specifically at “-rw-r--r--”.
- The first character, ”-” tells us what kind of file it is. In this case, - means it is a regular file, and if the first character is “d”, then the file is a directory.
- The next three characters, “rw-” are access rights, aka reading/writing.
- The next three characters, r-- are members for the file group, and the last three are for everyone else.

Cont.

- The digit “1” that comes after the -rw-r--r-- tells us the file’s number of hard links.
- Root means the the username of the file’s owner.
- The root field after the first one is the name of the group which owns the file.
- 213414 is the size of the file.
- And the rest are self-explanatory/obvious.

- You can use the “file” command to determine the type of a file.
- You can use the “less” command to view *text file* contents.
- Text is a simple one-to-one mapping of characters to numbers.

less commands

The table below lists the most common keyboard commands used by `less`.

Table 3-3: less Commands

Command	Action
Page Up or `b`	Scroll back one page
Page Down or space	Scroll forward one page
Up Arrow	Scroll up one line
Down Arrow	Scroll down one line
`G`	Move to the end of the text file
`1G` or `g`	Move to the beginning of the text file
`/`*characters*	Search forward to the next occurrence of *characters*
`n`	Search for the next occurrence of the previous search
`h`	Display help screen
`q`	Quit `less`

Symbolic Links

- Are files that can reference other files with different names.
- Rather than creating a new version of the file that is being occupied in a program or by a user, we can create a symbolic link/sym-link/soft link that points to the file every time it is updated. View pg. 47 for better understanding.

Chapter 4 - Manipulating Files and Directories

- Wildcards allow you to specify groups of filenames. Using wildcards is also known as *globbing.* It allows you to select filenames based on patterns.
- Example of using a wildcard: “ls [[:upper:]]*” this will list all of the files that match with the first character beginning with an uppercase.

Try the following Wildcard commands

- To get a better grasp of how these wildcard commands work, try the following commands:
- ls [ijk]* - list all files that begin with i, j, or k. If the file is a directory, it will list all of the contents in that directory.
- ls [[:upper:]]* - list all files that begin with an uppercase character, if the file is a directory, it will list all of the contents in that directory.
- Create a file called “index.html”, in the same cwd (current working directory), type “ls index?????”, index.html will show up. The command will list all files that start with ”index” and end with exactly five characters.
- There are obviously more options, but you get the point.

Creating directories using the “mkdir” command

- You can use the **mkdir** command to create directories.
- “mkdir name of directory”
- You can also create multiple directories: “mkdir dir1 dir2 dir3 ……. “

Copy files and destinations with “cp”

- The cp command copies files or directories.
- “cp item1 item2” - this copies the single file item1 or directory item1 to file or directory item2.
- “cp item1 item2 item3 dir1” - this copies multiple items or directories into the directory dir1.

Creating Links

- The **ln** command is used to create either hard or symbolic/soft links.
- “ln file *link”* creates a hard link.
- “ln -s item *link”* creates a soft link. In this case, item is either a file or a directory.

Hard Links

- Symbolic links are more modern compared to hard links, but they are the original way on Unix to creating links.
- By default, every file has a single hard link that gives the file its name. Recall that “-rw-r--r-- 1 …” from the output displayed using the long option, the field ’1’ represents the amount of **hard** links the file itself has.
- Hard links cannot reference a file outside its own file system. Which means a link cannot reference a file that is not on the same disk partition as the link itself.
- Hard links may not reference a directory.

Hard links cont.

- When a hard link is deleted, the content of the files remain unchanged until ALL hard links are removed. That means space is not deallocated.

Symbolic Links

- Symbolic links create a file that contain a text pointer to the referenced file or directory.
- If a symbolic link points to a file, then we may write to the symbolic link, this will also write to the file it is pointing to.
- If we delete the file that the link is pointing to, the link will still exist but will point to nothing. The link is said to be broken.
- The **ls** command will display broken links in red to reveal their presence.