Cayabyab, George
Onate, Kim
Laguio, Juliano

Laboratory Activity 1 Synthesis Paper

Overview

This paper talks about the basics on operating the linux operating system, which includes command prompt basics, some basic commands, and the linux file system.

The command prompt

The prompt is an interface that allows users to input commands, which are executed linux. The prompt is in the form *username@computer_name* :*current_directory* $, although this can be changed by the user and depends which shell is currently being used, or different shells uses different forms.

Linux Commands

A command is a statement in linux, when typed in a terminal, does something specific and is executed by the linux. There are 4 types of commands, which may be an executable program, a command built into the shell itself, a shell function, or an alias. The scope of the commands tackled in this paper would only include what came on a fresh install of ubuntu, thus contains no admin/user made/modified commands (Shotts, W. E.)

(McCarty, 1999) Commands in linux are in the form *(command) (-(options) (argument)?)* (arguments)**

The *command* is an identifier to what command linux would execute
The *options* modifies how a command would work
The *arguments* specifies on what the command would work on.
It should be noted that *command* is case sensitive, meaning that **CD** and **cd** are possibly 2 different commands

Going back to the types of command, in linux, the command, **type** *command*, where *command* is the name of the command, is a shell builtin command where it is used to determine what kind of command it is.

Command Documentation/Help Commands

Due to a high amount of commands, it is not always expected a user or a systems administrator would remember all those commands and there would be times where access to the internet is impossible. There are commands within linux that aid the user in searching for a command or to know what a certain command does (Getting help from the Linux shell (man, apropos, whatis), 2012).

The **apropos** *string* command is for searching linux's built-in manual pages and descriptions and does so as it searches for string. For example, executing the command **apropos** b, would display all command names, and command descriptions that contains the character 'b'

The **whatis** *command* is like apropos, the difference that the **whatis** command searches for an exact match, or returns "nothing appropriate" if no exact match is found. Executing **whatis ls** would display the command and it's description, on the other hand, executing **whatis ls** would display "nothing appropriate"

The **man** *command*, short for manual, is the main documentation for the linux commands. The manual contains the description of the command, sometimes the author of the command, and describes what options and parameters can be given to the command.

Keyboard Commands

Aside from the text-based commands, linux also features keyboard commands, there are many keyboard commands, only a few would be listed. The **up arrow** key would cycle through previous commands, and the **down arrow** key would cycle through "forward" command if the **up arrow** key was previously pressed, in addition, the command **history** would output previous text commands executed ever since the user logged-in. Pressing the **tab** key allows for autocompletion of command/file/directory names. The **CTRL+L** key commands clears the screen without typing the command **clear** (Chhetri, 2014). Th linux operating system features more commands, which is not listed here

Job/Process/Task Management for Users

Linux allows for bringing the current process to the background in the event the user want to "minimize" his work, by putting it in the background, and come back for it later to "maximize" it, by bringing it to the foreground (Chhetri, 2014). Pressing the **CTRL+Z** key would put the current process to the background, this would output a number in square brackets, this number would be needed to bring the process to the foreground. Using the command **jobs** would display processes in the background and their number. Using the command **fg** *number* brings the program with the number to the foreground. Pressing the **ALT+SysReq+I** would kill all processes. The **SysReq** key is commonly the same as the print screen key.

Linux File System

A file is a computer object that contains data, which could be in the form of text, videos, and many more, but essentially files contain binary. The form of data contained in the file can be read depending on the application determined by the user or the operating system. A video, such as an mp4, can be opened using a text editor, but would display gibberish.

A directory is a special type of file, it can contain other files, and thus other directories as well. Since directories can also contain directories, files must be navigated through and located in some way. Files are located using directory paths, this path dictate which directory to go through to locate the file. For example, the directory containing removable devices has the path */media*.

Relative and Absolute paths

There are 2 kinds of directory paths, which is relative and absolute. Relative paths are paths that are relative to the current working directory, the current directory the user or the program execution is currently at, and absolute paths are paths that starts at the root directory. The root directory named as '/' is the directory that contains all files, it is the starting directory, meaning it is the parent of all parent directories.

Initially, the user is always at the home directory, which is symbolized as "~" with the path "*/home*" The previous directory of the current directory is symbolized as "..", 2 dots, and the current directory as ".", A single dot. With these, it means that, while at the "*/home /sysmgmt*", the root directory can be accessed as either *"/"* absolute or "../../". Do note the previous example was a bad example, bad conveys the point that the previous directory symbols can be chained, however the problem with the relative, is in the event that the current directory changes, for example, the current directory is now "*/home /sysmgmt/documents*" the absolute path example would still refer to the root directory, in contrast to the relative path, which now refers to the "*/home*" directory.

File System Commands

File Navigation Commands

Linux contains commands that allow and aid users to navigate through the file system of linux. The most commonly used would be stated here. The **cd** *directory_path* command, also known

as the change directory command, it allows the user to change the current working directory, this allows the user to navigate through the directories. The **ls** list the names of the files in the current working directory. It has The options *-a* which shows hidden files, which are files starting with the character '.', dot, *-l* provides statistics, such as the date the file was created, it's permissions, the user who created it, and more. The **pwd** command is for outputting what is the current working directory.

File Manipulation Commands

Linux contains commands that allows users to move files to various locations, copy files to other directories, and much more. The **mkdir** *directory_name* allows the creation of new directories. The argument *directory_name* can be a name, or the path where to create the directory concatenated with the name. The **rmdir** *source* and **rm** *source* commands is for deleting delete files. Take note that not empty directories cannot be deleted, in order to delete non-empty directories, the *-r* or *-R*, which means recursion, must be specified, this option would delete the directory and its contents. The **mv** *source_path destination_path* command is for moving files The **cp** *source_path destination*_path command is for copying files. Files can be rename through the use of the **mv** command, whereas the *source_path* and *destination_path* are the same, being the difference is that the filenames are different. Users could use **rename** *source_name new_filename*, which is an advanced form of mv, that it uses RegEx. A quick hack to creating a file in linux is to use the **touch** *file* command, although **touch** is used for altering timestamps of files, based on the **man** pages, specifying a non-existent *file* argument would create an empty file, although there are other ways such as using vim, **echo**, and others.

References

It should be known that some of the information was gathered from the manual pages of linux.

- Shotts, W. E., Jr. (2017). Working with Commands. Retrieved June 28, 2017, from http://linuxcommand.org/lc3_lts0060.php
- McCarty, B. (1999). 4.2 Working with the Linux Command Prompt. Retrieved June 28, 2017, from http://www.oreilly.com/openbook/debian/book/ch04_02.html
- Getting help from the Linux shell (man, apropos, whatis). (2012, November 24). Retrieved June 28, 2017, from https://mutap.wordpress.com/2012/11/24/getting-help-from-the-linux-shellman-apropos-whatis/
- Chhetri, S. (2014, January 08). 30 useful Linux terminal keyboard shortcuts. Retrieved June 29, 2017, from http://sharadchhetri.com/2014/01/09/30-useful-linux-terminal-keyboard-shortcuts/