CSCI 274 - Intro to Linux OS

Week 2 - Interacting with the Command Line and ~/.bashrc Configuration File

Mona Wade (dwade@mines.edu)

Overview

1. CHEATSHEET.LNX
2. Interacting with the Command Line
3. Understanding the meaning of **~**, **.** , **..**
4. .bashrc

CHEATSHEET.LNX

Notes from the semester that students can use on the Midterm and Final exam. Must be written in a Linux environment using a text editor of choice.

- Will submit for review twice throughout the semester
- You can not share/distribute; Collaboration Policy applies

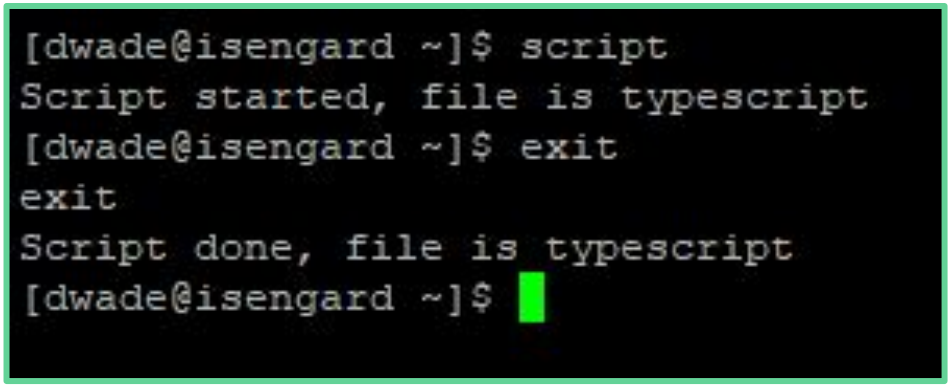
Sample

```
Vim/Vi techniques {
        1. delete blank all blank lines [ :g/^$/d
        2. delete from the current cursor position to the end of line [
                a. enter Visual Mode via Ctr-V
                b. do: Shift+d
                        OR
                a. d$
                                                                        ]
        3. change a word, globally [ :%s/\<word>/newWord/g ]
File Navigation {
  * $ man <softwareName> = short for manual. It helps gets the manual pages for dif    ferent software
  * $ apropos <keyword>  = searches through all the command lines tools and prints     what is availab
  * $ clear = makes the page blank
  * $ script <fileName> = captures everything in the terminal into a text file. It     records everyth
  * $ pwd = present working directory; tells you where you are in the directories.
  * $ ls = list; list everything in the directory that you are currently in.
  * $ ls -l = shows a long list of everything in the directoty that you are current    ly in.
  * $ ls -a = wiil show all files including hidden files.
  * $ ls <file/path> = will list the files and directories that exist at that path
  * $ cd <directroyName> = allows you to navigate to a different directory
  * $ cd .. = allows you to go to the parent directory
  * $ cd <file/path/name> = will take you the file name directory
  * $ cd = will take you back to your home directory.
  * $ cd ~/<directoryName = will brings you to the directory you want in the home d    irectory.

        1. How to View Files {Using a text editor to view files is not the best option.}
                * $ less <fileName> = allows you to view the file. Most common way to view big fi    l
                * # less is a prgram that was made to view files similar to the program "more". I    t

        2. Information about a file
                * $ file simple.cpp = this will tell you information about the file
                * $ file exe = compiled c++ code. talking about how it is an exectuable.
                * You are able to run this commmand on a directory and it will give                  i
                * $ wc - way to count things in a file
                * $ wc -l = counts the number of new lines that are in the file
                * SIDE NOTE: you can get vim to show you line number instead of the "~" character
                * $ wc -w = counts the number of words in a file

Common Pipeline Commands {
        1. read a file [ $ cat fileName ]
                * `cat file` and `cat < file` is the same thing
        2. tr { automatically translates (subsitutes, or maps) one set of characters to another. It co
                input to the standard output with substitution or deletion of selected characters. }
                * FORMAT -> tr [Options] set1 [set2]
                * options list {
                        a. `-d` = deletes chracters in set1 from the input
```

Tip

script is a UNIX command-line application that records a terminal session.

It stores the output as text file in the current directory and the default filename typescript .



Interacting with Command line

man - used to display the user manual of any command that we can run on the terminal

$ man [OPTION] .. [COMMAND NAME] ...

apropos - helps the user when they don't remember the exact command

$ apropos [OPTION..] .. KEYWORD ..

pwd - **P**rint **W**orking **D**irectory

$ pwd -L/P

Interacting with Command line

cd - used to **C**hange current working **D**irectory

$ cd [DIRECTORY]

less - used to read contents of text file one page(one screen) per time

$ less FILENAME

mkdir - allows the user to create directories

$ mkdir [OPTIONS] ... [DIRECTORY]

Interacting with Command line

rmdir - used remove empty directories from the filesystem

$ rmdir [-p] [-v | - -verbose] [- -ignore-fail-on-non-empty] directories ...

mv - used to **move** one or more files or directories

$ mv [OPTION] source destination

cp - used to **copy** files or group of files or directory

$ cp [OPTION] Source Destination

ls - lists directory contents of files and directories

$ ls [-t, -1, -l, -h, -d]

Understanding the meaning of ~, . , ..

~ = The tilde is a Linux "shortcut" to denote a user's home directory

. = denotes the current directory

.. = denotes the “parent” directory

.bashrc

Reminder: shell = an interpreter which can accept commands from the user and run them to perform operations

The **.bashrc** file is a shell script which is run every time a user opens a new shell. It is a good place to put commands you want to run every time you open a shell.

Typically, it contains configurations for the terminal sessions such as:

coloring	completion
shell history	command aliasing
And more ….	