

## **Command Line Commands**

Cd = change directory (.. = go up a level)

Ls = list of files in current dir

Ls -a = list all

Ls -l = list files in long format

Ls -t = list files in order of last modified

Ls -alt = all list options used together

Cp = copies files & dir (ex: cp Frida.txt Lincoln.txt -> copies Frida into Lincoln); can also copy multiple files at once (source files first & destination last)

Mkdir = make directory

Touch = updates "last modified" date to now. Also creates empty file if filename doesn't exist

Mv = move (ex: mv wonderwoman.txt batman.txt superhero/) = moves wonderwoman.txt & batman.txt to the superhero/ folder; can also be used to rename a file

Rm = remove; deletes files

Rm -r - ("recursive") deletes a directory and all child directories

Pwd = print working directory

Exit = close cmd line window

\*up arrow = repeat last command

Echo = command accepts string (ex: \$echo "Hello") as standard input (stdin) and echoes the same string back to the terminal as standard output (stdout). \*NOTE: (stderr) is a standard error message for failed process

> = (overwrites) command redirects standard output to a file (ex: \$echo "Hello" > hello.txt)

Cat = commands a file to output the contents to the terminal (ex: \$cat hello.txt)

>> = (appends) redirects output to file, but appends to content already in file

< = takes standard input from file on right & inputs it into program on the left

| = ("pipe") Takes the standard output of command on the left, and pipes it as standard input to the command on the right

Sort = takes the standard input and orders it alphabetically for the output

Uniq = ("unique") filters out adjacent, duplicate lines in a file.

Grep = ("global regular expression point") Searches for lines in files that match a pattern and returns the results. It is case sensitive. (ex: \$grep Mount mountains.txt -> searches for "Mount" in mountains.txt)

Grep -i = allows the command to be case insensitive

Grep -R = searches all files in a directory and outputs filenames and lines containing matched results (-R = "recursive") (ex: `$grep -R Arctic /home/ccuser/workspace/geography`) Searches that directory for string "Arctic" and outputs filenames and lines with matched results.

Grep -l = only outputs filenames with matched results

Sed = ("stream editor") Accepts standard input and modifies it based on an expression, before displaying it as output data. Similar to "find and replace". (ex: `$sed 's/snow/rain/' forests.txt`) "s" = substitution, ALWAYS used when using sed for substitution; "snow" = search string/text to find; "rain" = replacement string/text to add. (ex: `$sed 's/snow/rain/g' forests.txt`) "g" = global

Nano = command line text editor (similar to notepad). Keyboard input only. (ex: `nano hello.txt`) – opens a new text file called hello.txt. (ex: "Hello, I am nano") – text string entered in nano

^ = Ctrl in nano

Ctrl + O = save file

Ctrl + X = exit nano program

Ctrl + G = open help menu

Clear = clears the terminal window, moving the command prompt to the top

`$nano ~/.bash_profile` = (1) creating a new file in nano called `~/.bash_profile` (2) `~` = users home directory (3) `."` = hidden file (4) The name `~/.bash_profile` is important, since this is how the command line recognizes the bash profile. (5) The text `echo "Welcome, Jane Doe"` creates a greeting in the bash profile. It tells the CMD to echo the string "Welcome, Jane Done" when a terminal session begins (6) `source ~/.bash_profile` = activates the changes in `~/.bash_profile` for the current session, as opposed to close out and restart.

Alias = command lets you create keyboard shortcuts, or aliases, for commonly used commands in the `~/.bash_profile` (ex: `alias pd = "pwd"`) – creates an alias for the `pwd` command so `pd = pwd`

History = history of CMD instructions

Export USER = (environment variables) "Jane Doe" – variables that can be used across commands and programs to hold information about the environment. (ex: `USER = "Jane Doe"`) sets the environment variable USER to a name, usually the name of the computer's owner. (ex: `export`) makes the variable available to all child sessions initiated from the session you are in. This makes the variable persist across programs (ex: `echo $USER`) returns the value of the variable. \*NOTE: `$` is always used when returning a variable's value.

Export PS1 = ">>" - sets CMD prompt variable & exports. Changed default CMD prompt from `$` to `>>`

Echo `$HOME` = environment variable that displays the home directory file path

Echo `$PATH` = variable lists which directories contain scripts

Env = returns a list of environment variables for the current user.

Env | grep PATH = CMD displays the value of a single environment variable. Standard output is “piped” to the grep command. Grep searches for the value of the variable PATH and outputs to terminal

### **Git Commands**

Git init = initialize repository

Git status = check for changed files

Git add [filename1] [filename2] – add one or more files to staging area

Git diff [filename] = see what is different between versions

Git commit -m “Description” = commit to the repository

Git log (last version data) = info on version

HEAD = current commit

Git show HEAD = last commit

Git checkout HEAD [filename] = discard changes in working dir

Git reset HEAD [filename] (m=modification) = unstages file changes in staging area

Git reset [commit\_[SHA]] = (ex: git reset 5680901...\*first 7#'s) = resets to previous commit

Git add . = add all files

Git push origin master = push changes up to the master branch

Git pull = pull down changes

Git branch = which branch am I on?

Git branch [new\_branch] = create a new branch

Git checkout [branch\_name] = switch to branch

Git merge [branch\_name] = merge branch with master

Git branch -d [branch\_name] = delete branch

Git clone [remote\_location] [clone\_name] = clone repository

Git remote -v = list of remotes

Git fetch = bring down recent changes to your local copy

Git merge origin master = merge your master branch with the origin master

Git push origin [your\_branch\_name] = push your branch up to remote

