BASH GUIDE

This guide is a work-in-progress, and may change in the future – medatech@medasf.org

Mission Economic Development Agency | Instructor Eduardo Garcia

WHAT IS BASH?

Bash (Borne Again Shell) is a command line interpreter, meaning it reads and understands the text you write. Bash is used on Linux, MacOS, and BSD operating systems. There is also a lot of software that emulates Bash on operating systems that do not natively support Bash.

HOW DOES BASH WORK?

Bash generally runs on a Command Line Interface (CLI). Command Line Interface examples include Command Prompt on Windows and Terminal on MacOS. Bash works by waiting for input from the user and returns information based on the commands that were inputted. Note for MEDA's coding classes, we use the Git Bash Emulator instead of Command Prompt for Windows machines.

NOTES ON BASH

- The **command name** is the first part of the command.
- **Arguments** are additional information that is provided to the command, and are typed after the command. Arguments are usually surrounded by a space, except for the last argument.
- When providing values (like file names or folder names) that have spaces in them, you **must** surround that value with **double quotes.** For example *mynewfolder* does not need quotes, but *"my new folder"* does, otherwise the CLI may think there are three arguments when you type *my new folder.*
- A directory is another word for folders, the word directory is used very often in CLIs instead of the word folder.
- The prompt is usually a string of text that describes the current user, and the current folder. On different operating systems, it may have additional information as well. At the end of the prompt you should see a $ symbol, this means that the CLI is waiting for your instructions.

- Each CLI window can only be in one directory at a time, but you can open up multiple CLI windows to run other commands without affecting each other.
- If a command freezes and becomes unresponsive (meaning you do not get the prompt ($) back) then you can kill the current process by holding down Control (CTRL) and pressing they C key. This should kill the existing running command and return you to the prompt ($).
- Directories can be chained by adding a slash in between them. This is useful for "jumping" directly to a folder that is deeply nested. For example: Documents/MEDA/Eduardo's Files/

COMMON BASH COMMANDS & EXAMPLES

➔ **ls** – The ls command lets you list the files and folders of the current directory.

- ✔ ls
- ✔ ls -a *shows hidden files and folders*
- ✔ ls -l *shows the files and folders in a list format*
- ✔ ls -la *shows hidden files and folders in a list format*

➔ **cd** – The cd command lets you change your current directory to another directory provided as an argument after the command.

- ✔ cd myfolder/ *moves your CLI to the myfolder directory (the ending slash is optional)*
- ✔ cd myfolder/otherfolder/ *moves your CLI to the otherfolder directory which is inside the myfolder directory which is inside your current directory.*
- ✔ cd ~ *(tilde) moves your CLI to your home directory*
- ✔ cd .. *(double dots) moves your CLI to the current directory's parent*
- ✔ cd ../"Second Folder" *moves your CLI to the Second Folder directory that is located in the parent directory of the current directory*
- ✔ cd "../Second Folder" *same as above but the whole*

➔ **touch** – The touch command creates an empty file with the name provided as an argument after the command.

- ✔ touch myfile.txt

➔ **pwd** – The pwd (print working directory) will display the current directory file path.

➔ **mkdir** – The mkdir command creates an empty directory with the name provided as an argument after the command.

- ✔ mkdir “My New Folder” *because the folder name has spaces in it, it must be surrounded by quotes, otherwise it will create three folders, one for each word.*
- ✔ mkdir Pictures

➔ **mv** – The mv command lets you move files from one location to the other. The first argument is the target file or folder you want to move and the second argument is the destination folder. The target and destination must exist or it will not work as expected.

- ✔ mv myfile.txt some_other_folder *this will move the myfile.txt file to the folder with the name some_other_folder that should be in the same directory.*

➔ **cp** – Copies a file with a new name.

- ✔ Cp myfile.txt official.txt *makes a copy of the myfile.txt with the name official.txt*

➔ **rm** – This removes a file or directory. **Warning:** By default, there will be **no prompt to ask you if you are sure you want to delete it**, and **it is not send it to the Recycle Bin or Trash**, the file/folder is **immediately deleted and there is no undo option**.

- ✔ rm myfile.txt *deletes the file myfile.txt DANGEROUS!*
- ✔ rm -rf temp *deletes the temp directory and all files and folders within it. DANGEROUS!*

➔ **man** – Displays the manual for a specific command.

- ✔ man pwd *displays the manual for the pwd command.*

NOTES