

portfolio_4

2023-02-16

The Command Line

In this portfolio we will cover working with the command line. We are going to skip quickly over the super basics and then start listing some useful command and what they do. After describing the basics we will then dive into the more advanced. This portfolio serves as a good reference point for when working with the command line.

The Basics

ls can be used to list current contents of the directory, there are lots of variations of this command that you can use to list in different ways, a good explanation of all these commands is found here [“https://www.freecodecamp.org/news/the-linux-ls-command-how-to-list-files-in-a-directory-with-options/”](https://www.freecodecamp.org/news/the-linux-ls-command-how-to-list-files-in-a-directory-with-options/).

cp can be used to copy files like so: *cp* .

rm can be used to remove a file like so: *rm* .

Note we need to use the **-r** flag with the above commands if we want to remove or copy a folder with all its contents, it stands for recursive.

mv can be used to move a file like so: *mv* .

cat can be used to print the contents of a file to the terminal. Similarly **head** can be used to print out the start of a file and **tail** for the end of a file, we can then use the **-n** flag followed by a number to change the number of files printed. We can use **tac** to print out each line from bottom to top, **rev** to print out the file in reverse, **uniq** prints out the file without lines that are repeated, **sort** to sort a file alphabetically (by default, can use flags to sort by other things).

We can use `__*__` called wildcard to pattern match, eg. `_cat name*_` will print out all files (in current directory) that start with name and `_cat *.txt_` will print all files (in current directory) that end with the .txt extension.

less can be used to open a “file viewer”, to exit type q.

The following are a bunch of text editors that can be used: **nano**, **micro**, **emacs**, **jove**, **vim**. Let’s look at how we can use nano. If we type *nano* then it will open the file in nano, if the file doesn’t exist it will create the file and open it. Here are some of the basic commands in nano: *ctrl-o* is save, *ctrl-w* is search, *ctrl-x* is used to exit nano.

man can be used to launch the command line manual where we can check for command line commands.

grep is used to search inside files like so: *grep “”* . We can use flags to do thing such as count the number of times the string appears (*-c*), give the line numbers (*-n*), etc.

find lets us search for files like so: *find “”*. Note that `.` is used to represent our current location and `~` is your home directory. Also note that we can use flag **-iname** if we want find to be case insensitive. We can also use **fzf** which stands for fuzzy find and we can use it similarly to find but we can find similar matches without having to use wildcards.

We can use `>` to write the output of a command to a file like so: *head >* , this will write the “head” of the file you name to a new file. We can use `»` to append to a file instead.

We can use multiple commands together by using the pipe operator `|`, eg. `cat / sort` which would feed the print out of the file you choose to the sort function before printing it to the terminal.

More Advanced

Now we will move onto looking at some more “advanced” things we can do with the command line. This will include working with remote computers and shell scripts.

To work