Review - Command Line / Git / GitHub

Command Line Interpreter

A Command Line Interpreter is a program that allows a user to enter commands that are then executed on the operating system. They quite literally interpet commands to the underlying operating system.

These applications are also know as: CLI, command language interpreter, console user interface, shell, command line shell

This differs from the Graphial User Interface(GUI) that is commonly used to interact with the file system. With the CLI there are no icons, buttons, or interactions with the mouse. All interactions are done through the keyboard.

While the GUI is much easier to use, the CLI is much more powerful. Through the CLI we can automate commands to the OS through scripts, as well as, gain access to more system commands that what is offered through the GUI

Simply put, the command line allows a user to navigate the filesystem and run built-in programs or custom scripts.

Navigating Filesystem Structure

A computer's filesystem organizes the data stored by a computer, so that it can be easily retrieved by the user. Files are typically represented by a tree-like structure, in which any parent directory can have any number of children.

The root directory is then found at the base of the tree.

Common Commands

List ls

The shell command ls is used to list the contents of directories. If no arguments are given, it will list the contents of the current working directory.

```
$ ls Desktop
resume.pdf
photo.png
```

Print Working Directory pwd

The shell command pwd displays the file path from the root directory to the current working directory.

```
$ pwd
/Users/sonny/Downloads
```

Make Directory mkdir

The shell command mkdir can be used to make a new directory in the filesystem according to its argument. If a file path is given, the new directory will be placed at the end. Otherwise, it will create a new directory in the current working directory with the name given.

```
$ mkdir new-directory
$ ls
old-directory    new-directory
```

cd Change Directory

The shell command cd can be used to move throughout the filesystem of a computer. It accepts a variety of arguments:

- Full file paths.
- Names of children of the current directory.
- .. the parent of the current directory.

```
$ cd some-directory
$ cd ..
```

touch Create New File

The shell command touch creates a new file in the current working directory with the name provided.

```
$ touch secret-file.txt
```

View and Change the File System

cp Copy

The shell command cp is used to copy files or directories.

The basic argument structure is cp source destination, where the source is the file/directory to copy to the destination file/directory.

```
$ cp file1 file1_copy
$ cp file1 file2 destination_folder
```

Command Options

Options can be used to modify the behavior of shell commands. Shell command options are commonly represented by a single letter preceded by a -. For example, -l, -a, and -d could all be options that follow a shell command.

mv Move

The shell command mv is used to move a file into a directory. Use mv with the source file as the first argument and the destination directory as the second argument.

```
$ mv index.html website/
```

rm Remove

The shell command rm is used to delete files and directories. The -r flag deletes a directory and all of its files and directories (rm -r).

```
$ rm –r bad_selfies
```

ls List Command Options

The shell command ls is used to list the contents in a directory. It can be combined with the following command options:

- -a: lists all contents, including hidden files and directories.
- -l: lists all contents, in long format.
- -t: lists all contents, by the time they were last modified.

```
$ ls -a
$ ls -l
$ ls -t
```

Git

A versioin control system

Create a new repository

```
git init
```

checkout a repository

```
git clone /path/to/repository
```

This creates a copy of a local repository

```
git clone username@host:/path/to/repository
```

Creates a copy of a remote repository

Adding files to staging

```
git add <filename>
```

```
git add *
```

Commiting staged files

```
git commit -m "commit message"
```

Pushing changes

```
git push origin master
```

Adding a remote repository

```
git remote add origin <server>
```

GitHub

Git is a command-line tool, but the center around which all things involving Git revolve is the hub—GitHub.com—where developers store their projects and network with like minded people.