

Notes

- Bash commands
- Vim shortcuts
- Compiled vs interpreted programming language
- Shell scripting

Bash commands:

pwd: This command prints the current working directory.

ls: This command lists all the files in the current directory.

cd: This command allows you to move to another directory.

touch: This command allows you to create new files

cp: This command allows you to copy files

mv: This command allows you to move/rename files.

rm: This command allows you to remove a file

date: This command shows you the current date.

cal: This command shows you the calendar.

uptime: This command shows you how long the computer has been running.

uname: This command shows you information about the kernel.

ln -s [src] [dest]: This command creates a symbolic link.

wget: This command allows you to download a file from the web.

head -n[num]: This command allows you to get the first “n” number of lines, from the beginning of a file.

tail -n[num]: This command allows you to get the first “n” number of lines, from the end of a file.

grep: This command allows you to search a file via a search pattern.

cat: This command prints everything in a file.

ps -ef: This command allows you to see all running programs on the machine.

man: This command allows you to look up any bash command.

top: This command displays sorted information about processes.

less: This command allows you to move forward and backward through a file.

:[search term]

:[enter]

:[shift] < or >

more: This command allows you to move backward through a file.

wc -l **or** wc -w **or** wc -c: This command shows a file's count based on the number of words, lines, and characters.

| [piping]: This allows you to take the input of output of one command as input to another.

>:

>>:

<:

<<:

Vim shortcuts:

Compiled vs interpreted programming language:

Compile

A compiled language takes the source code and with the help of its compiler, converts it into an executable binary file, typically in the form of a .exe or a .o/ file. This file just contains the assembly code based on the architecture it was compiled on. Compilers are also smart and can provide optimization to code during the translation.

Interpreted

An interpreted language translates the source code into assembly language via its interpreter line by line as it is run. No need to compile. This means that many interpreted programming languages can be run on any computer. But because of those extra steps in the translation, interpreted languages are slower than compiled languages with few optimizations possible.

Shell scripting:

```
#!/usr/bin/bash
```

Make a file executable:

```
chmod 777 [script file]
```

To run file:

```
./<fileName>
```