

# Getting Started With Cpsc (Advanced)

Hosted by Jarrett Spiker

# Advanced?

- Assuming that everyone has done at least a year of CPSC already, or has a strong base knowledge.
- If not, there is a Beginners session at MS 160
- Feel free to ask questions

# Goal

Provide some tools to help with the rest of you CPSC classes and/or degree.

# BASH command line

The default in Unix systems.

- Mac
- Linux (Ubuntu, Fedora)

If you are on Windows

- I'm so sorry
- There's always partitioning
- There are tools like Cygwin which will allow you to use bash
- There is "Bash for Windows" terminal available for 64-bit Windows 10 Anniversary Edition

# BASH concepts

~ is your home directory

/ is the root directory

. is the current directory

.. is the above directory

| pipes output from one command to the input of another

> writes output to a file

# Basic BASH commands

- `cd <destination>`
  - Change directory
- `ls`
  - List contents on current directory
  - You can add a directory to change which directory will be listed
- `cp <source> <destination>`
  - Copy file from one location to another.
  - Using `-r` flag copies recursively
- `mv <source> <destination>`
  - Move a file/directory from one location to another
  - Also works as a rename operation
- `rm <target>`
  - Remove a file/directory
  - `-r` flag makes the deletion recursive. `-f` skips asking for confirmation

- mkdir <name>
  - Creates a directory

# “Advanced” BASH commands



# chmod

Changes access rights to a file/directory.

```
“chmod 777 ~/myFile.txt”
```

You can pass a 4 digit octal number.

<??><owner access><group access><world access>

4 - read

2 - write

1 - execute

“chmod 736 file” gives all access to owner, write and execute privileges to group, and read and write access to world

You can pass letter options to change specific bits

```
chmod <who><change><permission>
```

Who is either “u”(user), “g” (group), “o” (others/world). It defaults to all.

Change is either “+”, “-”, “=” (defaults)

Permission is “r” (read), “w” (write), “x” (execute)

So you can give everyone execute privileges with “chmod +x file”

Or take away group and world write privileges with “chmod go-w file”

# grep/find

grep and find are search tools.

“grep Jarrett file.txt” will return all of the lines in file.txt which contain the word “Jarrett”

-r option can be used to search all of the files in a directory

“grep -r Bar testDir/” will find all of the lines which contain Bar in the testDir directory

There are a bunch more options. “man grep” is useful.

find locates file names.

``find . "Test*"`` finds all the files in the current directory which begin with "Test"

Again, there are a lot of options, it's worth looking at "man find" (or just googling it)

# SSH

Connect to a server. This will be needed in some second/third year Cpsc courses.

“ssh <username>@<server name>”

The U of C servers are:

csa.cpsc.ucalgary.ca

csb.cpsc.ucalgary.ca

etc....

Close with “exit”

# SCP

Copy a file from a server to your local machine or vice versa

“scp <username>@<server name>:<file path> <local destination>”

Or vice versa to copy from your local machine to the server.

# Env Vars

Environment variables define locations on your system, or just work as nice shorthands.

`echo $PATH` will print the contents of the variable `PATH`

`Export PATH=<stuff>` will set `PATH` to `stuff` within the current terminal (NOTE DONT DO THIS. Use “`export PATH=$PATH:<stuff>`” to append to the path).

To make changes permanent you write the `export` command to the `~/.bashrc` file (Linux) or the `~/.bach_profile` file (Mac)



# Useful env vars

`JAVA_HOME` sets which JDK java processes will run with.

`PATH` is the set of paths which will be searched for executables when you enter them in the terminal.

Gradle, Maven, IDEA all have `_HOME` variables as well

# Internships

# Source Control

# Public Profile

Stack overflow

Github (Bitbucket)

Personal Website

Editors

# VIM/EMACS

- Light weight
- Fast
- Built in

Alternatively, the terminal has build in editors

- pico
- nano

# IDES

Integrated development environments.

Which one is largely personal preference.

Atom

- Extremely lightweight
- Most features and language support comes from plugins

Visual Studio

- Microsoft IDE
- Strong C#, C++, ect. support
- Very large, RAM intensive program

## IntelliJ

- Free community edition, the paid edition is free for students
- Basis for several other IDEs
  - Android Studio
  - Web Storm
  - Pretty much anything else IDEA does
- Attempts to take care of a lot of the overhead that comes with a large solution for the user
  - More or less succeeds



## Eclipse

- Based on plugins
- Support for a lot of languages (with varying quality)
- Free and open source
- Can have a decent learning curve
- Some overhead for more complicated projects

# Eclipse Demo