Intro to the Command Line

Logan Yokum

What is a Terminal (Emulator)?

- **Text I/O** environment
  - Input text into the computer through the keyboard
  - Receive text output on display
- Nowadays, used to run a *__shell__*

The Shell

- In short, a program to **run other programs**
  - similar to Start menu (Windows) or Launchpad (Mac)
- Most common is **Bash**

The Shell (Continued)

- Allows you to save sequences of programs in **scripts**
- Provides support for **connecting** programs together in **pipelines**
- Additional Functionality
  - Tab completion for programs and files
  - History (use Up/Down arrow keys)

Shell Environment - Directories

- You are always in a directory - similar to a file manager
- When you first log in, you start in your **home directory**
- You can reference files **relative** to the directory you are in
  - `./my_file.txt` or `my_file.txt` - file in current directory
  - `./my_dir/my_file.txt` or `my_dir/my_file.txt` - file in directory within current directory
  - `../my_file.txt` - file in directory above current

Important Directories

- `~` or `/home/username`: your home directory
- `/`
  - the root directory (base of the filesystem)
- `.`
  - alias for your current directory
- `..`
  - alias for the directory above the current directory
  - if your current directory is `/prev/curr/`, then `..` would be equivalent to `/prev/`

Permissions

- Restricts access to files to certain users
- Levels
  - User
  - Group
  - Global/Everyone

Permissions (Continued)

- Types
  - **Readable** - can I view the contents of this file?
  - **Writable** - can I change the contents of this file?
  - **Executable** - can I treat this file like a program and run its contents/code?

Unix Philosophy - Understanding the Command Line

1. Write programs that **do one thing** and do it well
2. Write programs to **work together**
3. Write programs to handle **text streams**, because that is a universal interface

Command Anatomy

- General Format (Built-ins)

```
command_name (-options) (args)
```

- General Format (User Programs)

```
/path/to/command_name (-options) (args)
```

Command Examples

- Commands: `cd` (change directory), `ls` (list directory contents), `cp` (copy data from one file to another)
- Options: `-v` (“verbose”), `-o` (“output file”), `-h`/`--help` (“display usage”)
- Arguments: filenames, strings of text

Man (“Manual”) Command

- If you want to find out info on **how to use** a command, you can use the `man` command
  - example: `man ls`
- VERY USEFUL for learning how to use commands

Input-Output of Commands

- While running, every command can **accept text input** and **produce text output**
  - Text input can be entered by typing it and then pressing ENTER
  - Text output is displayed **on the terminal**
- As we'll see in the workshop, we can **connect** the output of one command to the input of another to make a **pipeline**