

Class 1: Getting started

Overview

1. Announcements
2. Review
3. Q&A
4. Extras
5. Basic assignment

Announcements

- Basic - Intro released!
 - Due at the end of September 27
- No class next week
 - Next class will be September 29

Review

What is a terminal?

- I/O device/peripheral
- Sends text to a computer
- Receives text from computer to display

What is a shell?

- Program that the user goes through to interact with the OS
- Graphical and command line varieties
- Command line shells interpret user-entered text and do something with it
 - Navigate through files and directories

Review

File paths

- Describe how to get to a file/directory by traversing directories
- `/` is the directory separator
- "Root" is the highest level/topmost directory
- Absolute paths: start with a `/`
- Relative paths: don't start with a `/`
 - Implicitly start from the current directory
- "Special" directories
 - Each directory contains these
 - `.` leads to the same directory
 - `..` leads to the parent of the directory

Review

Basic commands

- `man`: "manual pages"
- `pwd`: "print working directory"
- `ls`: "list directory"
- `cd`: "change directory"
- `mv`: "move"
- `cp`: "copy"
- `touch`
- `rm`: "remove"
- `cat`: "concatenate"

Review

Command structure

```
$ <command> <argument 1> <argument 2> <argument 3>
^      ^      ^      ^
|      |      |      |
|      |      |      |-- programs are provided these to
|      |      |      |      interpret (remember argc and argv[ ]?)
|      |      |      |
|      |      |      |-- words separated by whitespace
|      |      |
|      |      |-- certain things are actual programs, certain things
|      |      |      are handled by the shell ("built-ins")
|
|-- this is called a "prompt" and can take many forms
```

Review

Pipes and redirection

- `|` passes output from one command to the input of another
- `>` sends the output to a specified file instead of a terminal
- `<` reads input from a specified file instead of a terminal

Q&A

Extras

- Command line text editors
- Terminal pagers
- Shell scripts
- How WSL files work

Activity 1

- Count the number of items in a directory
 - `ls` will list out items in a directory
 - `wc -l` will count the number of lines of given *input*
 - `ls` will change its behavior to output items on lines when it's not printing to a terminal..

Activity 2

- Using only your terminal, create a Hello World C++ program

Basic assignment

Addenda

Environment

- Terminal emulator: **Alacritty**
 - Former rxvt-unicode user until I learned how bad its font handling was
- Shell: **Zsh**
- Window manager: **i3-gaps**
- Compositor: **picom**
 - Does window transparency effects
- Notification server: **dunst**
 - Displays notifications

Software

- Editor: **Vim**
- System monitoring: **htop**
- Notetaking: **Xournal++**