

CSE 374: Programming Concepts and Tools

Eric Mullen
Spring 2017
Lecture 1: Welcome

Welcome

- We have 10 weeks (28 lectures): You'll learn a lot
 - Command line tools/scripts to automate tasks
 - C programming (like Java but totally different)
 - Programming Tools (thinking, gdb, valgrind)
 - Basic Software Engineering
 - Basic Concurrency
- This class will introduce a lot of things, too fast for mastery

Today

- In class
 - Course Mechanics
 - Overview/Plan
 - Dive in
- Before next time
 - Start on HW0, due Friday night
 - Set up machine, introduce yourself to us
 - Start reading Pocket Guide

Course Mechanics

Class Structure

- 3 Lectures per week, no lecture May 29
- About 7 homework assignments
- Midterm on April 28 (during lecture)
- Final on June 7 (2:30pm, this room)

Me

TAs

cse374-staff@cs.washington.edu

You

- Who has used Linux before?
- Who has written a C program before?
 - More than 100 lines?
- Who has used Git? Github?

Administrivia

- Office hours: 11am on Tuesdays in 218 (me), 5 days a week by TAs (also in 218), posted shortly
- Course website: <http://cs.washington.edu/374>
- Syllabus: on website
- Most homework due Thursday night at Midnight (HW0 is unique).

Grading Philosophy

- Mostly you'll be turning in programs
- There are 2 main ways to interact with programs
 - Run them
 - Read them
- We care about both: make sure we can do both

Laptops/Phones in Class

- Hard to just outright ban them, but...
- Course Policy:
 - Only flat on the table screens
 - Only for class related activities (no FB etc...)

Academic Honesty

- Turn in your own work
- Don't turn in anyone else's
- More detailed policy online
- When in doubt just be honest (tell us who did what)

Deadlines

- Late assignments are annoying to us, and can leave you trailing the class
- However, life happens
- You get 4 late days for this class. You can use them whenever you need, but a max of 2 per assignment.
- 10 minutes late = 24 hours late
- Once your late days are gone, late work is worth no credit

Course Overview

What is this class, really?

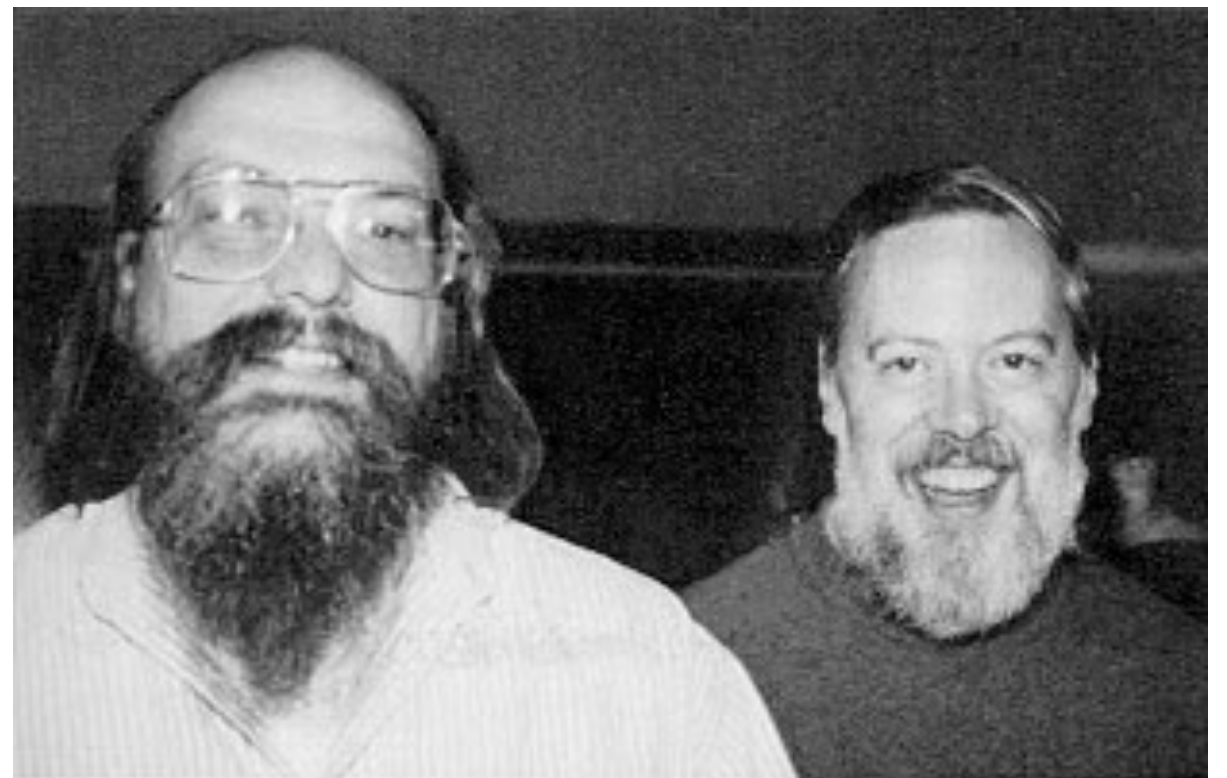
- Great quote from Hal Perkins:

There is an amorphous set of things computer scientists know about and novice programmers don't. Knowing them empowers you in computing, lessens the "friction" of learning in other classes, and makes you a mature programmer.

- Just as much ***How to Learn*** as ***What we Learn***

Almost a History Class

- C language is from ~1971
- Unix is from ~1971
- Bash is from 1989, concepts date back to ~1971



Ken Thompson and Dennis Ritchie

5 General Areas

1. The Command Line: *Never use a mouse again*
 - Text based computing
 - Allows for automation
 - We'll use Linux and Bash, but concepts are broader than these

5 General Areas

2. C (and C++)

- *The* programming language for low-level systems, networking, operating systems, embedded devices, etc...
- Manual resource management
- C trusts the programmer: up to you to not make mistakes
- Think before you write, and test often

5 General Areas

3. Programming Tools

- In reality it's far more complicated:
 - Compilers (vs interpreters)
 - Debuggers
 - Linkers
 - Recompilation Managers
 - Version Control
 - Profilers



5 General Areas

4. Software Development Concepts: How to write a million* lines of code

- Testing Strategies
- Team Programming
- Software Specifications (and their limits)

*No you won't actually write a million lines of code for this class

5 General Areas

5. Concurrency Basics

- Computers have more than one processor today. How does that affect programming?
- Whole new classes of bugs (even harder to debug)
- How to avoid those bugs and write correct concurrent programs

Dive In

Operating System (OS)

- We're going to use Linux
- Linux is a modern, Unix-like operating system
- More closely related to MacOS than Windows (but distinct from both)
- More common than you think:
 - Android built on it
 - Many servers run it

Filesystem

- The filesystem is a tree
 - (actually a directed graph)
 - Top node is / (called root)
 - Interior nodes are directories, files are leaves
 - Paths look like: `/home/emullen/hello.txt`

Shell

- Users log into a machine, which gives them a shell
 - They have permissions to access certain files/directories
 - They have a “home directory” somewhere
 - They can run programs. A running program is called a process (or could be more than one)

Klaatu

- Our preferred way for you to use Linux is to log into Klaatu (klaatu.cs.washington.edu)
- If you are registered for the course you will get login details soon
- In order to work on Klaatu, you'll need an internet connection

Demo!

```
ssh emullen@klaatu.cs.washington.edu
script lecture1.script
ls
pwd
mkdir lemon-ginger
ls
cd lemon-ginger
ls
cd ..
exit
more lecture1.script
scp
```

VM

- If you need to work without an internet connection, you can install a VM - you should have received an email with credentials within the last 24 hours
- Setup is a bit more complicated, and the files are quite big (~10 gigs once everything is installed)
- We do officially support this option, we just encourage you to use Klaatu if possible

File Manipulation

- It can be hard to say goodbye to drag and drop files
- But you can do everything with text
- Just like a file explorer, the shell always has a current directory
- Some key commands are useful to memorize:
 - `ls`, `cp`, `mv`, `rm`, `cat`, `cd`, `pwd`
- Current dir is `.` Up one is `..`

Why?

- You hate switching from keyboard to mouse
- You are actually from the 70s
- You want easier logging
- You want easy instructions
- You want to be able to script everything

Help!

- Bad news: program names are short
 - `ls`, `cp`, `mv`, `at`, `cd`, `rm`
- Good news: the computer knows more
 - `man <command>` for more information (stands for manual)
 - `info <command>` can work too (but start with `man`)
- Read your Linux Pocket Guide! It does help.

More Programs

- `less`
- `chmod`
- `mail`
- `exit`
- `echo`

Homework 0

- It will go out by the end of the day
- It's more of a checkpoint than an actual homework
- Don't waste late days on it
- If you are having trouble, email the staff (cse374-staff@cs.washington.edu)