

CSE 374: Programming concepts and tools

Spring 2019

Instructor: Megan Hazen



Where are you?

CSE 374 is a practical course about

- Command line tools and scripts to automate tasks
- C programming with explicit memory management
- Tools for programming
- Software engineering practice
- Basics of concurrency

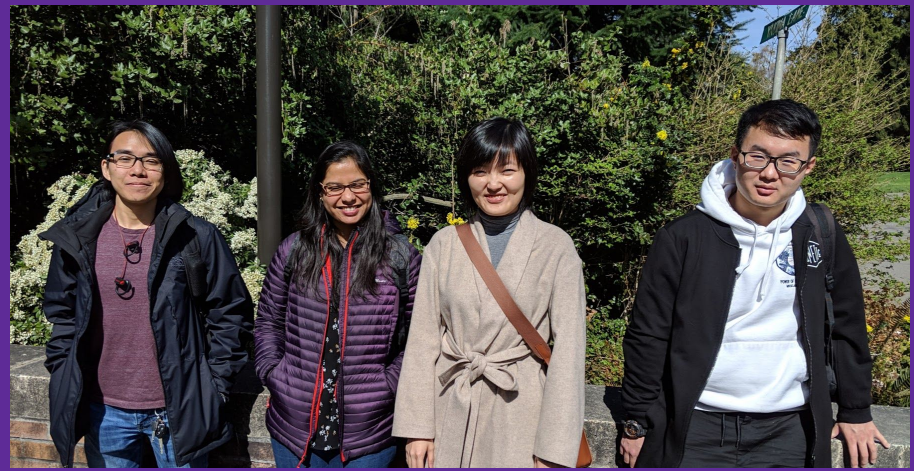
374 is also

- An introduction about how to learn what you want to know to move forward.

Who are we?

Your instructor: Dr. Megan Hazen

Your TAs: Andrew Tran, Soumya
Vasisht, Cynthia Zhang, Sujie Zhou



Who are you?

90 Students

What are your disciplines?

Who has used Linux?

Who has written a C program?

Who has used Gitlab/Github?

Who is nervous about this course?

Today

My job:

- ❑ About this course
- ❑ Schedule and homeworks
- ❑ Resources
 - ❑ TAs
 - ❑ Links on homepage
 - ❑ Canvas Discussion
- ❑ Course Rules
- ❑ Get started on *command shell*

Your job:

- ❑ Get a jump on HW0
 - ❑ Part A due Wed., Part B due Thurs.
- ❑ Start reading the Linux pocket guide
- ❑ Try things - log in, explore, test commands

If you are still trying to add the course:

Come to class

Keep an eye on the roster for changes this week

Complete HW0: Please email cse374-staff@cs.washington.edu for details

Course requirements

Lecture Monday, Wednesday, Friday 9:30-10:20

~ 8 homework assignments

~ 3 on shells and scripting

~3 on C programming

1 is a team project

1 midterm

1 final exam

Extra credit: minimal, but exists for more challenge

What to expect

You are responsible for material on Canvas and the webpage.

Assignments may be more open ended than you are used to - you'll need to figure out whether you are doing the right thing.

Learning how to learn is part of the plan

Get used to looking at documentation and searching for answers

Plan to understand, not just re-create

Tinker -expertise comes from experience

Course Resources

Instructor and TAs

Office hours every day: see syllabus

Use office hours to get ‘unstuck’

Canvas Discussion Group

Discussion for each assignment plus more!

Communications

Email [cse374-staff@cs](mailto:cse374-staff@cs.washington.edu)

Course website

<https://courses.cs.washington.edu/courses/cse374/19sp/index.html>

Books, ‘man’ pages, Google

Use Google as a starting place, be sure you understand

Recreate on your own; don’t just cut-and-paste

Tinker

Try things, experiment with new tools

Ask questions early and often!

Lectures

Come to pay attention: **No Laptops / Phones** *except potentially as noted in advance

Plan to learn big picture approaches and concepts

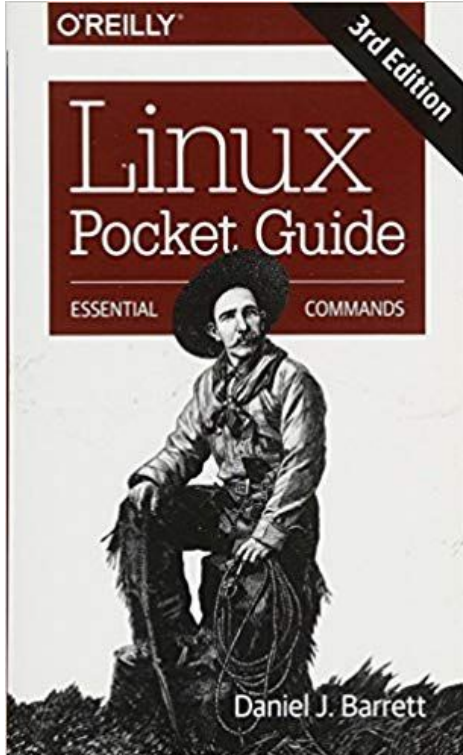
Jot down key words and ideas to look up later

Advice: plan to be an active learner

review notes, look up documentation, try ideas in the same day

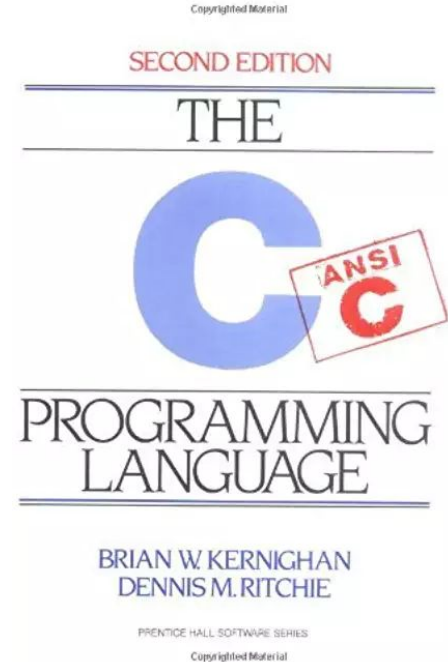
ask questions early and often

Books



Web searches provide great starting places, and good short reminders

For context and understanding nothing beats a book



Academic Integrity

<https://courses.cs.washington.edu/courses/cse374/19sp/integrity.html>

Policy on the course webpage

Do your own work, be ready to explain it

Integrity is everything - have high standards

Unless otherwise specified all work in this course is independent

Do not share code; discuss approach

When in doubt - ask and be honest

Late Policy

Turn things in on time
Plan ahead

You have 4 'late days' this quarter
(recorded on Canvas)

Counted in units of 1 day

No more than 2 'late days' per
assignment

Group project requires that both
partners use the requisite 'late days'.

Save 'late days' if possible

(Contact instructor for truly exceptional circumstances; before
deadline if possible.)

Major Ideas of 374

1. Command line and scripting tools
 - a. Linux, Bash, automation
2. C programming and memory management
 - a. Lower level than Java
3. Tools for programming
 - a. Compilers, debuggers
4. Software development and testing
 - a. Software specs, tests, and teamwork
5. Concurrency
 - a. Using multiple processors at once

Getting Started with Linux

Use a virtual machine

Or

Log into Klaatu

<https://courses.cs.washington.edu/courses/cse374/19sp/linux.html>

When I log-in I get a shell

- A shell - a text based interface to the computer
- Specifically 'bash'

Linux and Shells

Text is efficient - typing is fast, there are no graphics to pass around, and logging is easy when recording text. Scripting makes it easy to automate textual interfaces.

Linux does have a GUI interface

Windows (MacOS) do have shell interfaces

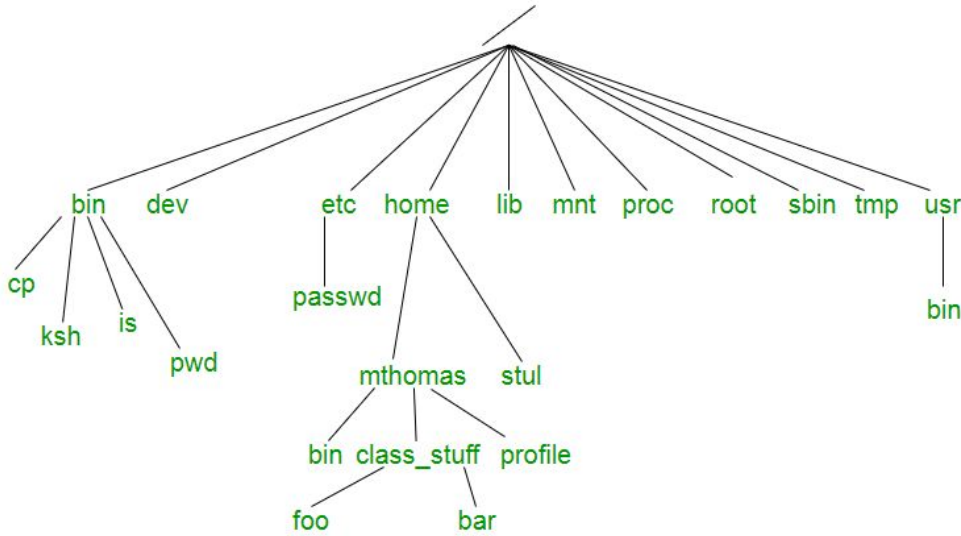
Most power users use both, depending on application.

Notes:

You could use any distribution of Linux that is up-to-date. Using CentOS through CSE ensures consistency.

All Assignments will be graded as run on CSE software

File Systems



- ❑ File systems are trees
- ❑ (or directed acyclic graphs)
- ❑ A file (or directory) is specified by its path from the top ('/')
 - ❑ Can be specified absolutely (entire path),
 - ❑ Relatively (from current location)
 - ❑ This directory './'
 - ❑ One directory up '../'
 - ❑ You have access to your 'home' directory ('~')

Get Going!

HW0

1. Look at the course webpage & canvas
2. Review syllabus
3. Respond to
4. o Welcome discussion
5. Look for email re: Klaatu log-in
6. Log in, and try today's commands
7. Explore the Linux pocket guide (available on Kindle for \$2)
