CS107, Lecture 1
Welcome to CS107!

reading:

Course Syllabus

Bryant & O'Hallaron, Ch. 1 (skim)

Honor Code and Collaboration Page

Asking Questions

- Feel free to raise your hand at any time with a question
- If you are more comfortable, you can post a question in the Ed forum thread for each day's lecture (optionally anonymously)
- We will monitor the thread throughout the lecture for questions

Visit Ed (or access via Canvas):
https://edstem.org/us/courses/56714/discussion

Today's thread:
https://edstem.org/us/courses/56714/discussion/4668127

Plan For Today

- Introduction
- CS107 Course Topics
- CS107 Course Policies
- Unix and the Command Line

Plan For Today

- **Introduction**
- CS107 Course Topics
- CS107 Course Policies
- Unix and the Command Line

Teaching Team


Nick Troccoli

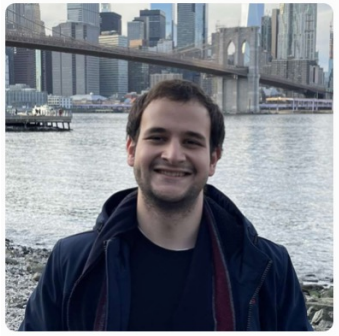

Ola Adekola (Head TA)


Andreea Jitaru


Luke Babbitt


Michael Granado


Sophie Andrews


Arman Aydin


Carolina Borbon Miranda


Kamran Ahmed


Trevor Carrell


Zoe Kaputa

About Nick Troccoli (troccoli@stanford.edu):

- Lecturer in CS, taught CS106X, CS107, CS110, CS111
- Stanford BS/MS (coterm) in CS
- Systems track undergrad, AI track grad

Companion Class: CS107ACE

- **CS107ACE** is an extra 1-unit “Pathfinders” or “ACE” section with additional course support, practice and instruction.
- Meets for additional weekly sections and has additional support
- Entry by application – see the course website for details: **cs107ace.stanford.edu**
- In addition to all normal CS107 requirements such as lab

Jessica Yu (jmyu@stanford.edu)

A Wonderful Community ❤️

I like to crochet!

I am scared of birds.

I can jump over a chair

I have two cats

I dance with Stanford XTRM K-pop!

I did not write a single line of code before coming to Stanford two quarters ago.

I cannot distinguish cheerwine and dr. pepper

I can whistle a fantastic Twitter notification imitation.

*Quotes taken from course welcome survey

Course Website

cs107.stanford.edu

*lecture videos / lecture grades on Canvas

Plan For Today

- Introduction
- **CS107 Course Topics**
- CS107 Course Policies
- Unix and the Command Line

What is CS107?

The CS106 courses (or equivalent) provided you with a solid foundation in programming methodology and abstractions (variables, functions, data structures, pointers, recursion, classes, algorithms).

CS107 follows on this to build up and expand your breadth and depth of programming experience and techniques to show you how machines really work.

CS107 key question: *how / why*?

CS107: How/Why?

The CS106 series taught you how to solve problems as a programmer. CS107 goes a level deeper to understand the **how** and **why**:

- **How** is data in our program really represented?
- **How** does heap memory work?
- **How** does a computer know how to run the code we write?
- **How** does a program map onto the components of computer systems?
- **Why** is my program doing X when I expected it to do Y?

Understanding computing at this level demystifies how these seemingly-complex systems work and can aid future projects you work on.

Course Overview

1. **Bits and Bytes -** *How can a computer represent integer numbers?*
2. **Chars and C-Strings -** *How can a computer represent and manipulate more complex data like text?*
3. **Pointers, Stack and Heap –** *How can we effectively manage all types of memory in our programs?*
4. **Generics -** *How can we use our knowledge of memory and data representation to write code that works with any data type?*
5. **Assembly -** *How does a computer interpret and execute C programs?*
6. **Heap Allocators -** *How do core memory-allocation operations like malloc and free work?*
7. **Ethics, Privacy, Partiality and Trust -** *How do we act responsibly in maintaining security, protecting privacy, and ensuring warranted trust in the systems we build and maintain?*

CS107 Learning Goals

The goals for CS107 are for students to gain **mastery** of

- writing C programs with complex use of memory and pointers
- an accurate model of the address space of C programs
- the compile/runtime behavior of C programs

to achieve **competence** in

- translating C to/from assembly
- writing programs that respect the limitations of computer arithmetic
- identifying bottlenecks and improving runtime performance
- working effectively in a Unix development environment
- using ethical frameworks and case studies to inform decision-making

and have **exposure** to

- understanding compilers and disassemblers
- a working understanding of the basics of computer architecture

CS107 and Programming Experience

- We hope that CS107 can help further develop your programming experience and comfort with programming.
- CS107 focuses heavily on **debugging** and getting to the root of why something is happening.
- Across assignments, we will be emphasizing how to become a better debugger, how to write better code, and how to further your software development skills.

Plan For Today

- Introduction
- CS107 Course Topics
- **CS107 Course Policies**
- Unix and the Command Line

Course Syllabus and Schedule

cs107.stanford.edu/syllabus

cs107.stanford.edu/schedule

SCPD Students

cs107.stanford.edu/scpd

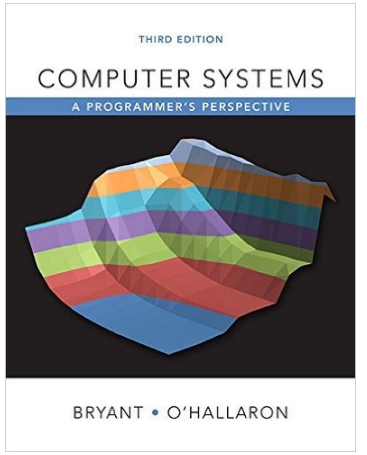
Getting Started Guide

cs107.stanford.edu/getting-started.html

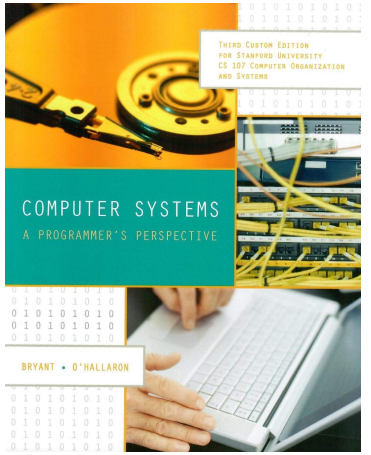
(on course website under ”handouts”)

Textbook(s)

- *Computer Systems: A Programmer's Perspective* by Bryant & O'Hallaron, **3rd Edition**
  - **3rd edition matters** – important updates to content
  - Stanford Library has generously scanned **all** readings for CS107 under "fair use" (private study, scholarship, research). [**Canvas -> Files**]. Please do not distribute.
  - If you want more context, you may want to purchase a full copy
- A C programming reference of your choice
  - *The C Programming Language* by Kernighan and Ritchie (free link on course website Resources page)
  - Other C programming books, websites, or reference sheets



Full textbook



CS107 full chapters

canvas CS107-specific readings

The textbook (and C programming references) are **very** good resources in this course, especially post-midterm!

Course Structure

- Lectures: understand concepts, see demos
- Labs: learn tools, study code, discuss with peers  **Great preview of homework!**
- Assignments: build programming skills, synthesize lecture/lab content
  - **assign0**: due next Monday (covers today and part of Wednesday's lecture)

Grading

******	52%	Assignments
*	8%	Lab Participation
*	5%	Lecture Points
**	15%	Midterm Exam
**	20%	Final Exam

Read our full course policies document:
https://cs107.stanford.edu/syllabus.html

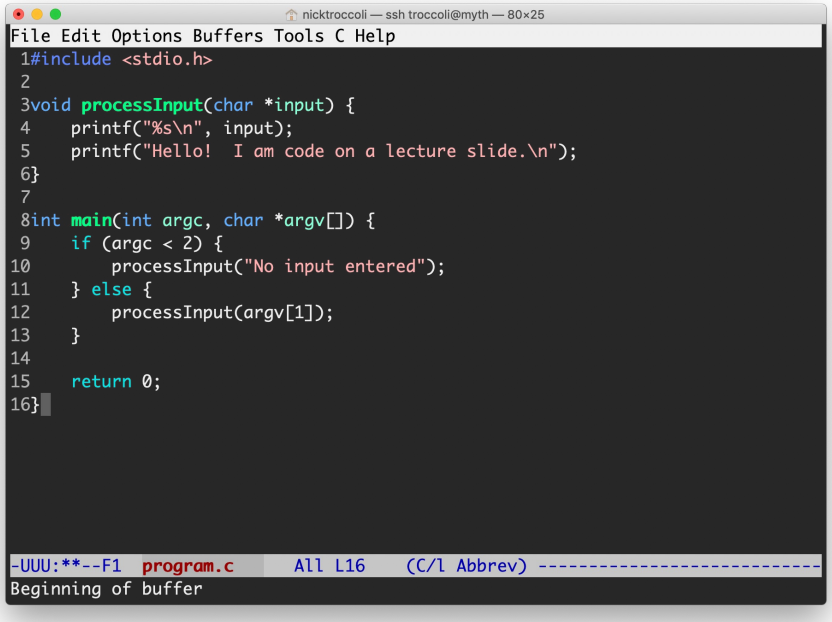
Grading

******	52%	Assignments
*	8%	Lab Participation
*	5%	Lecture Points
**	15%	Midterm Exam
**	20%	Final Exam

Read our full course policies document:
https://cs107.stanford.edu/syllabus.html

Assignments

- 7 programming assignments completed individually using **Unix command line tools**
  - Free software, pre-installed on Myth machines / available on course website
  - We will give out starter projects for each assignment
- Graded on **functionality** (behavior) and **style** (elegance)
  - Functionality graded using *automated tools,* given as point score – no TA review
  - Style graded via *automated tests* and TA code review, given as bucket score
  - Grades returned via course website



The Style Bucket System

+	An outstanding job; could be used as course example code for good style.
ok	A good job; solid effort, but also opportunities for improvement.
-	Shows some effort and understanding but has larger problems that should be focused on.
- -	Shows many significant issues and does not represent passing work.
0	No work submitted, or barely any changes from the starter assignment.

Assignment Late Policy

- **Start out with 5 "free late days"**: each late day allows you to submit an assignment up to 24 additional hours late without penalty. (No late days permitted for the first or last assignments)
- **Hard deadline 48 hours** after original due date
- Penalty per day after late days are exhausted (1 day: 80% cap; 2 days: 60% cap)
- Late days are "pre-granted extensions" – additional extensions for exceptional circumstances must be approved by the **Head TA** and received **prior to the on-time deadline** or as soon as possible if extenuating circumstances occur later, or extenuating circumstances prevent reaching out prior to the deadline. Please communicate with us! We are here to accommodate you as much as possible.

Question Break!

What questions do you have about the overall course goals, textbook or assignments?

Grading

******	52%	Assignments
*	8%	Lab Participation
*	5%	Lecture Points
**	15%	Midterm Exam
**	20%	Final Exam

Read our full course policies document:
https://cs107.stanford.edu/syllabus.html

Lab Sections

- Weekly 1-hour 30-minute in-person labs led by a CA, starting *next* week, offered Wednesdays through Fridays.
- Hands-on practice in small groups with lecture material and course concepts. Designed to act as prep/preview of homework!
- Graded on attendance + participation
- Lab preference submissions open **Tuesday 4/2 at 12PM PST** and **are not first-come first-serve**. You may submit your preferences anytime until **Saturday 4/6 at 5PM PST**. Sign up on the course website.

Grading

******	52%	Assignments
*	8%	Lab Participation
*	5%	Lecture Points
**	15%	Midterm Exam
**	20%	Final Exam

Read our full course policies document:
https://cs107.stanford.edu/syllabus.html

Lecture Recordings

- Because CS107 is offered on SCPD (for professional development students) this quarter, the course is being recorded for later viewing.
- See the calendar page (or lecture dropdown) on the course website for slides and lecture code. Materials are posted the evening before each lecture.

Lecture Points

At the same time, staying current with the material is essential to your success this quarter! Our primary goal is to incentivize staying current with lectures to enable you to **start early on assignments** and **have the material you need to work through section material.**

Two ways to get credit for a lecture:

1. Get 100% for the lecture if you attend in person and respond to all Poll Everywhere polls! (regardless of correctness).
2. Watch the recording and complete a corresponding Canvas quiz by **30 min prior to the start of the next lecture**. Quizzes will take about 10-15min each and are graded for correctness but permit 3 total attempts.

Lecture Points

We will also **drop the two lowest lecture scores**, which is intended to cover for lectures where you are both unable to attend in person and also unable to complete the quiz by the next lecture.

Further excused misses are granted by the **Head TA** only in cases where you have already used your 2 excused misses for extenuating circumstances and further extenuating circumstances necessitate additional accommodations.

We'll do a PollEV dry run (doesn't count) in lecture Wednesday, and lecture points will start with lecture on **Fri 4/5** (first real poll Fri., first quiz released after lecture Fri.).

Grading

******	52%	Assignments
*	8%	Lab Participation
*	5%	Lecture Points
**	15%	Midterm Exam
**	20%	Final Exam

Read our full course policies document:
https://cs107.stanford.edu/syllabus.html

Exams

- **Midterm exam** – Tuesday, May 7, 7-9PM outside of class
  - Contact the Head TA by 11:59PM on Friday, April 19 if you have an academic or University conflict with this time, and absolutely cannot make the regularly scheduled midterm. Please include all times 5/6 through 5/8 when you can take the exam.
- **Final exam** – Monday, June 10, 3:30PM-6:30PM
  - No alternate final! You **MUST** be able to take the final exam at the scheduled time (except for university athletics or OAE accommodations)

- Both exams are in-person closed-book paper exams, with an allowed notes sheet. You are also provided with a reference sheet.
- SCPD students have 36hr window during which to take the exams

Grading

******	52%	Assignments
*	8%	Lab Participation
*	5%	Lecture Points
**	15%	Midterm Exam
**	20%	Final Exam

Read our full course policies document:
https://cs107.stanford.edu/syllabus.html

Question Break!

What questions do you have about section, lecture or exams?

Getting Help

- Post on the **Discussion Forum**
  - Online discussion forum for students; post questions, answer other students' questions
  - Best for course material discussions, course policy questions, short debugging questions or general assignment questions (DON'T POST ASSIGNMENT CODE!)
- Visit **Helper Hours**
  - Chat about course topics or just hang out
  - Sign up in a queue for 1:1 TA help; schedule will be posted / hours start tomorrow.
  - Mix of in-person-only and online-only helper hours
  - Best for **group work**, **coding/debugging questions (with TAs only!)** or **longer course material discussions**

Course Staff Contact Information

- Email the **Head TA** for requests of a personal nature, such as about: assignment autograder test scores, Office of Accessible Education accommodations, extension requests or other accommodations, enrollment questions, auditing, or other personal matters.
- Email the **instructor** for questions about private/personal matters.
- Email your **lab TA** for questions about section attendance grades, or for section accommodations (e.g. missing a section due to extenuating circumstances).
- Email the **grader** listed at the top of your assignment grade report if you have questions about assignment style or manual review grades - for questions about assignment autograder test scores, please email the **Head TA**.
- We are not able to answer course material or assignment questions via email; instead, take advantage of Ed or Helper hours resources!

OAE Accommodations

We are eager to do everything we can to support you and make you successful in CS107! Please email the Head TA as soon as possible with any accommodations you may need for the course. In particular, please let us know of any needed exam accommodations by **Fri 4/19 if possible.**

Course Flexibility

If you are ever sick or encounter an emergency or other exceptional circumstance, we have a variety of accommodation mechanisms, including:

- Assignment late days
- Makeup labs or excused absences
- Lecture excused misses
- Exam accommodations for emergencies/illness
- Ability to attend all helper hours remotely with Head TA permission

If you feel ill or are sick, **please stay home and take care of yourself**. We never want you to feel that you must attend class or helper hours if you are not feeling well. And if you are ill or have another emergency or exceptional circumstance, please reach out to us so that we can help!

Stanford Honor Code

From http://honorcode.stanford.edu (newly updated Honor Code):

The Honor Code is an undertaking of the Stanford academic community, individually and collectively. Its purpose is to uphold a culture of academic honesty.

Students will support this culture of academic honesty by neither giving nor accepting unpermitted academic aid in any work that serves as a component of grading or evaluation, including assignments, examinations, and research.

Instructors will support this culture of academic honesty by providing clear guidance, both in their course syllabi and in response to student questions, on what constitutes permitted and unpermitted aid. Instructors will also not take unusual or unreasonable precautions to prevent academic dishonesty.

Students and instructors will also cultivate an environment conducive to academic integrity. While instructors alone set academic requirements, the Honor Code is a community undertaking that requires students and instructors to work together to ensure conditions that support academic integrity.

Honor Code and CS107

It is your responsibility to ensure you have read and are familiar with the honor code guidelines posted on the main page of the course website. Please read them and come talk to us if you have any questions or concerns.

https://cs107.stanford.edu/collaboration

Please help us ensure academic integrity:

- Indicate any assistance received on HW (books, friends, etc.).
- Do not look at other people's solution code or answers
- Do not give your solutions to others or post them publicly on the web or our Ed forum.
- Tutoring is not appropriate for help with work that will be submitted for a grade.
- Do not use AI tools to write code/responses for you on assignments or any graded work.

Honor Code and CS107

https://cs107.stanford.edu/collaboration

- Assignments are checked for similarity with help of robust software tools and processes. Concerns are reported to the Office of Community Standards.
- Any cases determined by the OCS process to be Honor Code violations will result in zero credit for the work of concern plus a course grade penalty of at least a one grade bucket decrease (e.g. B to B-) up to failing the course.
- If you need help, please contact us and we will help you.
  - We do not want you to feel any pressure to violate the Honor Code in order to succeed in this course.
  - We also have a retraction policy that permits retracting all or part of previously-submitted assignment work at any time, no questions asked, up to the start of the final exam.

Use of AI Tools

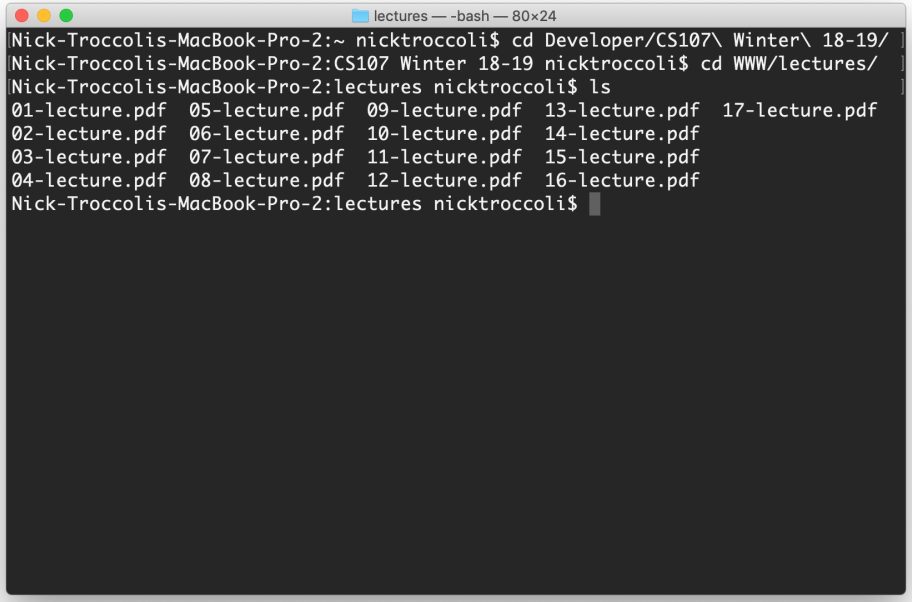
- AI tools can be extremely valuable in certain contexts and enable easier development / coding.
- However, for CS107 our focus is on not just the artifact but also the process: developing skills to write code, debug code, and think critically about existing code and code you write, all of which will make you a more powerful computer scientist and let you work more effectively and efficiently!
- For these reasons, you should only use AI tools in the same way that you would ask a friend in the class for help – high level questions, citations where needed, etc. You **should not use AI tools to write code/responses for you on assignments or any graded work**. Doing so is a violation of the Stanford Honor Code.

Plan For Today

- Introduction
- CS107 Course Topics
- CS107 Course Policies
- **Unix and the Command Line**

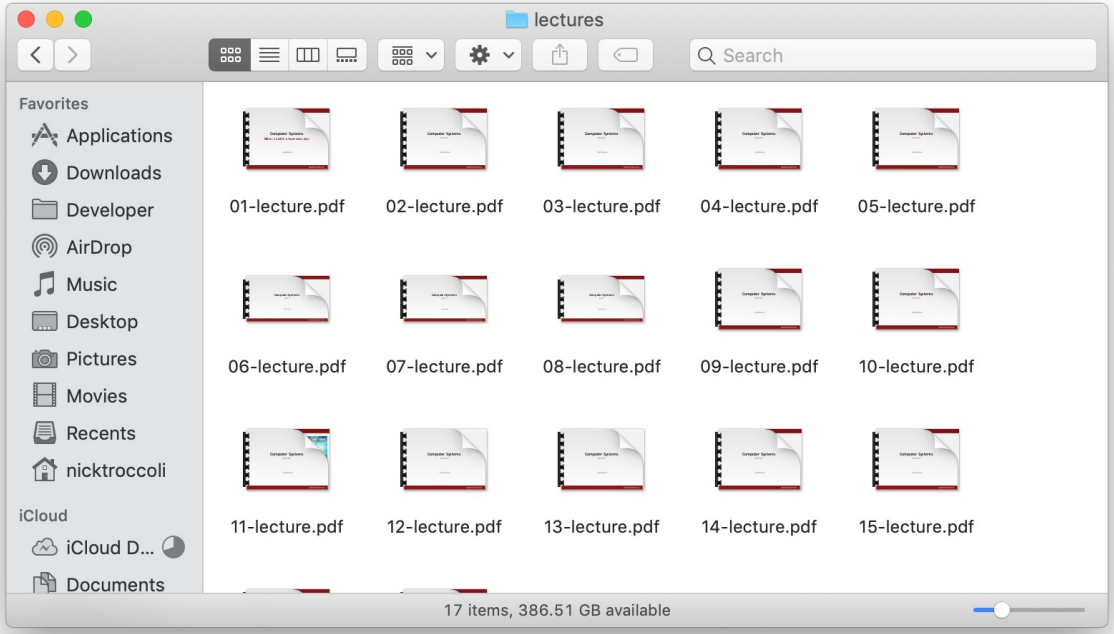
What is Unix?

- **Unix:** a set of standards and tools commonly used in software development.
  - **macOS** and **Linux** are operating systems built on top of Unix
- You can navigate a Unix system using the **command line** (“terminal”)
- Every Unix system works with the same tools and commands

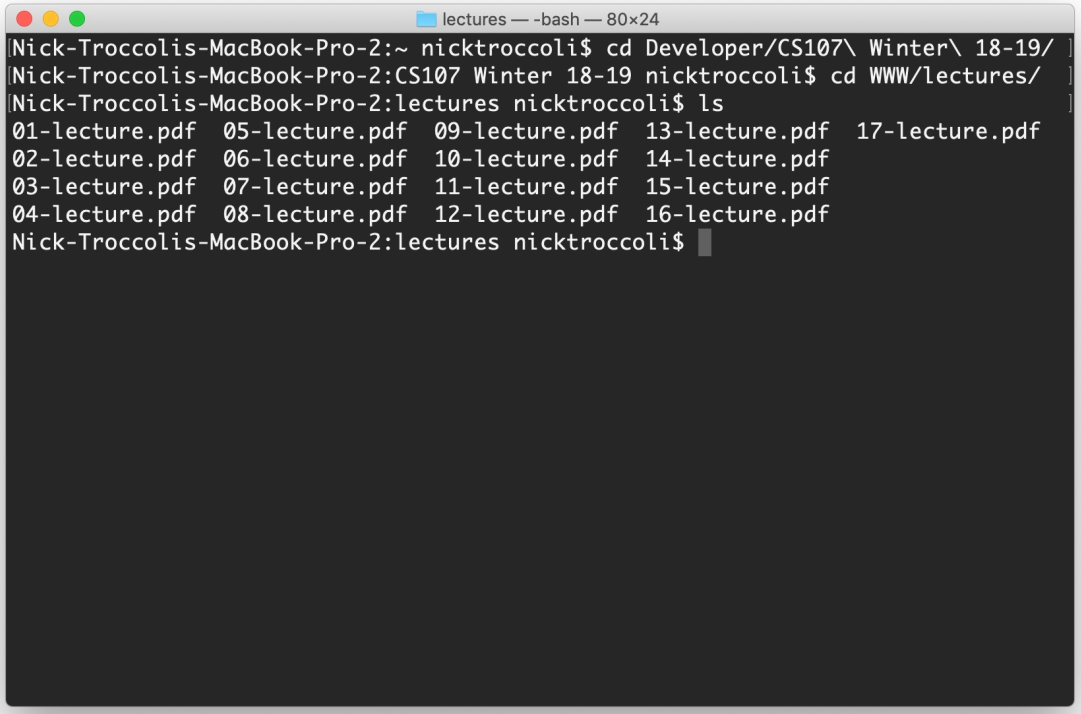


What is the Command Line?

- The **command-line** is a text-based interface (i.e., **terminal** interface) to navigate a computer, instead of a Graphical User Interface (GUI).



Graphical User Interface

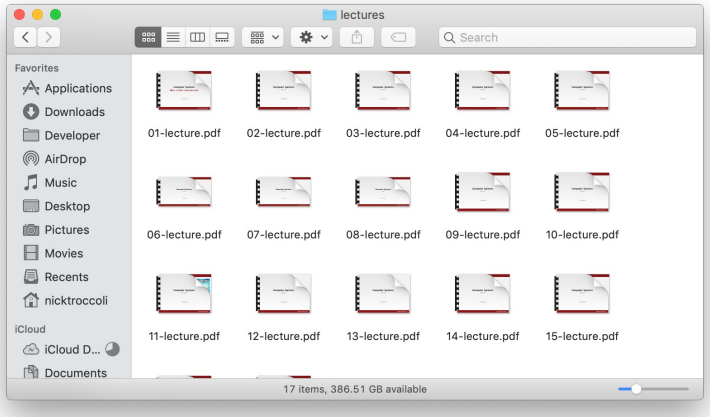


Text-based interface

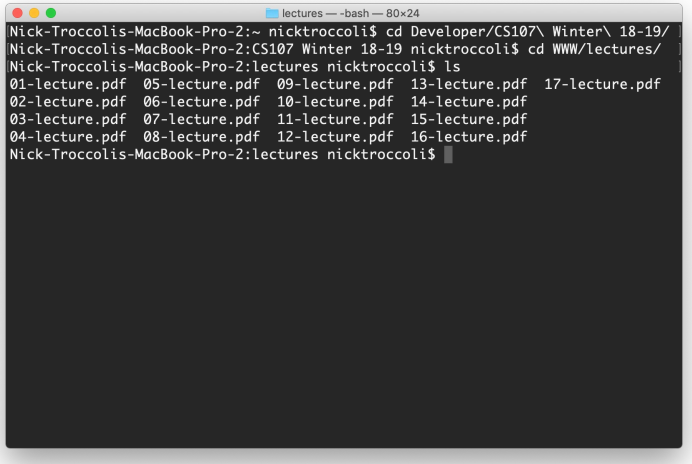
Command Line Vs. GUI

Just like a GUI file explorer interface, a terminal interface:

- shows you a **specific place** on your computer at any given time.
- lets you go **into folders** and **out of folders**.
- lets you **create new** files and **edit** files.
- lets you **execute programs**.



Graphical User Interface



Command-line interface

Why Use Unix / the Command Line?

- You can navigate almost any device using the same tools and commands:
  - Servers
  - Laptops and desktops
  - Embedded devices (Raspberry Pi, etc.)
  - Mobile Devices (Android, etc.)
- Used frequently by software engineers:
  - **Web development**: running servers and web tools on servers
  - **Machine learning**: processing data on servers, running algorithms
  - **Systems**: writing operating systems, networking code and embedded software
  - **Mobile Development**: running tools, managing libraries
  - And more…
- We'll use Unix and the command line to implement and execute our programs.

Learning Unix and the Command Line

- Using Unix and the command line can be intimidating at first:
  - It looks retro!
  - How do I know what to type?
- It's like learning a new language:
  - At first, you may have to constantly look things up (**resources** on course website!)
  - It's important to spend as much time as possible (during labs and assignments) building muscle memory with the tools

More next time!

Assign0

Assignment 0 (Intro to Unix and C) is due in one week on **Mon. 4/8 at 11:59PM PDT**. It relies on material from this and the next lecture.

There are **5** parts to the assignment, which is meant to get you comfortable using the command line, and editing/compiling/running C programs:

- Visit the website resources to become familiar with different Unix commands
- **Clone** the assign0 starter project
- **Answer** several questions in `readme.txt`
- **Compile** a provided C program and **modify** it
- **Submit** the assignment

Recap

- CS107 is a programming class in C that teaches you about what goes on under the hood of programming languages and software.
- We'll use Unix and command line tools to write, debug and run our programs.
- Please visit the course website, cs107.stanford.edu, where you can read the General Information page, information about the Honor Code in CS107, and more about CS107 course policies and logistics.

We're looking forward to an awesome quarter!