CS107, Lecture 1
Welcome to CS107!

reading:

Course Syllabus

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

Honor Code and Collaboration Page

This document is copyright (C) Stanford Computer Science and Nick Troccoli, licensed under Creative Commons Attribution 2.5 License. All rights reserved.
Based on slides created by Cynthia Lee, Chris Gregg, Jerry Cain, Lisa Yan and others.
NOTICE RE UPLOADING TO WEBSITES: This content is protected and may not be shared, uploaded, or distributed. (without expressed written permission)

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/97282/discussion

Today's thread:
https://edstem.org/us/courses/97282/discussion/7867674

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

Julián Rodríguez Cárdenas
(Head TA)

Ben Yan
(Assignment Grading Lead)

Peter Benitez

Diego Padilla

June Lee

L'hussen Toure

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 a 1-unit CR/NC supplementary companion course to CS107. It's designed to provide extra support and problem-solving resources for students, particularly for those who identify as coming from underrepresented or under-resourced backgrounds.


Isabel Berny

- Entry by application; section M/W 1:30-2:20PM in Lathrop 014, beginning week 2.
- In addition to all normal CS107 requirements such as lab
- see https://web.stanford.edu/class/cs107ace/ for more details and the link to apply
- Applications are still open! Isabel will go through applications and send out enrollment codes to accepted students by the end of the week. The final deadline for applications is Friday 4/3 at 5pm! (for late enrollment, email Isabel)

Course Website

cs107.stanford.edu

Ed Discussion (includes course announcements):
https://edstem.org/us/courses/97282/discussion

*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

Plan For Today

- Introduction
- CS107 Course Topics
- **CS107 Course Policies**
  - **Course website**
  - **Textbook**
  - **Grading**
  - **Getting Help**
  - **Honor Code**
- Unix and the Command Line

Course Syllabus and Schedule

cs107.stanford.edu/syllabus

cs107.stanford.edu/calendar

CGOE Students

cs107.stanford.edu/cgoe

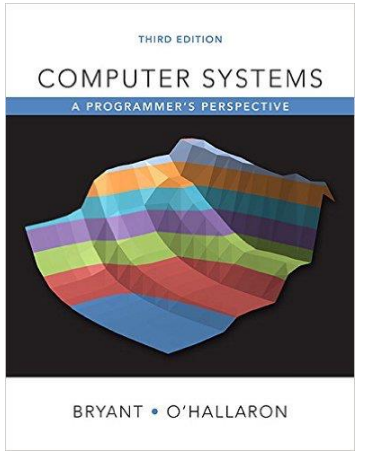
Getting Started Guide

cs107.stanford.edu/getting-started.html

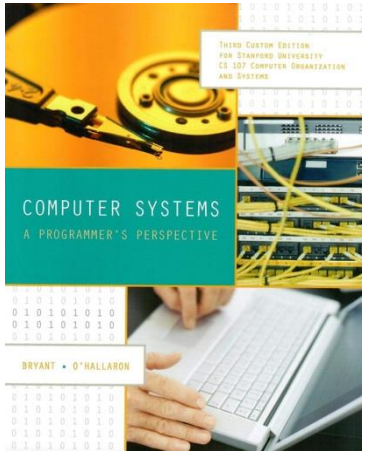
(on course website under ”handouts”)

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

****	35%	Assignments
*	5%	Lab Participation
*	5%	Lecture Points
**	20%	Midterm Exam
****	35%	Final Exam

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

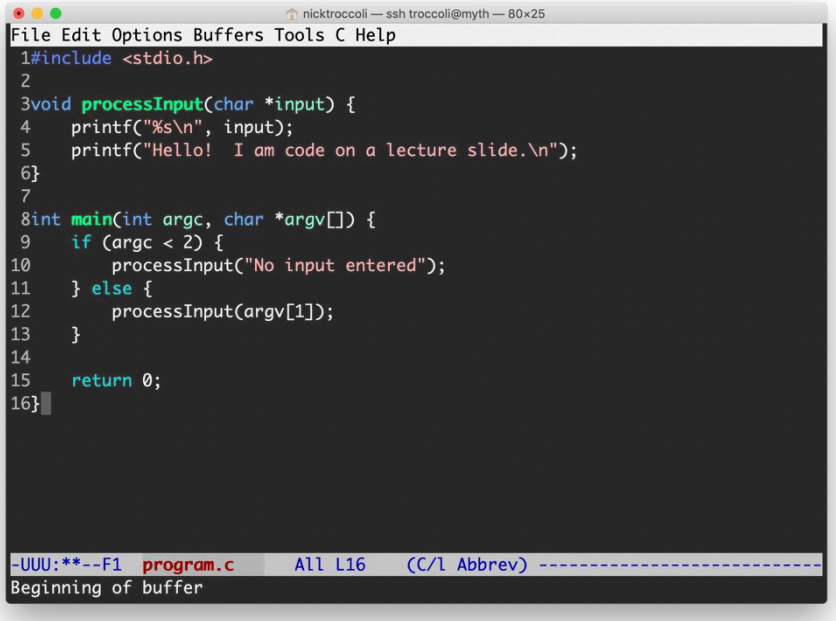
Grading

****	35%	Assignments
*	5%	Lab Participation
*	5%	Lecture Points
**	20%	Midterm Exam
****	35%	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). Style weighted 15%, functionality 85%.
  - 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
- Weights of each assignment in course syllabus



The Style Bucket System

great	An outstanding job; reflects code that is notably clean, elegant and readable, with no issues present.
minor-issues	A good job; reflects code that demonstrates solid effort and is fairly successful at meeting expectations, but also has opportunities for improvement.
major-issue	Has more problems, but shows some effort and understanding. There was either a large concern, or several smaller concerns, in the submission.
multiple-major-issues	Has significant issues, either several large issues or a multitude of smaller ones, that together constitute very poor style work.
0	No work submitted, or barely any changes from the starter assignment.

Assignments – Emphasizing the Process

- When working on the assignments, it's not just about completing the assignments, but *how* you complete the assignments.
- "Process, instead of artifact"
- Focus on *how* you approach the assignments to ensure you achieve the learning goals
  - e.g. when you encounter a bug, make sure you exactly understand the root cause and fix before continuing.
  - E.g. make sure you understand every code change you make and why it's necessary

Assignment Late Policy

Assignments are due at 11:59PM PDT, but also allow for late submissions up to 2 days after the published deadline, with a cap:

- Submitting by the published deadline does not incur any cap
- Submitting after the deadline but within 24 hours means you can get at most 95% of the points.
- Submitting between 24 hours and 48 hours after the deadline means you can get at most 87.5% of the points.

Submissions closed beyond 2 days, and this policy does not apply for the last assignment (assign6), where no late submissions are accepted except for Head-TA granted extensions. Assign5 permits submissions at most 1 day late except for Head-TA granted extensions. (extension form on course homepage)

Assignment Late Policy

Only the Head TA can grant extensions of the on-time deadline (via form submission on the course homepage) due to medical/emergency/extenuating circumstances, or extensions beyond the specified late period. All extension requests must be received in advance of the assignment deadline, or as soon as possible if extenuating circumstances occur later, or extenuating circumstances prevent reaching out prior to the deadline. Please do not hesitate to reach out to the course staff or the instructor if any personal circumstances or issues arise!

Question Break!

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

Grading

****	35%	Assignments
*	5%	Lab Participation
*	5%	Lecture Points
**	20%	Midterm Exam
****	35%	Final Exam

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

Lab Sections

- Weekly 1-hour 20-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 3/31 at 9AM PST** and **are not first-come first-serve**. You may submit your preferences anytime until **Thursday 4/2 11:59PM PST**. Submit your preferences on the course website.

Grading

****	35%	Assignments
*	5%	Lab Participation
*	5%	Lecture Points
**	20%	Midterm Exam
****	35%	Final Exam

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

Lecture Recordings

- Because CS107 is offered on CGOE (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 goal is to incentivize staying current with lectures to enable you to **start early on assignments** and **get the most out of labs**.

- In each lecture (starting Fri), we'll use Poll Everywhere and in-class worksheets to take polls and do practice questions. If you submit answers to all poll/worksheet questions in a lecture (regardless of correctness), you get credit for that lecture.
- We will provide **3 pre-excused absences** for when you are ill, in COVID-19 isolation, or have other extenuating circumstances. They are intended only for these scenarios!
- Further excused absences are granted by the **Head TA** only in cases where you have already used your excused absences for extenuating circumstances and further extenuating circumstances necessitate additional accommodations.

Lecture Points

Polls require being present in the classroom – CGOE students can complete per-lecture quizzes by the end of the week each week.

We'll do a PollEV dry run (doesn't count) in lecture Wednesday, and lecture points will start with lecture on **Fri 4/3.**

Grading

****|	35%	Assignments
*	5%	Lab Participation
*	5%	Lecture Points
**	20%	Midterm Exam
****|	35%	Final Exam

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

Exams

- **Midterm exam** – Tuesday, May 5, 7-9PM outside of class
  - Submit our exam information form by 11:59PM on Friday, April 17 if you have an academic or University conflict with this time, and absolutely cannot make the regularly scheduled midterm. We will offer 2 alternate midterm exam times.
- **Final exam** – Wednesday, June 10, 8:30AM-11:30AM
  - No alternate final! You **MUST** be able to take the final exam at the scheduled time (except for university athletics or OAE accommodations). If you have any concerns, please reach out to me (Nick).
- Both exams are in-person, closed-book, closed-note, closed-computer paper-and-pencil exams. You are provided with a reference sheet during the exam.
- CGOE students have a window during which to take the exams
- Submit our exam information form if you have a midterm conflict, are a CGOE student wanting to take exams on campus, or if you want a left-handed desk!

AIWG Proctoring Study

This quarter, CS107 is participating in the Proctoring Study, meaning the midterm and final exams will be proctored, and some OAE exams will be administered by the Centralized Testing Center (CTC).

For this reason, OAE exam accommodations must be submitted to us at least 10 calendar days prior to the midterm, and no later than May 25th, 2026, at 5:00 pm for the final exam. For urgent OAE-related accommodation needs that arise after the deadline, please consult your OAE adviser.

Grading

****	35%	Assignments
*	5%	Lab Participation
*	5%	Lecture Points
**	20%	Midterm Exam
****	35%	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

- 3 forms on the course homepage for OAE, extensions, and exam info.
- Email the **grader** on your assignment grade report for questions about assignment style or manual review grades - for questions about assignment autograder test scores, please email the **assignment grading lead TA**.
- 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 **Head TA** for other requests, such as about Office of Accessible Education accommodations or other accommodations, enrollment questions, auditing, or other personal matters.
- Can also email the **instructor** for questions about private/personal matters.
- 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 upload your OAE letter via our OAE form on the course homepage as soon as possible. In particular, please let us know of any needed exam accommodations by **Fri 4/17 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:

- Extended time on assignments
- Makeup labs
- Lecture pre-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

The resources that we designate as off limits are designated as such because:

- We want the work you submit to be your own
- This is the best way we believe we can achieve the learning goals for CS107
- The learning happens in the process; appropriating someone else's work as your own, or using help beyond what is allowed, means you deprive yourself of that process.
- 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 or other work up to 5 days after the on-time deadline, no questions asked (except for the final exam).

Honor Code and CS107

We have sophisticated tools and detection mechanisms. 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.

We tell you this to show you how seriously we take this – we want to foster an environment where everyone can do their best work with integrity.

Use of AI Tools

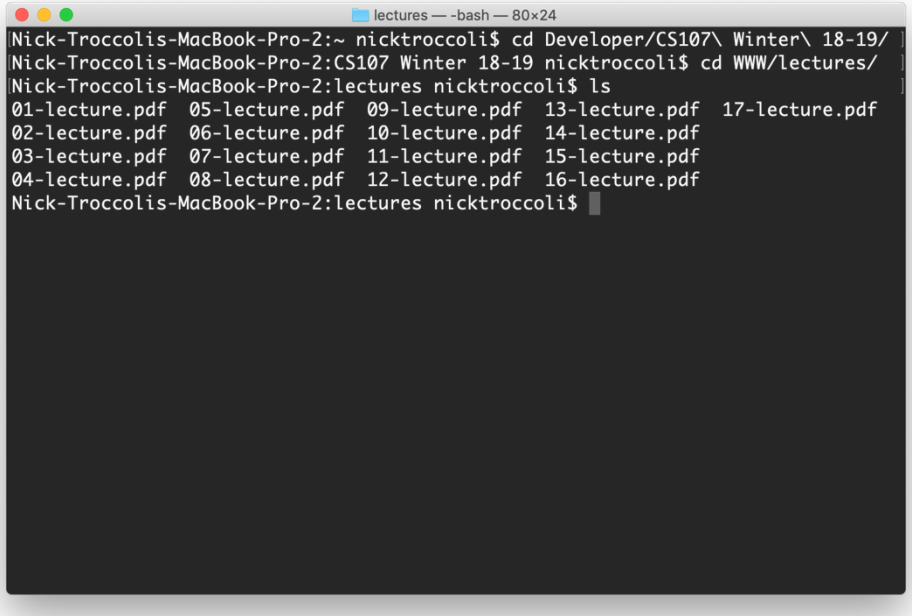
- AI tools can be extremely valuable in certain contexts and enable easier development / coding.
- However, for CS107 our focus is on not just the artifact but also the process: developing skills to write code, debug code, and think critically about existing code and code you write, all of which will 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**

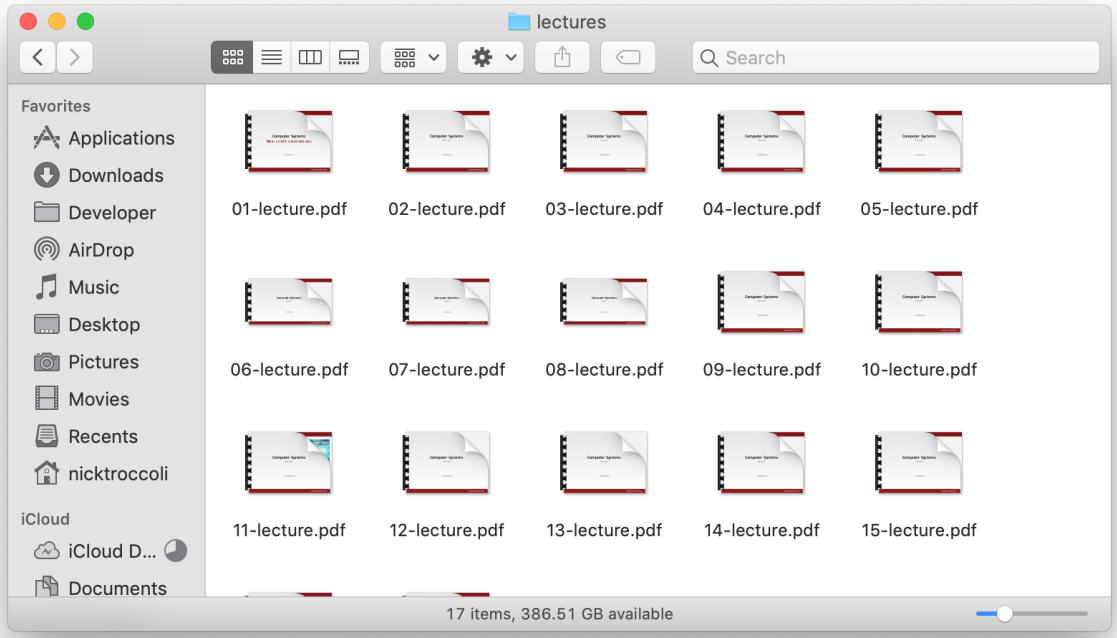
What is Unix?

- **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

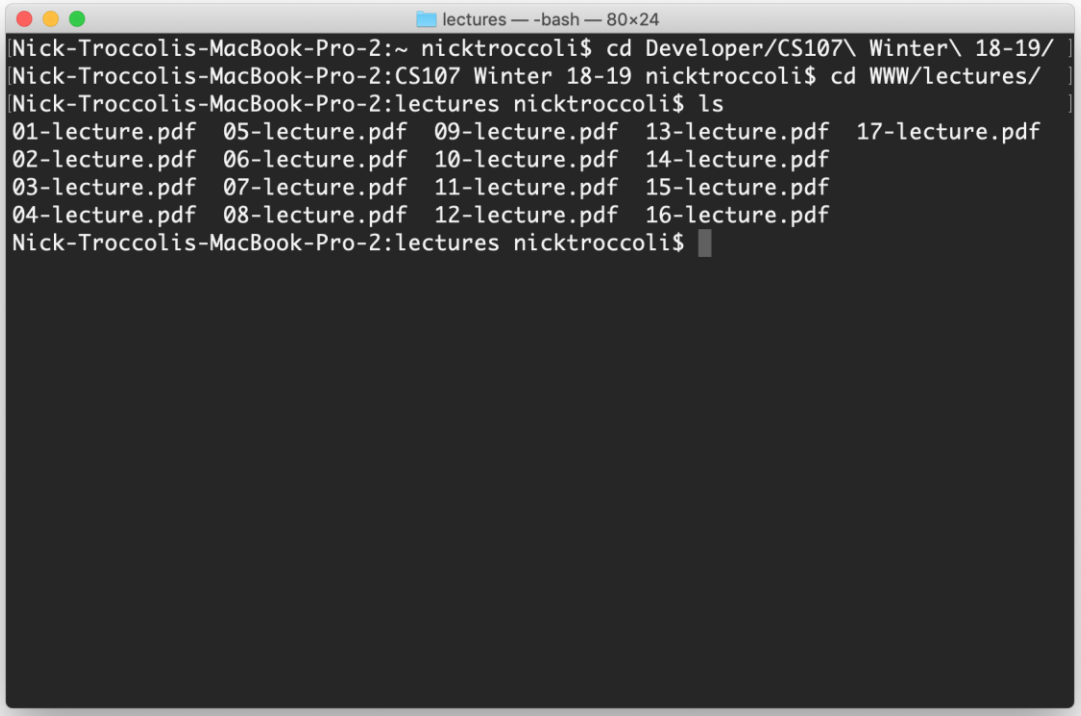


What is the Command Line?

- 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

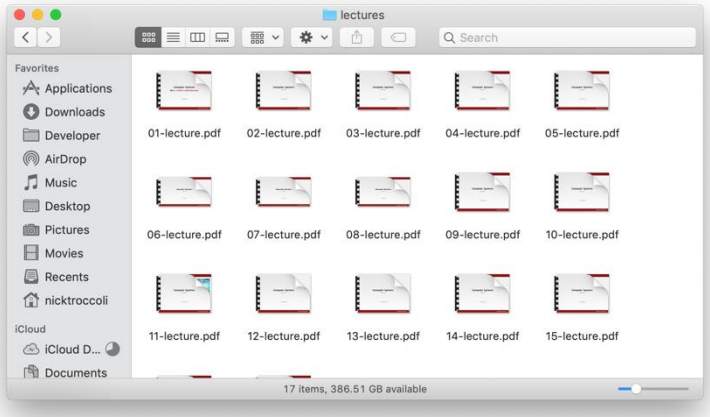


Text-based interface

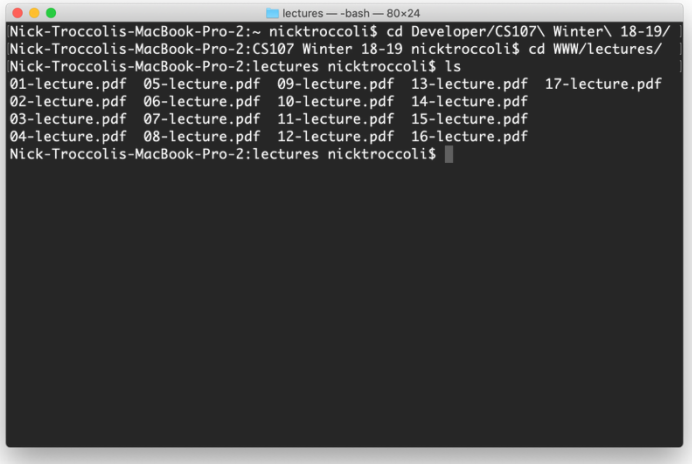
Command Line Vs. GUI

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. 4/6 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!