Welcome to CS170!

TA's: Benjamin Carter & Joseph Ng
ULA's: Sammy Lesner & William Chung

https://sites.cs.ucsb.edu/~rich/class/cs170/

About the TA’s and ULA’s

Your TA’s:

Ben Carter

- 1st year PhD student in RACELab (Professor Rich’s lab, distributed systems / IoT)
- Office Hours:
  - 3pm to 4:30pm Monday’s @ CSIL
  - 9:30am to 11:00am Tuesday’s @ CSIL

Joseph Ng

- CS Masters student (SecLab)
- Office Hours:
  - 2pm to 3:30pm Thursday’s @ CSIL
  - 2pm to 3:30pm Friday’s @ CSIL

Your ULA’s:

Sammy Lesner

- Office Hours:
  - 4pm to 6pm Monday’s @ CSIL

William Chung

- Office Hours:
  - 3:30pm to 5:30pm Wednesday’s @ CSIL

Helpful Links

Class Page: https://sites.cs.ucsb.edu/~rich/class/cs170/

Section Home / Lab Rubrics:
http://sites.cs.ucsb.edu/~rich/class/cs170/discussion.html

Section Slides:
http://sites.cs.ucsb.edu/~rich/class/cs170/discussion/section_slides.html

Canvas:

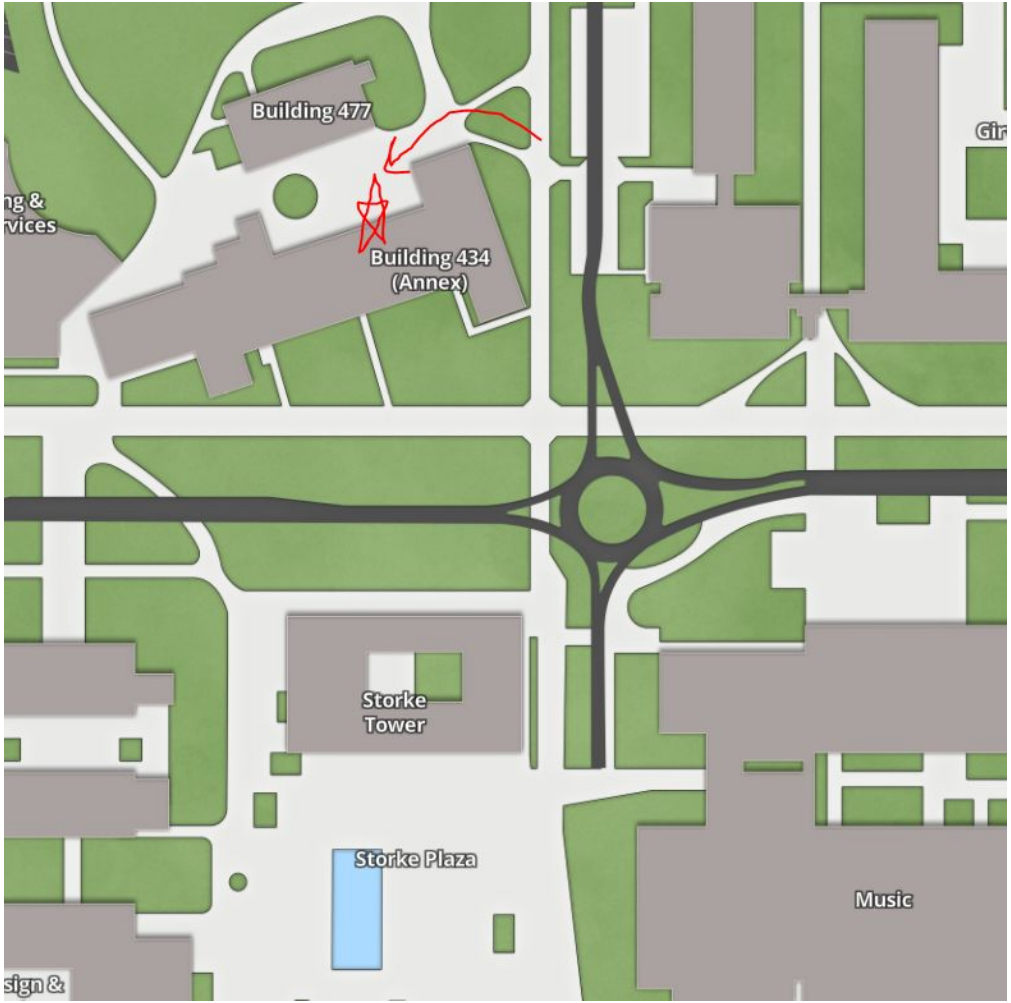
Lab info: https://sites.cs.ucsb.edu/~rich/class/cs170/labs.html

Lecture info:
https://sites.cs.ucsb.edu/~rich/class/cs170/lecture_notes.html

Where is CSIL??

It has moved!
It is behind building 434

You can also SSH into CSIL at
`csil.cs.ucsb.edu`



CSIL Computers

You will need CSIL to test everything.

Create account: https://accounts.engr.ucsb.edu/create

Tutorial on setting up SSH: https://ucsb-compsci-grads.github.io/CSIL/

On your device: log in via SSH

- user: <username>@csil.cs.ucsb.edu
- password of CoE account

User and password combination from CoE account creation.

VSCode NOT recommended on CSIL.

Recommended to use **ssh/sftp/scp/rsync** for remote shell and file transfer.

Recommended to use **vim** (and additionally **tmux**) for remote text editing.

Class Activity

- What is your name?
- Animal below that best describes you.

A. Parakeet

B. Penguin

C. Peregrine Falcon

D. Pelican

E. Peacock

Class Activity - Group Breakout

Once you have created a group, fill out this form:

https://forms.gle/KDqxXD4q83ZoYh5L9

Tips for class

- Go to lecture
- Come to section
    - We will be reviewing info for labs as well as review for final later in the quarter.
- Don’t procrastinate
- Come by office hours!
    - Every week there are 12 hours of office hours! (Not including section or lecture) There are times M-F
- Please please please *don’t copy/paste ChatGPT!!* (It’s obvious and it only harms you, not to mention ChatGPT’s dislike of C)
- Read, read, and read again.

Tools and Infrastructure

Debugging

GDB - GNU Debugger

In makefile, add the flags to gcc: `-g -O0`

Then, load executable via gdb command.

```
gdb ./program

(gdb) <GDB setup commands>
(gdb) run   starts the gdb run section
(gdb) <GDB control-mode commands>
(gdb) q    quit
```

GDB overview:

Setup:

- *b* Create breakpoints
- *run* Start program

Control-mode: (program currently running)

- *c* Continue
- *s* Step
- *n* Next
- *p* Print

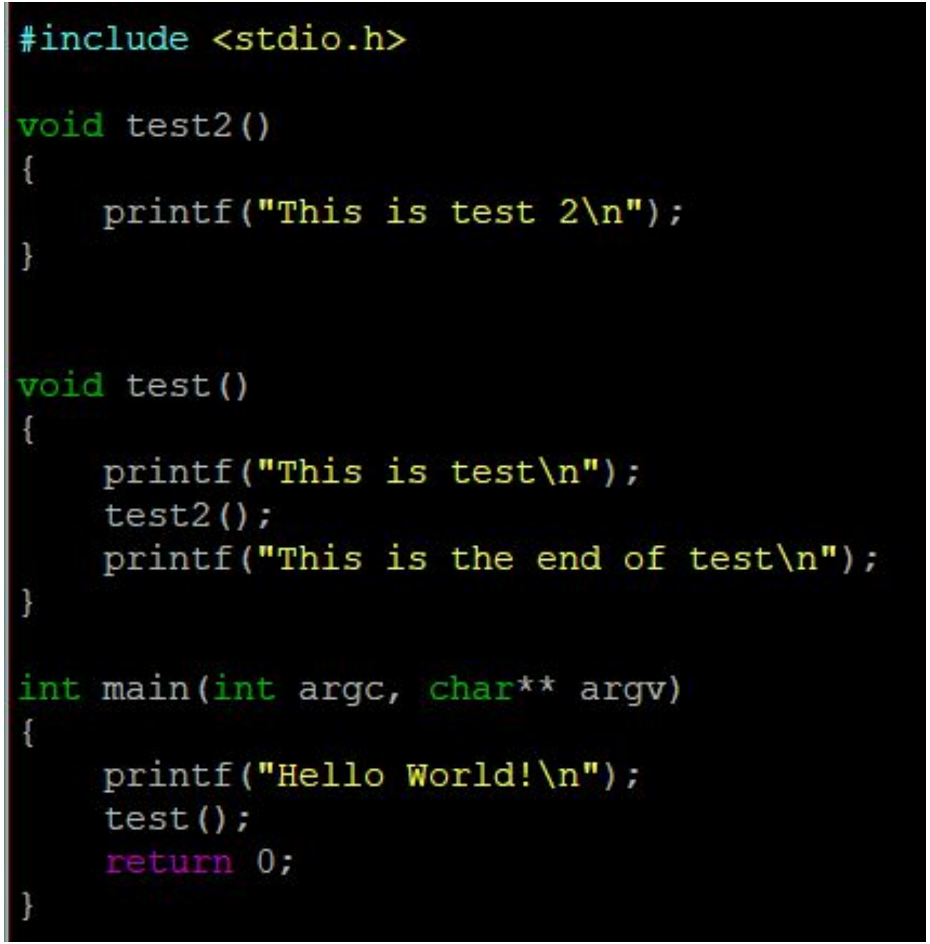
Debugging

Control-mode: (program currently running)

- *c* Continue
  continue to next breakpoint
- *s* Step
  “down a level”... Go inside a func
- *n* Next
  Next command inside the current scope
- *p* Print value
- *list* View code around the current line

Tips:

- Breakpoints at beginning of functions
- Don’t forget to *make* first!
- Don’t forget adding the flags `-g -O0`



```c
#include <stdio.h>

void test2()
{
    printf("This is test 2\n");
}


void test()
{
    printf("This is test\n");
    test2();
    printf("This is the end of test\n");
}

int main(int argc, char** argv)
{
    printf("Hello World!\n");
    test();
    return 0;
}
```

Posted on section slides page ^^

Lab0: MyMalloc Overview

Lab0: MyMalloc

Link: https://sites.cs.ucsb.edu/~rich/class/cs170/labs/lab1.malloc/index.html

Due: Tuesday, Jan 14, 2025 at 11:59 p.m.

IMPORTANT NOTE ABOUT AUTOGRADER/GRADESCOPE:

- Your submission might not exactly match with Gradescope feedback. That is OK! Your Malloc might use a different allocating strategy and later labs (KOS) has asynchronous operations.