CS 326: Operating Systems

Exploring the OS

Lecture 2

Today's Schedule

- Development Environment Setup
- Exploring xv6

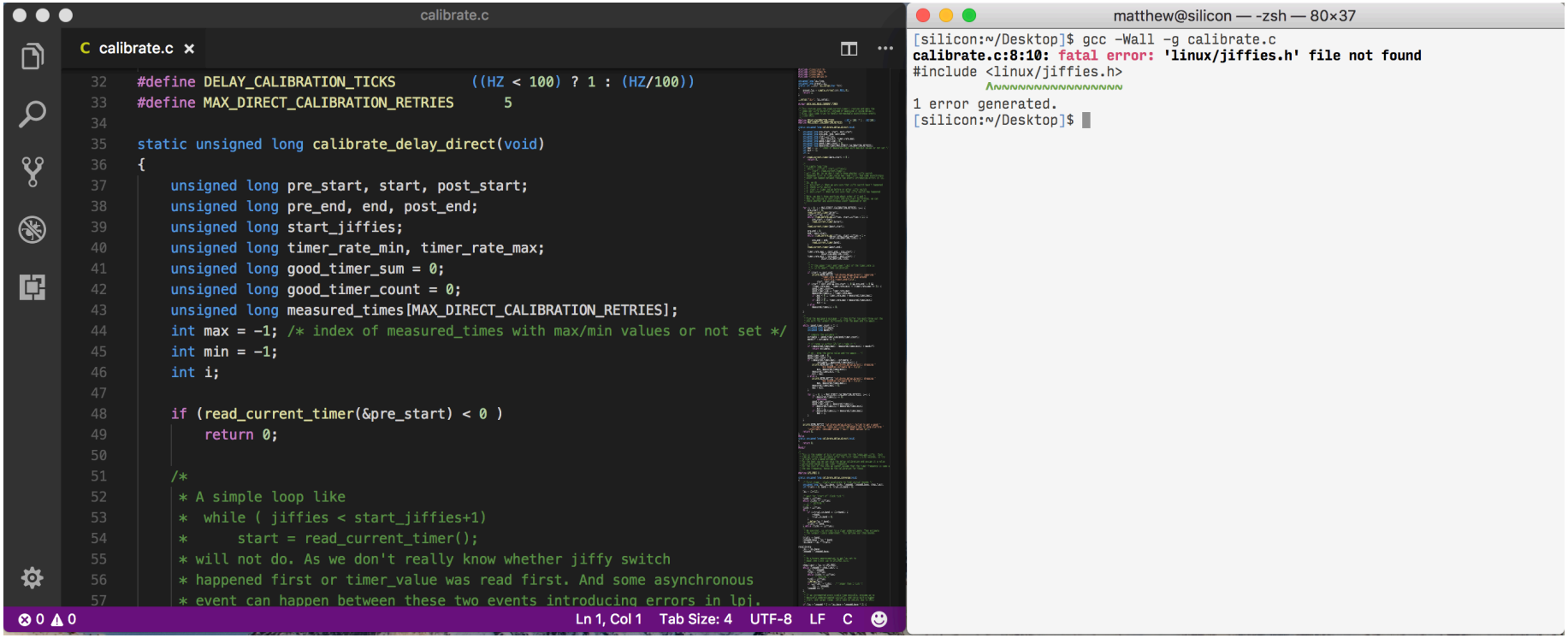
Today's Schedule

- **Development Environment Setup**
- Exploring xv6

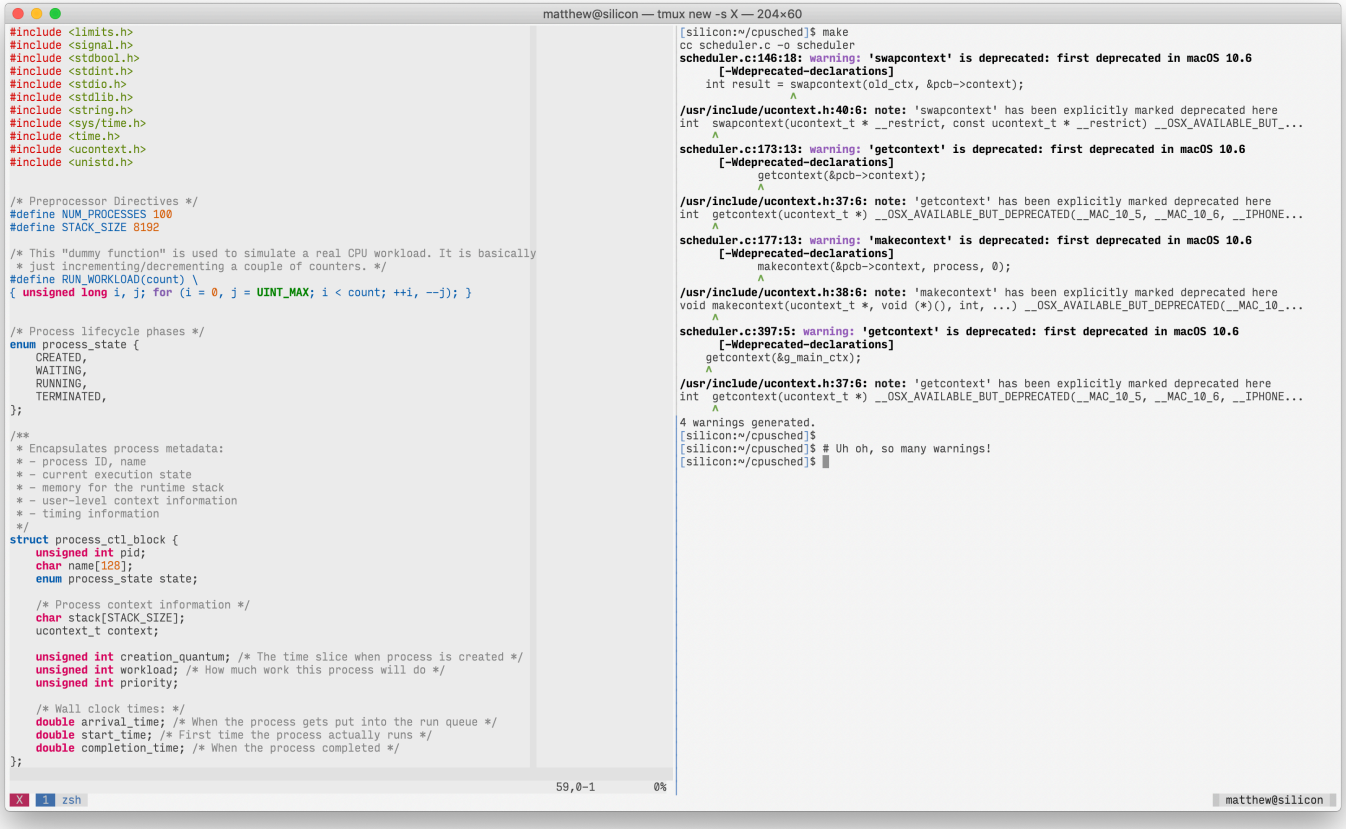
Writing C Programs [1/3]

- Using an IDE (like Eclipse, IntelliJ, etc) is less common in the C world
- Many C developers prefer to use a text editor and a terminal to write their programs
  - Text editor: edit, save
  - Terminal: compile, run
- There's information on the course schedule page for setting up your editor and C compiler

Writing C Programs [2/3]



Writing C Programs [3/3]



My Recommendation [1/2]

- In 326, you are going to be using the terminal and command line interfaces **a lot**
- I recommend you to embrace it, learn it, and (maybe) love it
  - Sets you up for working in cloud computing, DevOps, system administration
  - The interface is all text: facilitates command **composition**
- If you're not super comfortable with Unix commands, don't worry! You'll get lots of practice

My Recommendation [2/2]

- Learn vim
  - …or emacs, nano, micro, etc.
- The point: know enough about using a terminal text editor to be able to get things done
- Maybe you won't spend 100% of your time there, but it can come in handy in a pinch

Other Options

- Lots of IDEs have remote editing functionality
  - *rsub* (Remote Sublime) is a popular option
  - Visual Studio Code has very powerful remote editing, syntax highlighting, autocompletion
- FTP/SFTP clients like **Cyberduck**, **Termius**, **Forklift** can automatically sync your changes with a remote server
- Ultimately, use what you're comfortable with
  - Spending 60 hours learning vim is awesome, but not if it means you can't get your projects done

Getting Help

- When you're working from the terminal, the `man` (manual) pages are a great resource for help
- Many times your Google searches are just going to locate man pages that have been converted to HTML!
- There are a few sections in the man pages:
  1. User commands
  2. System calls
  3. C library functions
  4. …and more

Reading man pages

- `man whatever`
  - `man man`
- Specify the section like so:
  - `man 3 printf`
  - This is particularly important for our class: we need section **2** for system calls and section **3** for C functions
- Man pages also will often explain config files' syntax and options

Creating an ssh alias

- Please take a look at the Working Remotely page on the course schedule for more hints!
- The best bit of advice: creating an `ssh` alias for **gojira**
  - I also highly recommend setting up an ssh key so you won't need to type your password over and over

Today's Schedule

- Development Environment Setup
- **Exploring xv6**

Exploring xv6

- By building your OS off xv6, you benefit from a lot of existing code
  - …but now you might have to read and understand that code
  - reading code is a good skill to learn, but it's not always fun
- Let's take a tour of xv6 in just a minute…
  - To start, how big is the codebase? `sloccount` can give us an idea…

Make

- You'll use `make` to build your code in this class. That means we need a *Makefile*
  - This tells the `make` utility what to do
  - Essentially just a recipe for building your program
- Hints:
  - `make` – compile kernel
  - `make qemu` – compile kernel, user space, create file system, and run the OS in QEMU
  - `make clean` – clean up all build artifacts

Exploring

- Take 5 minutes to look around the xv6 codebase
  - Find something that you think is interesting
- We'll regroup and go through its structure