Introduction

Don Porter

What is an OS?

• All of the stuff between you/your application and the hardware
  – Kernel
  – Device Drivers
  – API libraries
  – UI
• Our focus is mostly on the kernel, with some attention to the others

Why Operating Systems?

• Primary Goal: Demystify how computers work
  – Lots of abstractions and heuristics between your application and the hardware
  – A good computer scientist should understand what happens inside the system when one types a command
• Secondary: Learn how to write robust programs
  – OSes like Linux have many users and work on a wide range of hardware
  – Deal with subtle issues: concurrency, consistency, etc.

Labs: Learn by doing

• This course is coding intensive
  – You should know C, or be prepared to remediate quickly
  – You will learn basic, inline x86 assembly
  – You must learn on your own/with lab partner
• You will write make substantial modifications to xv6, a simple x86 Unix variant
  – Code is written to be easy to understand, but lacks many modern OS features
  – Challenging work, but a very marketable skill

Lab Teams

• Can work alone, but better with help
  – No need to be a hero
• Choose your own partners
  – Piazza a list good for finding them
• Same for entire course
  – Changes only with instructor permission

Paperwork

• I am handing out a survey on your background and mini quiz
• Please complete and return before you leave.
• Academic honesty homework due Thurs 2/11 in class
Challenge Problems

• Each lab may include challenge problems, which you may complete for bonus points (generally 5—10 points out of 100)
  – Unwise to turn in a lab late to do challenge problems
  – Can complete challenge problems at any point in the semester—even on old labs
• Indicate any challenge problems completed in challenge.txt file

Administrative

• Syllabus, schedule, homework, etc. posted on course website
• www.cs.stonybrook.edu/~porter/courses/cse306/s16

Required Readings

• Primarily from the class textbook
• Should be completed before the lecture
• Required reading material may appear on the exams, even if not discussed in lecture
• Several recommended (optional) texts will be posted
  – Several free on SBU safari online site
  – Papers you can print out or read electronically
  – Others on reserve at library

Lectures

• Discuss and supplement reading material
• An important chance to clarify issues
  – Questions are encouraged!
• I expect you to arrive prepared to answer and ask questions about the reading material
• Everything in lectures may appear on the exams, even if not in the book

Recordings

• I usually record lectures for students to review later
  – NB: This is pending help from the tech staff. This room is not equipped with SBcapture.
• Assuming this works out, recordings are best effort
  – Recordings may fail, or get deleted by accident
  – Or be discontinued if too many students stop attending
    • I need your facial expressions and questions to know if lectures make sense
• Do not use this as a substitute for class attendance

Guest Lectures

• Senior graduate students will give some lectures to gain teaching experience
  – Including today!
• Professor Porter will review and critique guest lectures (in person or recorded) with guests

• Please:
  – Ask questions if something is unclear: in class or on piazza
  – Give Prof. Porter comments on guests (and his lectures)—positive and negative
Prerequisites

• CSE 219 (CS III) or CSE 260 (CS B, Honors)
• CSE 220 (Systems-level Programming) or ESE 380 (Embedded Microprocessor Design I)
• The background courses are necessary
  – I strongly encourage students to take (new) CSE 320 first
• In some cases, industry experience is ok
  – In-class quiz, due before you leave
    • If you can't answer 50% of these questions you are not prepared
• C programming
• Basic Unix command-line proficiency

C Programming

• You should have learned C in the prerequisite courses
• If you have not and want to take the course, you should read “The C Programming Language” by Kernighan and Ritchie cover to cover this week
  – And complete all exercises in the book
• If you can do this, you will be prepared to complete this course on schedule

Course email list

• We will use Piazza this semester. Link on course website
• This is the primary announcement medium
• And for discussions about course work
  – Do not post code here or other solutions
  – Goal: Everyone can learn from general questions
• Material discussed on the mailing list can be an exam question

Other administrative notes

• Read syllabus completely
• Subscribe to the class piazza forum
• 2 exams cover: lectures, labs, mailing list
• Every student will get a VM for lab work
  – You may use your own computer, staff can’t support it
• All staff email goes to cse306ta@cs.stonybrook.edu
  – Except private issues for instructor only

Special Offer!

• You can write your own exam questions
  – Send them to me in advance of the test, if I like them, I will use them
  – Do NOT share with anyone else

Academic Integrity

• I take cheating very seriously. It can end your career.
• In a gray area, it is your job to stay on right side of line
• Never show your code to anyone except your partner and course staff
• Never look at anyone else’s code (incl. other universities)
• Do not discuss code; do not debug each other’s code
• Acknowledge students that give you good ideas
Why do we care?

- Analogy: This is the programming dojo
  - If you don't do your exercises, you will be unprepared for battle
  - You've wasted your money and both of our time
  - It brings dishonor on the dojo when you lose every battle
- Similarly, a lot of what I have to teach (and what will make you a valuable employee when you graduate) has no short cut
  - How do you learn to punch through a board?
  - You punch a board over and over until your fist goes through it

Productive Frustration

- One of the “meta skills” that distinguishes an excellent programmer is the ability to get un-stuck
  - Fixing a “heisenbug” has this property
- How do you learn this skill?
  - Get stuck on a hard, but solvable problem
  - Learn which strategies will get you moving again
- If you take a quick cheat, you won't learn the skills to solve truly hard problems

Integrity Homework

- Exercises applying course policies and ethics to several situations
- Due in class 2/11

Lateness

- Each student gets 72 late hours
  - List how many you use in slack.txt
  - Each day after these are gone costs a full letter grade on the assignment
  - If you work in a team, each member loses 1 hour for each hour late
- It is your responsibility to use these to manage:
  - Holidays, weddings, research deadlines, conference travel, Buffy marathons, release of the next Zelda game, etc.
- 3 Exceptions: illness (need doctor's note), death in immediate family, accommodation for disability

Getting help

- TA’s will keep office hours (TBD)
- Instructor keeps office hours
  - Note that “by appointment” means more time available on demand

Questions?

- Remember:
  - Hand-in survey
  - Do academic honesty homework
  - Lab 1 coming out soon
  - Reading assigned for Thursday