Introduction
CS 111
Operating System Principles
Peter Reiher

Outline

- Administrative materials
- Why study operating systems?

Administrative Issues

- Instructor and TA
- Load and prerequisites
- Web site, syllabus, reading, and lectures
- Quizzes, exams, homework, projects
- Grading
- Academic honesty

Instructor: Peter Reiher

- UCLA Computer Science department faculty member
- Long history of research in operating systems
- Email: reiher@cs.ucla.edu
- Office: 3532F Boelter Hall
  - Office hours: TTh 1-2
  - Often available at other times

TA

- Vahab Pournagshband
  - vahab@cs.ucla.edu
- Lab sessions Fridays from 2-4 PM, in 5420 BH
- Office hours to be announced

Instructor/TA Division of Responsibilities

- Instructor handles all lectures, readings, quizzes, and tests
  - Ask me about issues related to these
- TA handles projects
  - Ask him about issues related to these
- Generally, instructor won’t be involved with project issues
  - So direct those questions to the TA

Web Site

- http://www.lasr.cs.ucla.edu/classes/cs111_summer2013
- What's there:
  - Schedules for reading, lectures, quizzes, exams, projects
  - Copies of lecture slides (Powerpoint)
  - Announcements
  - Sample quiz, exam and final problems

Prerequisite Subject Knowledge

- CS 32 programming
  - Objects, data structures, queues, stacks, tables, trees
- CS 33 systems programming
  - Assembly language, registers, memory
  - Linkage conventions, stack frames, register saving
- CS 118 networking
  - Packets, addressing, routing, protocols,
  - Protocol layering
- I will complement CS 151 coverage of
  - Traps, interrupts, DMA

Course Format

- Two weekly (average 20 page) reading assignments
  - Mostly from the primary text
  - A few supplementary articles available on web
- Two weekly lectures
  - Each preceded by a quiz on the reading
  - First quiz before lecture 2
- Four (10-25 hour) team projects
  - Exploring and exploiting OS features
- One design project (10-25 hours)
  - Working off one of the team projects

Course Load

- Reputation: THE hardest undergrad CS class
  - Fast pace through much non-trivial material
  - Summer schedule only increases the pace
- Expectations you should have

– lectures	4-6 hours/week
– reading	3-6 hours/week
– projects	3-20 hours/week
– exam study	5-15 hours (twice)

- Keeping up (week by week) is critical
  - Catching up is extremely difficult

Primary Text for Course

- Saltzer and Kaashoek: *Principles of Computer Systems Design*
  - Background reading for most lectures
- Supplemented with web-based materials

Course Grading

- Basis for grading:

– 14 daily quizzes	10% (total)
– 1 midterm exam	20%
– Final exam	25%
– Projects	45%

- I do look at distribution for final grades
  - But don't use a formal curve
- All scores available on MyUCLA
  - Please check them for accuracy

Quizzes

- When? Before each lecture, in class
- Scope: Reading assigned for that lecture
- Format:
  - 4 simple questions (definitions, examples, ...)
  - Should require at most one sentence answer
- Closed book
  - You should have read it already
- Goals:
  - To test your familiarity with major concepts
  - To persuade you to do reading **prior** to lecture

Midterm Examination

- When: end of the 4th week (in recitation section)
- Scope: All lectures up to the exam date
  - Approximately 60% lecture, 40% text
- Format:
  - Closed book
  - 10-15 essay questions, most with short answers
- Goals:
  - Test understanding of key concepts
  - Test ability to apply principles to practical problems

Final Exam

- When: Last day of 8th week (recitation section)
- Scope: Entire course
- Format:
  - 6-8 hard multi-part essay questions
  - You get to pick a subset of them to answer
- Goals:
  - Test mastery of key concepts
  - Test ability to apply key concepts to real problems
  - Use key concepts to gain insight into new problems

Lab Projects

- Format:
  - 4 regular projects
  - 2 mini-projects
  - May be done solo or in teams
- Goals:
  - Develop ability to exploit OS features
  - Develop programming/problem solving ability
  - Practice software project skills
- Lab and lecture are fairly distinct
  - Instructor cannot help you with projects
  - TA can’t help with lectures, exams

Design Problems

- Each lab project contains suggestions for extensions
- Each student is assigned one design project from among the labs
  - Individual or two person team
- Requires more creativity than labs
  - Usually requires some coding
- Handled by the TA

Late Assignments & Make-ups

- Quizzes
  - There are no make-ups
  - This would defeat their purpose
- Labs
  - Due dates set by TA
  - TA also sets policy on late assignments
- Exams
  - Only possible with prior consent of the instructor

Academic Honesty

- It is OK to study with friends
  - Discussing problems helps you to understand them
- It is OK to do independent research on a subject
  - There are many excellent treatments out there
- But all work you submit must be your own
  - Do not <u>write</u> your lab answers with a friend
  - Do not <u>copy</u> another student's work
  - Do not turn in solutions <u>from off the web</u>
  - If you do research on a problem, <u>cite your sources</u>
- I decide when two assignments are too similar
  - And I forward them immediately to the Dean
- If you need help, ask the instructor

Academic Honesty – Projects

- Do your own projects
  - Work only with your team-mate
  - If you need additional help, ask the TA
- You must design and write <u>all</u> your own code
  - Other than cooperative work with your team-mate
  - Do not ask others how they solved the problem
  - Do not copy solutions from the web, files or listings
  - Cite any research sources you use
- Protect yourself
  - Do not show other people your solutions
  - Be careful with old listings

Academic Honesty and the Internet

- You might be able to find existing answers to some of the assignments on line
- Remember, if you can find it, so can we
- It IS NOT OK to copy the answers from other people's old assignments
  - People who tried that have been caught and referred to the Office of the Dean of Students
- ANYTHING you get off the Internet must be treated as reference material
  - If you use it, quote it and reference it