Introduction to 15-410/605

Dave Eckhardt
de0u@andrew.cmu.edu

Dave O'Hallaron
droh@andrew.cmu.edu
Wait List

• If you're here but not registered or on a wait list, see me immediately after class

• We admit based on readiness (mixed with need)
  – Usually our estimate centers on your advisor

• The wait list is not a FIFO queue
  – Waiting doesn't actually get you anything
  – Talking to me and talking to your advisor may
Logistical Query

- Who had trouble with 213?
  - Contact me after class (potential for big trouble)
  - *If you didn't get a B or an A, see me*
  - *If the malloc() lab didn't go well, see me*
Self-Assessment

- Self-assessment exercise on course web site?
  - Not mandatory if you did well in 15-213
  - A very good sanity-check, though!
Textbook (traditional)

• Option 1
    • Silberschatz, Galvin, & Gagne
• Multiple “cheap” options exist!
  – eBay/Amazon/Alibris/...
  – If you try an e-book edition instead of paper, please tell us if you like it
  – Used copies of 8th edition work pretty well
    • Web site lists reading assignments for 6th through 10th editions
Textbook (experimental)

• Option 2
  – Operating Systems: Principles & Practice
    • Anderson & Dahlin

• Main differences
  – More focus on typical modern kernels and hardware
  – Less focus on historical systems
  – Stronger coverage of file systems and storage
  – Weaker coverage of security
Textbook (which one?)

• We think you can use either one
  – Heavily-tested material is typically covered in lecture and projects

• We are interested in your opinion!
  – Which one, physical book vs. e-book, e-book purchase vs. rental...
  – We will ask for your thoughts at the end of the semester
Outline

- People
  - Me, us, you
- Administrative information
  - Academic conduct
- Class goals
- Reading material
Dave Eckhardt

- Teaching Professor, CS
  - Ph.D., Computer Science, CMU, 2002
    - “An Internet-style Approach to Managing Wireless Link Errors”
  - http://www.cs.cmu.edu/~davide

- Building Unix kernels since ~1985
  - PDP-11, Version 7 Unix
  - “Not really a BSD bigot”
Dave O'Hallaron

- Professor, CS and ECE
  - Ph.D., Computer Science, Univ. VA, 1986
    - “Models for Concurrent Programming”
    - http://www.cs.cmu.edu/~droh

- Research: High-Performance Computing

- Former director, Intel Labs Pittsburgh

- Co-creator of 15-213

- Co-author (with Randy Bryant) of “Computer Systems: A Programmer's Perspective” (3rd ed.!)
TA's

- Mixture of “repeat offenders” and “this year's model”
- As a team
  - Strong background
  - Here to help!
Yinz - Reading

• Read a Ph.D. thesis?
• Academic journal article?
• Attended an academic conference?
• Read a non-class CS book last semester?
Information Sources

Web site http://www.cs.cmu.edu/~410

- You are *utterly required* to read the syllabus

Q: Can I used a linked list for...?
Q: I have a final exam conflict...
Q: The license server is down...
Q: AFS says “no such device”...

- A: staff-410@cs.cmu.edu
Information Sources

Q: I am experiencing [delicate situation X] ...
A: e-mail to faculty

Note: most likely no Piazza this semester
  - Experiment was run in a previous semester
  - Results equivocal
Course Goals

• Operating Systems
  – What they are
  – Design decisions
  – Actual construction

• Team programming
  – Design, documentation
  – Source control
  – People skills
Course Plan

• Lectures
  – *Many* topics will be covered by text
  – But skipping many lectures *will* challenge your grade
    • The map is not the terrain, the slides are not the lecture
    • You will miss Q&A
  – We expect you to attend lectures
    • Details: see syllabus
Course Plan

• Projects
  – “Stack crawler” - readiness check [1-person project]
  – Bare-machine video game [1-person project]
  – Thread library
  – OS kernel
  – Kernel extension

• Project environment
  – Wind River Simics™ PC simulator
  – Your projects can also run on real PC hardware
Course Plan

• Homework assignments
  – ~2, to deepen understanding of selected topics
• Reading assignment
  – Pick something fun, write a *brief* report
• Mid-term, Final exam
  – Closed-book
Team programming

• Why?
  – Allows attacking larger problems
  – Teaches *job skills* you will need
    • Setting milestones
    • Setting up a productive work flow
    • Involving “management” before it's too late

• Team programming != “software engineering”
  – No requirement analysis
  – No release staging, design for growth, ...
  – Not a complete “life cycle”
Health Problems

- *Somebody* will probably get mono or pneumonia
  - If not, only because of something more creative
- Work-blocking health problem?
  - Go *early* to University Health (etc.)
  - *Avoid* “For the past two weeks I dragged myself to class but couldn't focus on programming”
  - Try to get paper documentation of work restrictions
  - Your program staff will inform instructors
    - CS: Cathy/Mary; ECE: Janet/Vickie/Jillian/Nesli
Partner Problems

- **Somebody** will have serious partner trouble
  - You need to “involve management” early
    - Sometimes (50%) we can fix the problem
    - If the problem can't be fixed, we can reduce the fallout
      - ...only if we know while the trouble is happening
  - Don't “buffer up” partner trouble until the last week of classes
    - At that point, we basically can't help
  - Details: see syllabus
Academic honesty

• See syllabus!
  – Reading the syllabus on this topic is not optional

• Learning is good
  – ...practices which avoid learning are double-plus ungood

• Plagiarism is bad
  – ...credit must be given where due

• “Outside code” is not a simple yes/no issue
  – You must not read any outside code without carefully consulting the syllabus
Academic conduct

• Being a partner
  – Responsible
    • I am writing three grad school applications next week
  – Irresponsible
    • [vanish for 1 week, drop class]
Closing

- “RISKS Digest” (en.wikipedia.org/wiki/RISKS_Digest)
  - Developers should read this
  - Managers should read this
  - Journalists should read this

- OSC textbook
  - Chapters 1, 2; Chapter 13.1, 13.2, 13.3.3

- OS:P+P textbook
  - Chapters 1, 2; Sections 3.0, 3.5; Section 11.3

- Start choosing a partner for P2/P3