

# Introduction to 15-410

Dave Eckhardt  
[de0u@andrew.cmu.edu](mailto:de0u@andrew.cmu.edu)

# Wait-List Issues

- There isn't room for everybody on the wait list
  - Some students will need to try again next semester
  - Sorry!
- If you're not on the wait list yet, you are *invisible*
  - Invisible students *definitely* won't get into the course!
  - If you are invisible, send mail *before noon today*
- Registrar's wait-list order is an illusion
  - Not an automatic queue; we admit based on readiness
  - Most wait-list enrollments will happen later today

# Self-Assessment

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

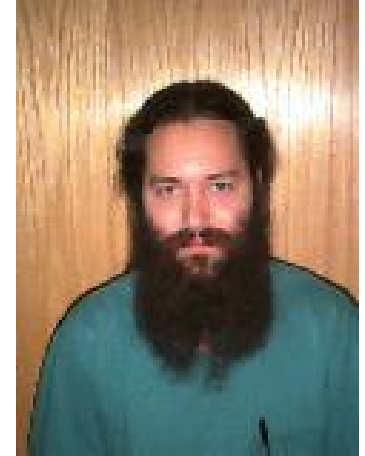
# Textbook

- What we're using
  - Operating System Concepts, 8<sup>th</sup> edition
    - Silberschatz, Galvin, & Gagne
  - CMU Bookstore should have copies of text
- Multiple “cheap textbook” options exist!
  - If you try an e-book edition instead of paper, please tell us if you like it
  - Used copies of 7<sup>th</sup> edition work pretty well
    - Web site lists course reading assignments for 6<sup>th</sup> through 8<sup>th</sup> editions

# Outline

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

# Dave Eckhardt



- Associate Teaching Professor, CS
  - Ph.D., Computer Science, CMU, May 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”

# TA's

- 5 “repeat offenders”
  - Josiah Boning, Ryan Pearl
  - Caroline Buckey, Alex Crichton, Eric Faust
- New!
  - TBD (how mysterious!)
- As a team
  - Strong background
  - Here to help!

# Yinz - Background

- Junior/senior/other?
- CS/ECE/INI/other?
- Group programming before?
- Done a branch merge before?



# 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 use a linked list for ...?
  - A: [academic.cs.15-410.qa](#)
  - Reading this will be to your benefit
- Q: Important announcements from course staff...?
  - A: [academic.cs.15-410.announce](#)
  - *You are responsible for reading this often*

# Information Sources

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](mailto:staff-410@cs.cmu.edu)

# Information Sources

Q: I am experiencing [delicate situation X] ...

A: e-mail to faculty

# Health Problems

- *Somebody* will probably get mono or pneumonia
  - If not, only because of something more creative
- Work-blocking health problem?
  - Go *early* to Student 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 administrator will inform instructors
    - CS: cathyf@cs ; ECE: jmpeters@ece / jhsun@cmu ; ...

# 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]



# 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?
  - *Not* for instructor's convenience!
  - Allows attacking larger problems
  - Teaches *job skills* you will need
    - Very few “individual contributor” jobs, even academia
- Team programming != “software engineering”
  - No requirement analysis
  - No release staging, design for growth, ...
  - Not a complete “life cycle”

# Team programming - people skills

- Working with other people is *hard*
  - People think differently
  - People plan differently
- Pre-planning
  - Agree on work style, arrangements
    - Setting milestones
    - Pre-scheduled common time slots
- Handling problems
  - Involving “management” before it's too late
    - See syllabus

# Grading philosophy

- C – all parts of problem addressed
- B – solution is complete, stable, robust
- A – excellent
  - Somebody might want to re-use some of your code
- Numbers
  - A = 90-100%, B = 80-90%, ... (roughly)
- “Curving”? Maybe, not necessarily
  - Lots of A's would be *fine with us*
  - *But this requires clean, communicative code!*

# Closing

- comp.risks newsgroup
  - Developers should read this
  - Managers should read this
  - Journalists should read this
- Textbook
  - Chapters 1, 2
  - Chapter 13.1, 13.2, 13.3.3
- *Start choosing a partner for P2/P3*