Introduction to 15-410

Dave Eckhardt
de0u@andrew.cmu.edu

Outline

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

Dave Eckhardt

- Buzzword compliance
  - Ph.D., Computer Science, CMU, May 2002
  - “An Internet-style Approach to Managing Wireless Link Errors”
- Building Unix kernels since ~1985
  - PDP-11, Version 7 Unix
  - Not really a BSD bigot
- Don't go here
  - www.cs.cmu.edu/~davide/

Tadashi Okoshi

- Ph.D. Candidate, CS
- Background
  - Real-time operating systems
  - Network protocols
  - Smart cards
  - Mobile and ubiquitous computing

Undergraduate TA's

- Jonathan Giloni
  - Took 15-412 from me in the spring
- Joey Echeverria, Brian Railing
  - Had 15-412 from Greg Kesden
- As a team
  - Strong background
  - Recommended by past 412 TA's

Y'all - Background

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

Y'all - Reading

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

Y'all – Career plans

- Industry
- Graduate school
- Mountain top?

Information sources

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

Academic honesty

- See syllabus
- Learning is good
  - ...practices which avoid learning are ***double-plus ungood***
- Plagiarism is bad
  - ...credit ***must*** be given where due

Academic conduct

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

The deadline disaster

- “If you wait until the last minute, it takes only a minute!” -- Vince Cate
- Small problem
  - Your grade will probably suffer
- Big problem
  - ***Learning*** and ***retention*** require sleep

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
  - Skipping every lecture will challenge your grade
    - The map is not the terrain, the slides are not the lecture
    - How will you ask questions?

Course Plan

- Projects
  - Bare-machine video game **[1-person project]**
  - Thread library
  - OS kernel
  - Kernel extension
- Project environment
  - Virtutech Simics™ PC simulator
  - Can also run on real PC hardware

Course plan

- Homework assignments
  - ~3, 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
- Not “software engineering”
  - No requirement analysis
  - No release staging, design for growth, ...
  - Not a complete “life cycle”

Team programming – Styles

- Waterfall model
- Spiral model
- “Extreme Programming”
- “Pair Programming”
  - Williams & Kessler, Pair Programming

Interlude

- What is source code “for”?
  - What do we do with it?

Team programming - Design

- Decomposition into modules
  - (Yes, even in C)
- Design for ***team implementation***
  - Parallel effort may require adjustment

Team programming - Documentation

- For the non-compiler consumers of source code
- Doxygen documentation extraction system
  - Embed documentation in comments
  - Generate HTML index
  - Generate LaTeX
  - ...
- We intend to ***read your documentation***
- We intend to ***read your code***

Team programming - Source control

- Other buzzwords
  - Revision control, configuration management
- Goals
  - Re-create past builds
  - Compare stable states
  - Control inter-developer interference
  - [Manage multiple shipped product versions]

Team programming - Source control

- Even for “small” projects?
  - “It worked 3 hours ago, now it dies on start-up”
  - “I thought I fixed that already!”
- Survey of 15-412 students, 2003-05
  - Q: What should we do differently about source control?
  - A: “Make it mandatory”

Team programming - People skills

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

Warning about 15-213

- It's an important class
- We expect you to ***know***
  - Byte, word, register, 1<<2
  - Thread, stack
  - malloc(), free() (when & why)
- Trouble with 213?
  - Expect to spend extra effort on 410

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”
  - Lots of A's would be ***fine with me***

Closing

- comp.risks
  - Developers should read this
  - Managers should read this
  - Journalists should read this
- Textbook
  - Chapters 1, 2, 3
  - Chapter 13.1, 13.2, 13.3.3
- ***Start choosing a partner for P2***