Introduction to 15-410

Dave Eckhardt
de0u@andrew.cmu.edu
Bruce Maggs
bmm@cs.cmu.edu

Synchronization

- Textbook
  - Silberschatz, Galvin, & Gagne
    - Operating System Concepts, 7th edition
  - CMU Bookstore has copies of text
- If you plan to add, please do so *today*
  - Creating your personal AFS volume takes time
- Self-assessment exercise?
  - Not mandatory
  - A very good sanity-check, though

Outline

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

Bruce Maggs

- Buzzword compliance
  - Ph.D., Computer Science, MIT, September 1989
- Allegedly a theorist, but...
  - World’s Top Amateur Expert on DESQview/X
  - VP, Research, Akamai
- Campus photos from atop Hammerschlag Hall (and more!)
  - www.cs.cmu.edu/~bmm

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
  - Is Plan 9 any less esoteric than DESQview/X?
- Not as awful as it once was
  - http://www.cs.cmu.edu/~davide

TA's (so far)

- Nathaniel Wesley Filardo (not his full name)
  - A man of many talents (and majors)
  - Winner of the “mutex throw weight” award
  - Fourth time serving as a 410 TA
- Matt Brewer
  - A delectable mix of talent, taste, and discretion
  - Fewer middle names than Wes
    - Fewer ***first*** names than Wes
  - Second time serving as a 410 TA

TA's (so far)

- Mike Kasick
  - ECE B.S. alumnus (15-410), ECE Ph.D. student
  - Organizational powers of a dean
  - Ask him about obscure Linux kernel dynamic linking bugs on the Alpha
- Mystery TA
  - TBA

TA's

- As a team
  - Strong background
  - Here to help!

Your Background

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

Reading

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

Career plans

- Industry
- Graduate school
- Law/med/business school?
- Mountain top?

Information sources

- Web site http://www.cs.cmu.edu/~410
  - You are ***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 (hmm...)
  - ***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

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

Academic honesty

- See the syllabus. Read it carefully.
- 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
  - Why work super-hard only to forget?

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 every lecture 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
  - 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
- Team programming != “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
- What you choose is up to you
  - This is an opportunity to read about models

Team programming - Design

- Decomposition into modules
  - (Yes, we expect modularity even in C!)
- Design for ***team implementation***
  - May need to adjust design to work in parallel

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!”

- Most students who really try it keep using it

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

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***

Further Reading

- Sleep to Remember
  - Matthew P. Walker
  - American Scientist, July/August 2006
  - “The brain needs sleep before and after learning new things, regardless of the type of memory. Naps can help, but caffeine isn't an effective substitute.”