Advanced Programming

Henning Schulzrinne
Dept. of Computer Science
Columbia University

Overview

- Background
  - What is Advanced Programming?
  - Differences to "old" Software Engineering
  - Who should be here?
- Syllabus
- Logistics
- Grading

Background

- Experimental course – trial run for replacement for traditional CS 3156 (*Software Engineering*)
- Sequence:
  - Intro: modules, few dozen lines
  - DS: few hundred lines, stand-alone
  - AP: larger programs, different languages and styles
  - OS, networking, ISP, …: specialized knowledge

What is Advanced Programming?

- Become comfortable with multiple languages
- Multiple programming styles
- *Professional* programming:
  - program building
  - multi-person code development
  - systematic debugging
  - performance tuning
- Goal: robust, performing, industrial-strength code

Differences to *Software Engineering*

- Less emphasis on software engineering principles
  - more on tools and techniques
  - some SE is best read, rather than lectured
  - hopefully, SE will make more sense afterwards…
- CS4156 will remain
- Group programming project?

Who should be here?

- Know Java (and 313x), but no or little C/C++
- Likely doing systems or general track
- Want to do well in OS, DB, Networks, …
- Interested in how real systems work
- Affinity for programming, not complexity theory

Syllabus

- C programming language for Java programmers (2 weeks, 2 assignments)
  - summary of differences
  - pre-processor
  - strings
  - stdio
  - pointers and pointer arithmetic
  - malloc/free
  - unions
  - function pointers

Syllabus

- Software development models, including open source
- Software development tools:
  - Unix
  - make
  - automake
  - autoconf -- cross-platform development
  - CVS
  - doc++ and other documentation tools Oxygen
  - gdb and DDD
  - purify, ElectricFence
  - truss

Syllabus

- profiling: gprof
- Windows
  - Visual C
  - Code Warrior
- Program models: data driven, event driven, cgi, RPC, ...
- UML
- Data interchange
  - classical Unix style
  - serialization
  - XML

Syllabus

- Scripting languages
  - sh
  - awk, sed
  - Tcl, expect
  - Perl
  - Python
- Standard Unix system call interface:
  - file access
  - time and date manipulation
  - system files
  - process management
  - signals
  - getopt
  - syslog

Syllabus

- Internationalization
- secure programming
- design patterns
- web programming models (servlets, cgi, JavaScript, PHP, ...)
- web database interfaces (JDBC, mySQL, ...)
- Time permitting (these topics are also covered in CS 4118):
  - threads and concurrency
  - shared memory

Logistics

- Web page: http://www.cs.columbia.edu/~hgs/teaching/ap/
- Mailing list:
  - advanced-programming@cs.columbia.edu
  - Mailman (lists.cs.columbia.edu)
  - As-written or *digest* once a day
  - Web archive ~ newsgroup
- Office hours: Th, 5.30-7.30 pm, 815 CEPSR
- TA: Nikil Tiwari (nst8@columbia.edu)

Books - required

- *Advanced Programming in the Unix Environment*, by W. Richard Stevens, Addison-Wesley. ISBN 0201563177
- *Code Complete : A Practical Handbook of Software Construction*, by Steve C McConnell, Microsoft Press. ISBN 1556154844

Books – useful reading

- *Mythical Man Month*, by Frederick P. Brooks, Addison-Wesley. ISBN 0201835959
- *Building Secure Software*, by John Viega and Gary McGraw, Addison Wesley. ISBN 020172152X
- *Software Project Survival Guide*, by Steve C McConnell, Microsoft Press. ISBN 0201563177

Grading

- Midterm 30%, final exam 40%, homework assignments 30%
- Group project(s) - NO

Assignments

- Weekly programming assignments
- To be done individually.
- Conform to coding style guidelines.
- You **will lose points** if you don't follow the guidelines.

Ombudsperson

- New course – new problems
- Designated, random individual
- Act as relay for "issues"
- Assumed to speak for (subsection of) class, not themselves
- Quicker than mid-term review