COMP 2400 UNIX Tools

Christian Grothoff
christian@grothoff.org

http://grothoff.org/christian/

README

http://grothoff.org/christian/teaching/2007/2400/

Overview

- UNIX tools $\equiv$ productive programming
- **Productive:** UNIX shell, scripting, debugging, profiling, static analysis
  Programming: Implement realistic projects in C, C++ and Java
- You will write code in C, C++ and Java.
- You must already be able to write simple algorithms in either C, C++ or Java.

Academic dishonesty

- Webpage says what is allowed.
- If in doubt, ask first.
- Cheating can be detected with automated tools.
- Any violation will be reported to the dean.

Expectations

- Read the indicated chapters of the textbook – not every detail is covered in class, but it may still be helpful in exams!
- Study additional material (software documentation, other books, additional textbook chapters) as needed.
- Deliver tested, documented, packaged and working versions of projects on time using subversion.
- Demo your projects in class.
- Answer questions in midterm and final exams.

Programming Assignments: The Rules

- You need 30 points in both C and Java (and you also cannot do more)
- Points are indicated for each project
- You can freely form groups, project points wil be divided among the team members
- You are allowed to suggest alternative projects, given a specification they will be rejected or assigned a number of points

‘Grading Criteria

50% Correctness, as established by testing

25% Coding style (formatting, variable naming, API design)

15% Project documentation (source & end-user)

15% Performance study (implementation, documentation, achieved performance)

15% Packaging and build system

You can get more than 100% of the points if you excel in all areas.

Suggested Programming Assignments: C

- A* algorithm (40)
- Fast `class`-file parser library (40)
- Fast `RTF`-file parser library
- Flexible `GtkCellRenderer` (60)
- Chess game with AI using Glade (60)

Suggested Programming Assignments: Java

- Hashtable with iterators (10)
- Persistent suffix-tree (40)
- 4x4x4 game (40)

Suggested Programming Assignments: any-language

- Data recovery tool (40)

Questions

?

Why study UNIX tools?

Why study UNIX tools?

- UNIX is a key reference point for all modern operating systems
- UNIX influenced Linux, Solaris, BSD, OS X and Windows NT/XP
- UNIX tools are available on all of the above platforms
- Knowing the right UNIX tool for a long list of tasks can boost productivity by many orders of magnitude

Why study SVR4/POSIX?

- SVR4/POSIX are a standardized UNIX APIs for C programming
- Standardization ensures broad availability and gives common terminology
- Core of the SVR4/POSIX APIs are also available on Windows
- SVR4/POSIX defines what is commonly seen as the essential functionality of any modern operating system
- High-level Java APIs can often be directly related back to SVR4/POSIX functionality

Why study Java?

- Large amount of code written in Java
- Memory safe language $\Rightarrow$ fewer bugs!
- Knowing Java makes it easier to learn C++, C#, Scala, X10, ...
- You will need good Java skills for COMP 3351

Questions

?

XP: eXtreme programming

XP is a software engineering methodology:

1. XP runs counter to almost all software engineering practice
2. XP is not a solution for all problems (mostly for smaller teams)
3. XP is a programmer friendly "religion"

Software Development Challenges

1. Schedule slips – worst case: project canceled
2. System goes sour – more bugs over time until so buggy that it is unusable
3. Business misunderstood or business changed – software useless
4. Software has unused features – hard to maintain, too costly
5. Staff turnover – nobody around to maintain the code

Questions

?

General Homework Hints

- $ svn add filename ; svn commit -m “logmessage”
- $ gcc -o binary sourcename.c ; ./binary
- $ latex filename.tex ; xdvi filename.dvi
- $ javac pack/Type.java ; java pack.Type

Homework Summary

Before the next lecture:

- Generate password with `htpasswd` and register account.
- Read the first chapters of the subversion manual and "Introduction to the Unix shell".
- Install software (or use department machines).
- Implement "Hello World", test and submit!
- Form groups for the Java project and decide on a project!

Questions

?