CS 253: Intro to Systems Programming

Catalog Description

CS 253 INTRODUCTION TO SYSTEMS PROGRAMMING (3-0-3)(F,S). Structure of C programs, function pointers, variable argument lists, other generic programming techniques. Introduction to build systems, debugging techniques and process management. Basic systems programming including topics such as streams, buffers and pipes, system calls, multi-threading, and libraries for Linux and Microsoft Windows. PREREQ: CS 221.

Learning Objectives

By taking this the course the student will be able to:

- design and develop programs of moderate complexity in C,
- translate their knowledge of object-oriented programming in Java to C,
- use various tools like editors, build tools, debuggers, version control and memory checkers to improve their productivity,
- use shell commands and system utilities, and
- use basic system calls related to files, processes and threads.

Textbooks and References

- Lecture notes (on class website).
  URL: http://cs.boisestate.edu/~amit/teaching/handouts/cs-unix.pdf
- The UNIX Programming Environment, Kernighan and Pike.
- Managing Projects with GNU Make, Robert Meclenburg

Teaching Assistants

Piazza

This term we will be using Piazza for class discussion. The system is highly catered to getting you help fast and efficiently from classmates, the TA, and myself. Rather than emailing questions to the teaching staff, I encourage you to post your questions on Piazza. Piazza will also be used for disseminating information. Subscription is required. Students are responsible for knowing anything posted via Piazza.

Attendance

Students are expected to attend all classes. Missing classes without explanation may result in a grade penalty.

Exams and Quizzes

In-Class Quizzes

Quizzes will be given in class over assigned reading or material covered previously in class.

- In-class quizzes must be taken in class on the day they are given. Make-up quizzes will not be granted other than for exceptional reasons.
- Credit for group portion of the quizzes will only be given to group members who participate in the quiz.
- The individual portion is worth 65% and the group portion is worth 35% of the total quiz score.
- If 80% of students complete the end-of-semester course evaluations, then the lowest quiz score will be dropped.

Final Exam

The final exam date is fixed and shown below. Unless alternate arrangements are made in advance, only officially excused absences will be accepted for missing an exam. Any resources allowed for exams will be at the instructor’s discretion.

Final exam date: See class website.

Grading Policy

- Programming assignments: 40%
- In-class quizzes: 30%
− Individual Quizzes: 65%
− Group Quizzes: 35%

• Final exam: 30%

• All submitted code must run on the onyx cluster. A score of 0 will be awarded for code that does not compile or run on the onyx cluster.

**Academic Honesty**

Students are expected to work on their own unless explicitly instructed otherwise. Students who copy from each other or from any other source on assignments will be considered to be cheating as will students who allow their work to be copied. This includes trying to find answers to problems or programs from the Internet or other sources (and uploading your completed assignments to Internet sites that are publicly accessible).

The official University Academic Dishonesty Policy (reproduced below) may be found at [http://registrar.boisestate.edu/general-information-and-policies/academic-integrity/](http://registrar.boisestate.edu/general-information-and-policies/academic-integrity/)

*Any violation of the Academic Honesty policy is grounds for immediate failure of the course and will be reported.*
Course Topics

• Introduction

• Topics in C
  – Tutorial introduction to C
  – Types, operators and expressions
  – Control flow
  – Functions and program structure
  – Pointers and arrays (and C-style strings)
  – Structures
  – Input and output
  – Variable argument lists: C style method overloading
  – Function pointers and generic programming in C (with case studies)
  – Creating and using shared libraries in C
  – Plugins in C (aka Method overriding using loadable code)

• Tools and Techniques
  – Make: object-oriented build process
  – Debugging Techniques using Eclipse, GDB, DDD
  – Memory checker: Valgrind
  – Subversion: introduction to version control

• The Linux/Unix Environment

• Shell scripts and system utilities
  – Advanced command-line tools
  – Object oriented use of Unix filters and pipes
  – Basic shell scripting

• Basic systems programming
  – Linux Systems Programming: files, buffers, processes, signals and pipes.
  – Multi-threaded programming with PThreads library.
  – Multi-threaded programming in Microsoft Windows API.