CMPSC 311 - Introduction to Systems Programming

Module: Introduction

Professor Patrick McDaniel
Fall 2013
Who makes systems?
Who makes systems?
Who makes systems?
This course ... 

• This course is a **systems** course covering general topics in the creation of systems:
  
  ‣ System programming concerns the development of **software modules** and **services** used by applications, e.g., web-servers, search tools, backup systems, etc.
  
  ‣ This course will provide information and experience required to understand, design and implement components of large and small software systems.
You need to understand ...

- Basic programming
  - C++ or Java, CMPSC211

- Data structures

- Discrete Mathematics

- Basic computer science literacy
  - E.g., File systems, tools, algorithms, etc.
Goals

‣ My goal: *to provide you with the tools to professionally develop systems.*
  ‣ Basic technologies
  ‣ Engineering/performance/feature trade-offs
  ‣ How to design, develop and maintain code

• *This is going to be a hard course.* The key to success is sustained effort. Failure to keep up with readings and project will likely result in poor grades, and ultimately little understanding of the course material.

• *Pay-off:* systems programming skills are essential to a career in computer science related professions.
Course Materials

• Website - I am maintaining the course website at
  ‣ http://www.cse.psu.edu/~mcdaniel/cmpsc311-f13/

• Course assignments, slides, and other artifacts will be made available on the course website.

• Course textbooks
  ‣ *Computer Systems: A Programmer's Perspective*, Bryant and O'Hallaron
  ‣ *C Programming: A Modern Approach*, K.N. King
  ‣ *(Optional)* *Harlen Hahn's Guide to Unix & Linux*, Hahn
Course Calendar

• The course calendar as all the relevant readings, assignments and test dates

• The calendar page contains electronic links to online materials assigned for course readings.

• Please check the website frequently for announcements and changes to the schedule. Students are responsible for any change on the schedule.
Grading

• The course will be graded on exams, assignments, projects, and class participation in the following proportions:

  40% Course Programming Projects
  12.5% Mid-term Exam 1
  12.5% Mid-term Exam 2
  25% Final Exam
  10% Class Participation
There are a large amount of readings in this course covering various topics. These assignments are intended to:

- Support the lectures in the course (provide clarity)
- Augment the lectures and provide a broader exposure to systems programming topics.

Students are required to do the reading!

About 10-20% of questions on the tests will be off the reading on topics that were not covered in class. You better do the reading or you are going to be in deep trouble when it comes to grades.
Course Projects

• There will be 6-7 course projects that will require students to *individually* develop some non-trivial systems functionality.
  ‣ Students will have 1-4 weeks to complete each assignment
  ‣ Programs *must be* in C as per the assignment
  ‣ You will be installing your own Linux VM to manage and develop code on (your responsibility)

• Programs will be graded on
  ‣ correct function
  ‣ clarity/quality of design and code
  ‣ documentation (as required)
  ‣ other factors as described by assignment, e.g., performance
Course Projects

• There will be 6-7 course projects that will require students to individually develop some non-trivial systems functionality.
  ‣ Students will have 1-4 weeks to complete each assignment.
  ‣ Programs must be in C as per the assignment.
  ‣ You will be installing your own Linux VM to manage and develop code on (your responsibility).

• Programs will be graded on:
  ‣ correct function
  ‣ clarity/quality of design and code
  ‣ documentation (as required)
  ‣ other factors as described by assignment, e.g., performance

Policy: All course projects are to be carried out individually. Students are explicitly not allowed to share information, source code, or even discuss the contents of the projects. Students are also forbidden from copying code from the Internet. Any violation of these policies will be considered a violation of ethical conduct (cheating) and will result in the student being removed from the class and assigned an 'F' grade. There will be no exceptions given for any reason.
Ethics Statement

• This course considers topics involving systems. As part of this investigation we will cover technologies whose abuse may infringe on the rights of others. As an instructor, I rely on the ethical use of these technologies. Unethical use may include circumvention of existing security or privacy measurements for any purpose, or the dissemination, promotion, or exploitation of vulnerabilities of these services. Exceptions to these guidelines may occur in the process of reporting vulnerabilities through public and authoritative channels. Any activity outside the letter or spirit of these guidelines will be reported to the proper authorities and may result in dismissal from the class and or institution.

• When in doubt, please contact the instructor for advice. Do not undertake any action which could be perceived as technology misuse anywhere and/or under any circumstances unless you have received explicit permission from Professor McDaniel.
A last (important) note ...

• Show me the respect not to ...
  ‣ Surf, text, email, or other activity in my lecture.
  ‣ Show up late.
  ‣ Lie to me.

• I take this very seriously.
Class Structure

1. Systems Programming Basics
   • Systems architecture
   • Systems programming (C)
   • Systems administration

2. Systems Programming Tasks
   • Debugging
   • Profiling
   • Version control systems

3. Advanced Systems Programming
   • Structured data (e.g., XML)
   • Network programming
   • Third-party library integration
Homework Assignment #1

• See handout.