Systems and Internet Infrastructure Security

Institute for Networking and Security Research
Department of Computer Science and Engineering
Pennsylvania State University, University Park, PA

Introduction to CMPSC 311

Devin J. Pohly <djpohly@cse.psu.edu>

Who am I?

- Devin Pohly
- Grad student in systems security
- Systems programming is what I do every day

Name ? Link

A B C D E F G H I J K L M
N O P Q R S T U V W X Y Z
a b c d e f g h i j k l m
n o p q r s t u v w x y z
1 2 3 4 5 6 7 8 9 0 . -
←
END
A-Decide ● B-Cancel

One other person to know

- Prof. Patrick McDaniel
- Redesigned 311 this year
- Overseeing the course behind the scenes

About this course

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

About This Computer

System Software 7.6.1
© Apple Computer, Inc. 1983-1996

Macintosh

Total Memory : 12,288K Largest Unused Block : 8,819K

System Software 3,442K

Things you should know

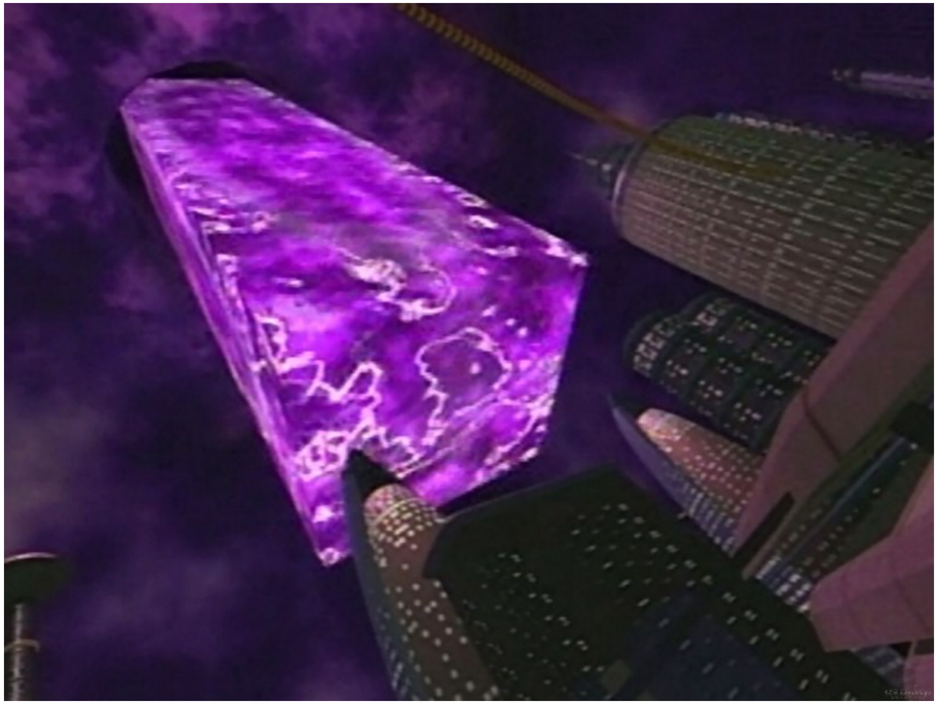
- Basic programming
  - C++ or Java, CMPSC221
- Data structures
- Discrete mathematics
- Basic computer science literacy
  - File systems, tools, algorithms, etc.

Goals

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

Warning!

- This is going to be a hard course. The key to success is sustained effort. Failure to keep up with readings and projects 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 Website

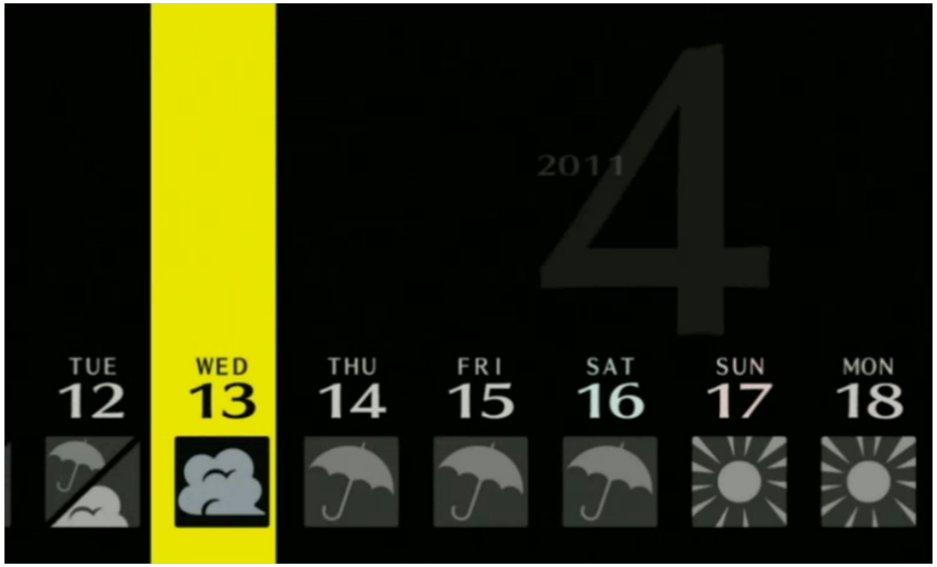
- Website: I am maintaining the course website at

  `http://www.cse.psu.edu/~djp284/cmpsc311-s14/`

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

Course Calendar

- The course calendar has 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

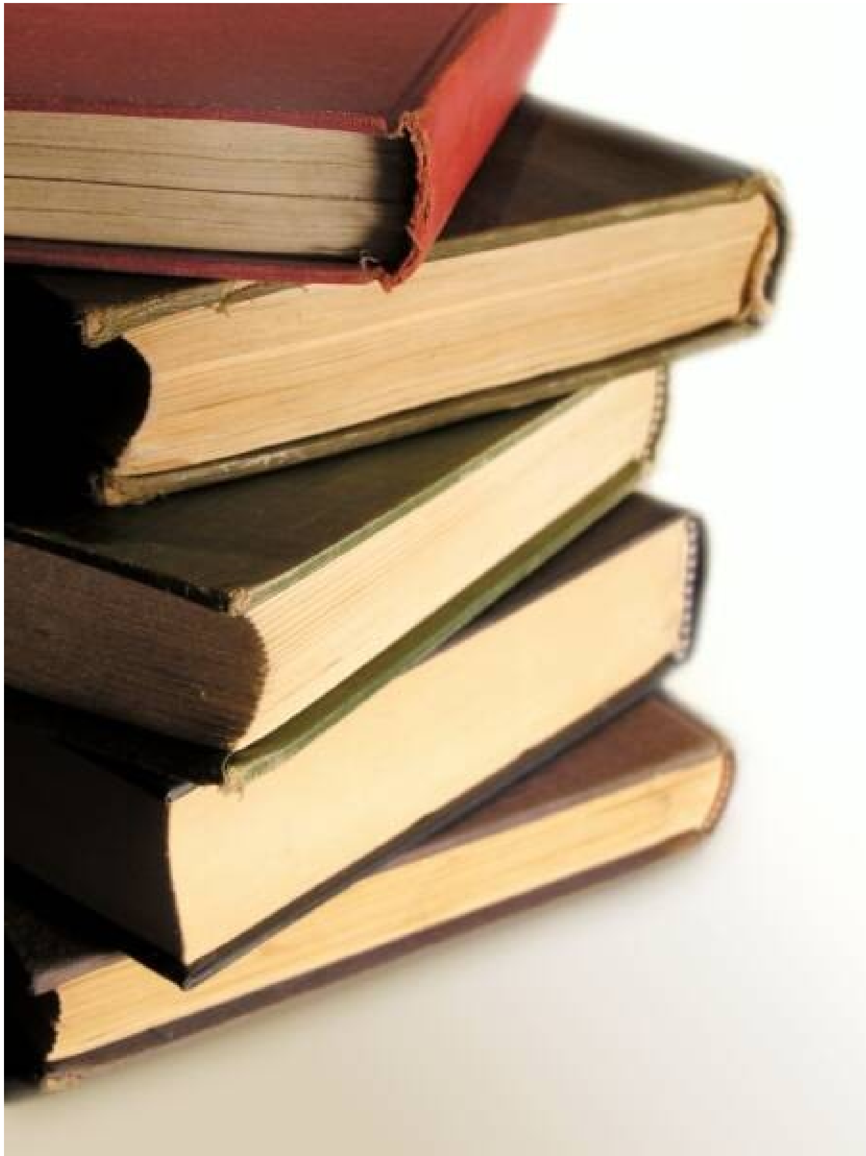
- The course will be graded on exams, projects, and class participation in the following proportions:
  - 40% Course Programming Projects
  - 12.5% Mid-term Exam 1
  - 12.5% Mid-term Exam 2
  - 25% Final Exam
  - 10% Class Participation

Readings

- 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 exams will be off the reading on topics that were not covered in class*. If you skip the readings, you will be in deep trouble when it comes to grades.

Course Textbooks

- *Computer Systems: A Programmer's Perspective*, Bryant and O'Hallaron
- *C Programming: A Modern Approach*, King
- (Optional) *Harley Hahn's Guide to Unix and Linux*, Hahn



Course Projects

- There will be 3–5 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 described by assignment, e.g., performance

Course Projects

- There will be 3–5 course projects that will require students to individually develop some non-trivial systems f

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.**

op

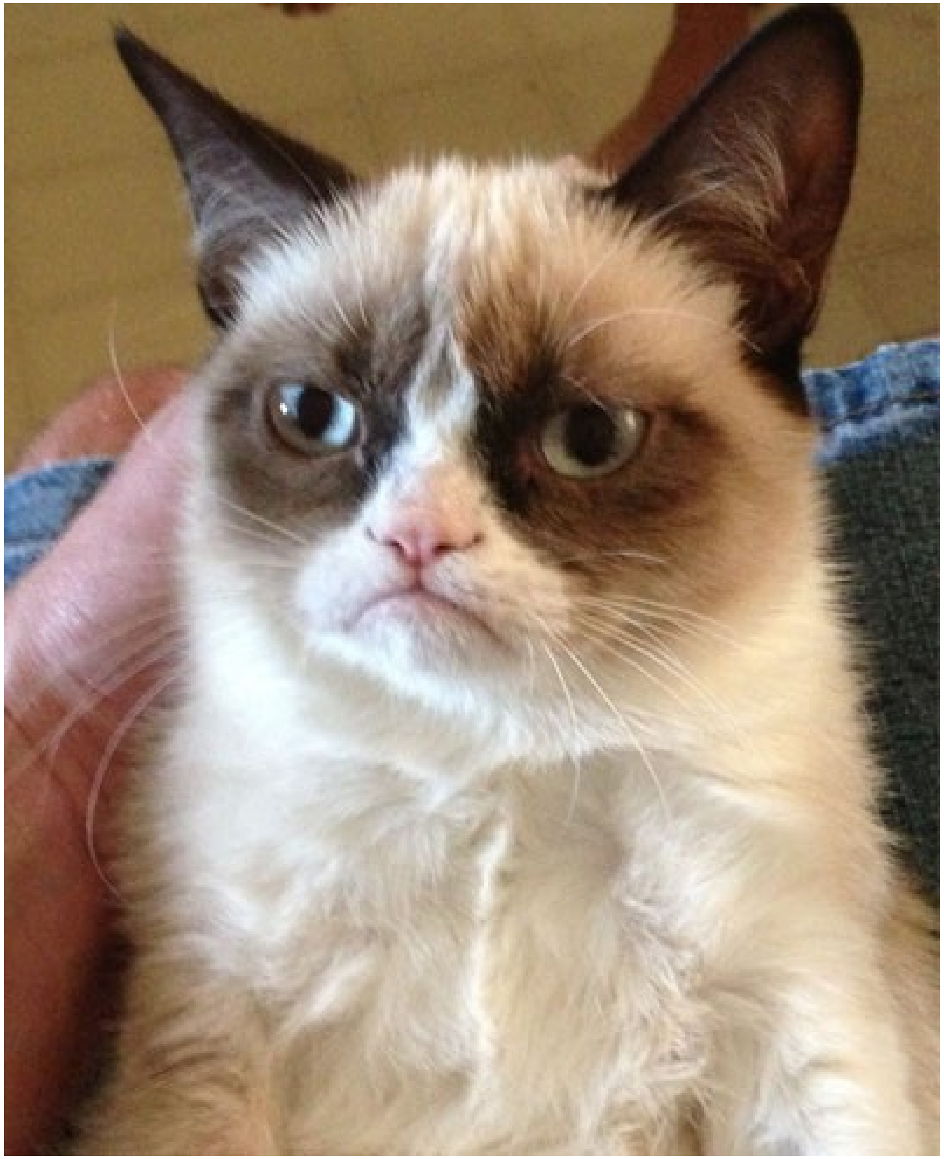
- P
  - documentation (as required)
  - other factors described by assignment, e.g., performance

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 measures 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...

- Please have the courtesy not to:
  - Surf, text, email, or other activity in lecture
  - Show up late
  - Lie to me
- I take this very seriously.



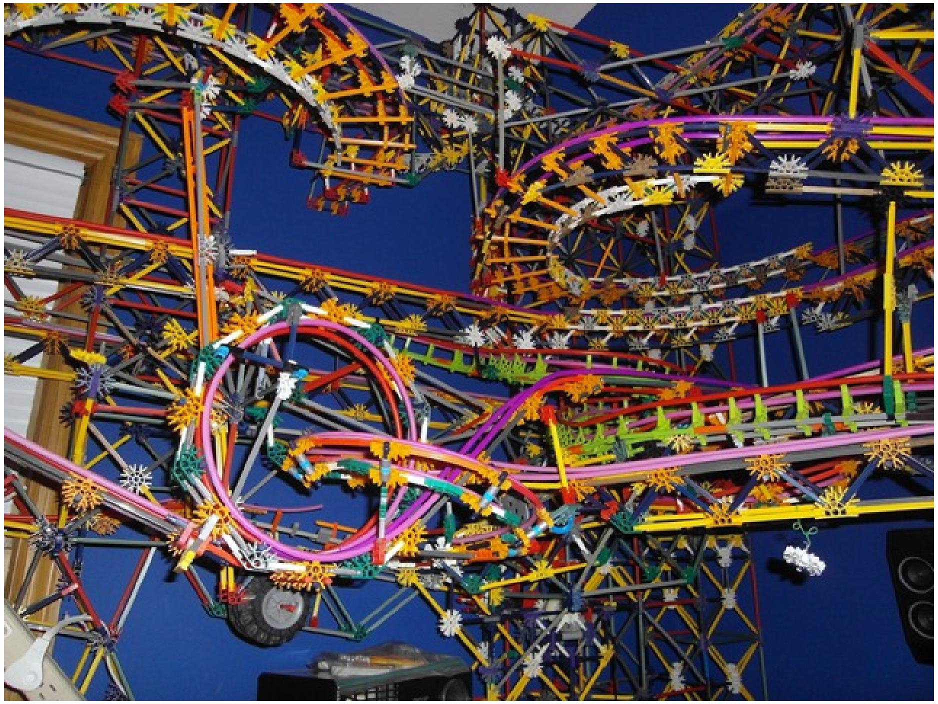
Talk to me!

- If you find yourself in a sticky situation
  - The sooner the better
- If you have any questions
- If you have any feedback

- Email or office hours

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

- Just some basic setup
- See handout
- MUST be done before you can attend class on Wednesday!

Any questions?