CIS 415 – Operating Systems -
Introduction
Joe Sventek
Announcements
Course Outline

- Logistics
- What is CIS 415?
- What is expected of you?
- What will you learn in CIS 415?
Course Logistics

- **Lecture time**
  - CRN 11662: Tue/Thu @ 12:00-13:20, 208 Deady Hall

- **Midterm**
  - Thursday, 26 October, 12:00-13:20, 208 Deady Hall

- **Final**
  - Monday, 4 December, 8:00-10:00, 208 Deady Hall

- **Undergraduate course prerequisites**
  - CIS 313, CIS 314, CIS 330

- **Discussion/Lab**
  - CRN 11663: Wednesday, 10:00-10:50, 26 Klamath Hall
  - CRN 11664: Wednesday, 16:00-16:50, 26 Klamath Hall
  - CRN 11665: Friday, 12:00-12:50, 26 Klamath Hall
Personnel

- **Instructor**
  - Joe Sventek ([jsventek@uoregon.edu](mailto:jsventek@uoregon.edu))
    - Internet of Things
    - Complex Event Processing
  - Office hours, 358 Deschutes Hall
    - Tue 2:00-3:20, Wed 12:00-1:20
    - And by appointment

- **Teaching assistant**
  - Roscoe Casita ([rcasita@uoregon.edu](mailto:rcasita@uoregon.edu))
  - Office hours, 100 Deschutes
    - TBA
Resources


- CIS 415 web page: [https://www.cs.uoregon.edu/classes/17F/cis415/](https://www.cs.uoregon.edu/classes/17F/cis415/)

- Discussion board on Piazza: [https://piazza.com/class/j7s460yu3cs16j](https://piazza.com/class/j7s460yu3cs16j)

- We use Canvas for turning in projects, assignments, and accessing files not already on the web site.
Course Structure

- Lectures (Prof Sventek)
  - Focus on core concepts
  - Quizzes and exams

- Lab sessions (TBA)
  - Present material needed for programming assignments – C and Unix, threads, signals, etc
  - Provide programming assignment assistance
  - Tutorials and practice sessions

- Grading
  - 10% lab attendance
  - 20% midterm (26 October, in class)
  - 40% final exam (4 December, 8:00-10:00, 208 Deady Hall)
  - 30% programming projects (3 individual projects)
  - Extra Credit programming project in week 10 worth an additional 10%

- Ungraded homework assignments are made throughout the quarter. These provide you with exercises similar to those found on the midterm and final exams.
Course Plan

- **Topics covered (18 lectures)**
  - Heap review & ADTs, OS Overview
  - OS structures and system calls, Processes
  - IPC, Threads and Thread-safe ADTs
  - CPU scheduling, Real-time systems
  - Synchronization, Deadlocks
  - Main memory, Virtual memory, Cache memory
  - Virtual machines, File system interface
  - File system implementation, IO and mass-storage systems
  - Protection

- **The Schedule lists all relevant readings, assignments, test dates**
  - Links to online papers assigned for course readings
  - Supplements to OSC book

- **Check course web page for announcements and updates**
Lectures

- OSC book and online materials are your main sources for broader/deeper OS information
- Most lectures will stick close to the OSC book content
  - Cover fundamental topics of more importance
  - Cannot cover everything in a single quarter
  - Other materials are provided for lectures 1, 8 and 13
- Lectures will complement programming component with respect to overall ideas, but the online materials will be more useful for implementation
What is expected of you?

- **Background**
  - CIS 313 – intermediate data structures
  - CIS 314 – computer organization and architecture
  - CIS 330 – C/C++ and Unix programming

- **I expect you to maturely engage with the course**
  - Attend lectures (I discuss material that is not in the book or the notes)
  - Attend labs (use of tools that you must master to successfully tackle the projects)
  - Submit projects on time – **N.B. no late submissions will be accepted!**
  - Take advantage of office hours to help you understand the material.
  - Read handout material before turning to Piazza

- **Effectively use source material, online documentation, books, and the Internet to look things up**

- **Persistence – stay on top of the work**
What will you get out of CIS 415?

- **My goals**
  - Provide you with an understanding of the fundamentals of modern operating systems
  - Provide you with in-depth practical experience in working with the Linux OS and system programming tools

- **Your goals**
  - Commit to a challenging course
  - Maintain sustained effort throughout the quarter – failure to do so will reduce learning

- **Pay-off**
  - OS knowledge is fundamental
  - Systems programming skills are highly marketable

- **As with all courses, you will get out of it what you put into it!**
Course Projects

- Best way to understand the material is by doing
- Focused on programming in a Linux environment and understanding systems issues
- All are individual projects
- Learning targets:
  - Project 0 – sophisticated C programming of ADTs
  - Project 1 – build a process scheduler using the process control system calls in Linux
  - Project 2 – build a concurrent device driver using Pthread mechanisms
Environment for Projects

- You have been provided with 64-bit and 32-bit Arch Linux images for use with Oracle’s VirtualBox.
- I do not prescribe which environment you use for developing and testing your code.
- Your submission will be graded on a 64-bit Arch Linux system running in VirtualBox.
- Therefore, it is imperative that you test your code in the Arch Linux environment before you submit; things that work perfectly well in your development/testing environment will almost certainly work differently in the VM.
- If you do not/cannot install and run VirtualBox on your personal system, VB is installed on all systems in 100 Deschutes and 26 Klamath for your use.
Course Schedule

- The CIS 415 web site has a “Logistics” link which takes you to a page with the course schedule
- It shows lecture topics and provides links to slides
- It shows the readings that you should complete before each lecture, with links to online documents beyond the book
- It shows assignments and due dates
- There may be changes to the schedule
  - Students are responsible for checking the schedule for changes
Academic Misconduct

- The programming projects are individual efforts
- You may discuss the projects with your classmates; it is fine to share pseudocode
- It is statistically impossible to submit identical source code unless you have coded it together or you have copied it from another student
- Do **NOT** share your source code with another student, no matter how pitiful they sound
- If I detect collusion (submission of identical source code), all individuals involved will receive an F in the course immediately; note that changing the spacing or variable names in an attempt to hide collusion is a known technique and our tools will find it.
- If I detect anyone using a rent-a-coder, not only will you receive an F, but I will actively seek to have you expelled from the University.
- If you collude, **I will catch you.**
Acknowledgements

- The content of CIS 415 was created with content obtained from several sources with permission.
- The following contributions are particularly acknowledged:
  - Prof Kevin Butler’s course in Spring 2014 provided excellent lecture material and structure for programming projects.
  - Prof A Malony’s course from Fall 2014 also provided excellent material.
  - The OSC book comes with instructor materials, including chapter-by-chapter lecture slides.
  - “What every Programmer Should Know About Memory” by Ulrich Drepper of Red Hat Inc., from which the Cache handout was derived.