CSCI 5103
Operating Systems

Instructor: Abhishek Chandra

General Information
- Class: Tu Th 11.15 am-12.30 pm
- Office Hours: Tu Th 10-11 am
  - Location: Keller Hall 4-209
- Teaching Assistant: Jack Perisich
  - Office Hrs: TBA
- Course Web page: http://www-users.cselabs.umn.edu/classes/Spring-2020/csci5103/
  Also a Canvas page

Course Objectives
- Learn Operating System abstractions and principles applicable to a variety of OS’s
  - Processes, File Systems, Virtual Memory
- Learn concepts applicable to variety of systems
  - Concurrency, Synchronization, Scheduling
- Focus on internals of OS
  - OS structure and components
  - OS design principles
  - OS implementation: Data structures, algorithms
- Case studies to illustrate various principles

What you will learn in this course
- At the end of the course, you should be able to:
  - Understand OS structure and components
  - Understand internal workings of OS
  - Understand performance tradeoffs in OS design and implementation
  - Be able to implement some OS structures
  - Build better programs and systems!
**Pre-requisites**

- CSCI 2021 (Computer Architecture): Requires good understanding of computer organization and hardware concepts
- CSCI 4061 or equivalent (helpful)
  - Familiarity with general OS concepts
  - Programming experience with C
  - Comfortable with UNIX environment
- Strong programming experience
- Good understanding of data structures and algorithm fundamentals

**CSCI 4061 vs. CSCI 5103**

- 4061:
  - Focus on external view of OS
  - How to use OS APIs in programs
  - Focused on UNIX APIs
- 5103:
  - Focus on internal structure of OS
  - How to build parts of OS
  - Broader coverage of OS's
- Example:
  - 4061: Process fork and exec
  - 5103: Process representation and scheduling

**What you won’t be taught in the class**

- Unix Tools: You are expected to be comfortable working in a UNIX/Linux environment
- APIs and system calls:
  - Learnt in 4061
  - Brief discussion where needed
- C/C++/Java programming: You are expected to pick it up yourself
  - Helpful to have programmed in earlier classes
- Distributed Systems: See CSCI 5105

**Textbooks**

- **Required:** "Operating System Concepts (10th Edition)" by Silberschatz, Galvin and Gagne
- Weekly readings from the textbook(s), lecture notes and additional material
Course Work

- 3 Programming Assignments (45%)
  - Work in **groups of 2**
  - Due in ~2 weeks
- 3 Written Assignments (15%)
  - To be done **individually**
  - Due in a week
- Exams (40%):
  - 1 Mid-Term (15%)
  - 1 Final (25%)

Programming Assignments

- Programming environment:
  - OS Simulator
  - Provides basic OS functionality
- Goal: Implement specifications provided
  - Extend OS functionality based on concepts learnt in class
  - Systematic evaluation for performance, tradeoffs

Programming Assignments: Submission

- One submission per team
- Provide code and any other files
- Report: Describe program design and evaluation
- Online submission by 11:59 pm on due date (via Canvas)
- Late penalty: 10% for <24 hrs, +30% each extra day (open to change under certain circumstances)

Programming Assignments (contd.)

- **The submitted code should be original**
  - **DO NOT** copy or derive from the Web or other external sources (e.g., prior offerings, senior students, programmer friend, ...)
  - **No sharing** of code across teams
- Discuss and ask questions on class forum, from TA or instructor
- Grading: Points for
  - Functionality and Correctness
  - Program Design and Evaluation
  - Documentation and Code readability
**Written Assignments**

- Based on concepts discussed in previous 3-4 lectures
  - Goal is to test your understanding, practice solving problems
- Have to be done **individually**
  - Not with your project teammates
  - All answers must be original, *in your own words*
  - Do not copy or search for solutions from others, Web, etc.
- Due at beginning of lecture on due date

**Exams**

- Mid-Term exam would cover the material of first half of the course
- Final exam will be comprehensive
- Closed notes/closed book
- No electronic devices allowed

**Class Discussion Forum**

- On Canvas
- You can post questions, discuss topics, course material
- Try responding to each other as far as possible
- Instructor, TA will regularly monitor the forum
- Please avoid:
  - Irrelevant mails, flame wars
  - Posts that break the rules/spirit of honesty

**Class Participation**

- Engage in class
- Ask questions, answer to queries, initiate and respond to discussion
- Also use the Class Forum
**Academic Dishonesty**

- What does it include?
  - Copying assignments, cheating on exams, plagiarism
  - Written homework must be done by yourself – do not copy from textbook, web or others
  - Code should be original (not copied or derived from the web or other sources)
  - Providing help is also considered cheating
- Can result in **serious consequences**:
  - Can range from 0 on assignment to F in class
  - U requires report to Office of Student Affairs
  - See Dept. Academic Conduct Policy on class website
  - If unsure, just ask!

**Disability Statement**

- If you have, or think you have, a disability, contact Disability Services
- Please get a letter from DS for any special accommodation request on course work
- I will try my best to make the required accommodations

**UNITE Mechanics**

- Lecture available on streaming video
  - Live to off-campus students
  - With 10 days delay to on-campus students
- Off-campus students can phone-in
- Assignments to be handed to UNITE coordinator
  - Timestamped by due date/time
- Exam can be given on-campus or arranged with UNITE coordinator

**What is an Operating System?**

-
Examples of Operating Systems?

What does an OS do?
- What services does an OS provide?

What is an Operating System?
- Extended Machine
- Resource Manager
- Control Program

Extended Machine
- Simple abstractions of hardware resources
  - CPU -> Processes, Threads
  - Memory -> Virtual Memory
  - Disks -> Files
  - Network interfaces -> Sockets
- **Goal**: Simple, easy to use
**Resource Manager**
- Efficient utilization of resources and performance
  - Good CPU utilization
  - Good I/O throughput
- Multiplexing of resources among multiple users, programs
  - Multiple processes on same CPU
  - Multiple files on the same disk
  - Multiple connections on same network link
- **Goal:** Maximum system performance

**Control Program**
- Handle concurrent and asynchronous events
  - User typing commands on keyboard
  - Bytes being read from the disk
  - Packets arriving on the network interface
- **Protection and Security**
  - Prevent runaway programs from hogging CPU
  - Prevent malicious programs from corrupting memory
- **Goal:** Ensure correctness and fairness

**Computer System**

<table>
<thead>
<tr>
<th>Hardware</th>
<th>Applications</th>
<th>Operating System</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU, Memory, Disks, Devices</td>
<td>User Programs</td>
<td>Processes, File System, Virtual Memory, Threads, Sockets</td>
</tr>
<tr>
<td></td>
<td>Shells, Tools and Utilities</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CPU, Memory, Disks, Devices</td>
</tr>
</tbody>
</table>

**Course Topics**
- OS design and structure
- Process management and scheduling
- Threads and synchronization
- Memory management
- File system implementation
- Mass storage and I/O management