GMU Spring 2021
CS 531 - Computer Systems and Fundamentals of Systems Programming

Instructor: Hal Greenwald (hgreenwa@gmu.edu)
Thursday 7:20pm - 10:00pm ONLINE
Office hours by appointment

CS 531 provides a hands-on introduction to systems programming with an emphasis on data structures and interfacing with the UNIX operating system. While focusing on the fundamental data structures necessary for designing and implementing systems applications, we provide an introduction to the Unix Application Programming Interface (API), signals, threads, and inter-process communication (IPC). This course is instructed from a programmatic perspective using the C programming language, with special topics in both Java and Python.

The objective of this course is to focus on the principles and pragmatic methods for designing and implementing solutions to problems in Computer Science. The course will be instructed in the C programming language and the techniques discussed will be relevant for low-level or systems programming tasks. The data structures and algorithmic design aspects of the course are relevant across many different fields and are language independent. We will, however, implement several of these in C.

This course will prepare you directly for CS571 (Operating Systems) and CS555 (Networking), as well as for CS540 (Language Processors), CS550 (Database Systems), CS551 (Computer Graphics), and CS580 (AI).

Topics for this semester will include.

- C Programming Language
  - Foundations, Control Flow, and Functions
  - Memory and Pointers, Address Arithmetic, Structures, Bitwise Operators
  - Text and Binary File I/O
  - Header Files
  - Static and Dynamic Memory Allocation
  - Unix API Calls
- Analysis of Algorithm Complexity
- Stacks, Queues, and Linked Lists
- Hashing, Trees, and Graphs
- Heaps and Priority Queues
- Unix Processes, Signals, and Exceptions
- Multithreading and Inter-Process Communication
- Python and Java Overviews
Textbooks

Required:


Recommended:


Supplemental Texts: (not required, but helpful C references)


Grading:

- Homework: 40 points
- Midterm Exam: 30 points
- Final Exam: 30 points

Grades will be assessed on the following scale:

<table>
<thead>
<tr>
<th>Grade</th>
<th>Cut-off</th>
<th>Grade</th>
<th>Cut-off</th>
<th>Grade</th>
<th>Cut-off</th>
<th>Grade</th>
<th>Cut-off</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>90%</td>
<td>B</td>
<td>80%</td>
<td>C</td>
<td>70%</td>
<td>F</td>
<td>0%</td>
</tr>
</tbody>
</table>

As a Graduate-level Course, a satisfactory grade is a B or higher.
A passing grade is C or higher, however, a C is an unsatisfactory grade.
# Course Schedule

(Subject to adjustment)

<table>
<thead>
<tr>
<th>Week</th>
<th>Topics include</th>
<th>Reading</th>
<th>Assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Greetings, Course overview, C Programming Language overview: Variables, Expressions, Operators, Control Flow</td>
<td>Lecture Notes TCPL: Chapters 1, 2, 3</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Header files, Functions, Pointers, Arrays &amp; Strings, Typedef, Unions, Structures, C standard library</td>
<td>Lecture Notes TCPL: Chapters 4, 5, 6</td>
<td>Homework 1 assigned</td>
</tr>
<tr>
<td>3</td>
<td>Pointers cont, Memory Management, Standard I/O Library, Buffering, Strings, Math, Utilities</td>
<td>Lecture Notes TCPL: Chapters 7, 8 APUE Chapter 5</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Address Arithmetic, Linked Lists: Simple, Doubly, and Circular</td>
<td>Lecture Notes</td>
<td>Homework 1 due</td>
</tr>
<tr>
<td>5</td>
<td>Linked lists continued, Recursion, Stacks and Queues</td>
<td>Lecture Notes</td>
<td>Homework 2 assigned</td>
</tr>
<tr>
<td>6</td>
<td>Trees and Graphs</td>
<td>Lecture Notes</td>
<td></td>
</tr>
</tbody>
</table>
| 7    | Homework 3 discussion  
**MidTerm Exam - Online**  
*Currently scheduled for 3/11* | Lecture Notes | Homework 2 due Homework 3 assigned |
<p>| 8    | Hashing, Heaps, and Priority Queues | Lecture Notes |  |
| 9    | Bitwise Operators, Files, Directories, Byte Ordering (Big/Little) Endian | Lecture Notes APUE: Chapter 4 |  |</p>
<table>
<thead>
<tr>
<th>Week 10</th>
<th>Process Control, Daemon Processes, Signals</th>
<th>Lecture Notes APUE Chapters 8,10,13</th>
<th>Homework 3 due Homework 4 assigned</th>
</tr>
</thead>
<tbody>
<tr>
<td>Week 11</td>
<td>Interprocess Communication</td>
<td>A Lecture Notes APUE Chapter 15</td>
<td></td>
</tr>
<tr>
<td>Week 12</td>
<td>Advanced IPC, POSIX Threads</td>
<td>Lecture Notes APUE Chapter: 11,12,17</td>
<td></td>
</tr>
<tr>
<td>Week 13</td>
<td>POSIX Threads (continued) and Conditional Variables,</td>
<td>Lecture Notes APUE Chapters: 11,12</td>
<td>Homework 4 due</td>
</tr>
<tr>
<td>Week 14</td>
<td>Socket Programming in C, Java, and Python (Language Comparison and Contrast)</td>
<td>Lecture Notes APUE Chapters:16,17</td>
<td></td>
</tr>
<tr>
<td>May 6</td>
<td>Final Exam – Cumulative Online</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

CLASS COMMUNICATIONS

CS 531 will be using the Blackboard system for most class communications. You are responsible for any notifications or information posted on Blackboard, and you will need to check Blackboard regularly for such notices. Some information may be disseminated through Blackboard rather than in class. Individual communications with the professor may be done by email using your GMU email account.

*When you email, please be sure to include your name, the class number and the topic in the subject header.* (E.g.: Subject: Sam Jones / CS 531 / assignment 2)

PROGRAMMING POLICIES

(1) **No sharing or discussion of code for assignments.** Unless specifically stated otherwise, all assignments are individual projects, not group projects. Students are expected to do their own work, not to share programs with each other, nor copy programs from anyone else. Any discussion or sharing of code outside these guidelines constitutes an honor code violation. Suspected honor code violations are taken very seriously, and will be reported to the Honor Committee.

(See https://oai.gmu.edu/mason-honor-code/)
(2) **No incorporation of code from any source external to the course.** You may **not** incorporate code written by others. Of course, you may freely use any code provided as part of the project specifications, and you need not credit the source. Working something out together with the instructor usually will not require crediting the source.

(3) **Back up your program regularly.** You are expected to backup your program in separate files as you get different pieces working. Failure to do this may result in your getting a much lower grade on a program if last minute problems occur. (Accidently deleting your program, having problems connecting, etc., will **not** be accepted as excuses.)

(4) **Keep an untouched copy of your final code submission.** It is important that you not touch your programs once you have made your final submission. If there are any submission problems, consideration for credit will only be given if it can be verified that the programs were not changed after being submitted.

(5) **Code must compile with Mason gcc.** Students may develop programs using any computer system they have available. Please note, however, that submitted projects must run under a C compiler available on Mason. Your documentation should clearly state which software was used for compilation, and once makefiles are introduced, a makefile should be included with each assignment submission. **No extensions** will be given due to compiler incompatibilities.