CSE320 System Fundamentals II

Syllabus

Term: Spring 2022
Instructor: *Tony Mione*
Course Meeting Times: *Tue & Thurs, 10:30-11:50 AM*
Office: *B425*
Phone: *+82 032-626-1226*
Email: *antonino.mione@sunykorea.ac.kr* ***Office***
Office Hours:
Mon-Thur: 1:00-2:00PM
(or by appointment) [B425]
Course Homepage:
www3.cs.stonybrook.edu/~amione/CSE320_Course/index.html
Text: Bryant, Randall E., O'Hallaran, David R., *Computer Systems: A Programmer's Perspective (3rd Edition)*, Pearson, 2016, ISBN: 978-0134092669
Recommended Reference: Kernighan, Brian W., Ritchie, Dennis M., *The C Programming Language Second Edition,* Prentice - Hall, 1988, ISBN10: 013113709
McDowell, Charlie, *C for Java Programmers: A Primer*

Course Overview

In the course, we will learn the run-time environments with C programs, essential concepts of operating systems with system calls, concurrent programming, and performance analysis with the focus on several cross-cutting issues like memory management, error handling, and multi-threading.

Major Topics Covered in the Course

- C Programming
- Memory Hierarchy, Caches, Virtual Memory
- Operating System Processes and POSIX abstractions
- Operating System Scheduling and I/O
- Basic Networking and Socket Programming
- Multi-threading and Parallel Programming

Course Objectives/Outcomes

Upon completion of the courses, students are expected to:

- Develop an understanding of the layers of software that lie between an application program and the underlying hardware and how they interoperate.
- Develop an ability to program with operating system APIs.
- Develop an ability to write and analyze multi-threaded programs.

Prerequisite

- C or higher: CSE 220 and CSE major

Grades and Evaluation

The course provides a total of 500 points distributed across the below categories. Your grade in the course will be based on the following work:

Assignments– *45% (225 points)* - Assignments [about 8 or 9] will be given that will involve using concepts learned in class related to Unix APIs, memory management, and other OS related features. **Class Attendance/Participation – 5%** *(25 points)*
Midterm Exam 1 – 15% *(75 points)* - A midterm exam based on reading and concepts presented in the lecture.
Midterm Exam 2 – 15% *(75 points)* - A midterm exam based on reading and concepts presented in the lecture.
Final Exam – 20*% (100 points)* - A cumulative final exam will provide questions that will cover the key concepts taught through the entire semester.

Final Grade Calculation

The final grade is based on the accumulated points from all quizzes, exams, and assignments (with the entire class comprised of 500 points). Letter grades are given on the following scale:

Letter	**Minimum Percentage**	**Minimum 'points'**
A	93	465
A-	90	450
B+	87	435
B	83	415
B-	80	400
C+	77	385
C	73	365
C-	70	350
D+	67	335
D	60	300
F	<60	<300

Attendance

The range of topics covered in this course is extensive, and due to the limited lecture, these topics are covered in an intensive manner. Therefore, attendance at all lectures is mandatory in order to keep up and perform well.

- Attendance will be taken in each lecture class.
- ***If a student has over 20% unexcused absences, the final course grade will be an F.***

Re-grading

For re-grading of an assignment or exam, please meet with the person (instructor or teaching assistant) responsible for the grading. All such requests that are later than one week from the date the graded work is returned to the class will not be entertained. To promote consistency of grading, questions and concerns about grading should be addressed *first to the TA* [if graded by a TA] and then, if that does not resolve the issue, to the instructor. You are welcome to contact the TA by email or come to his/her office hour. If you would like to speak with the TA in person, and have a schedule conflict with his/her office hour, you are welcome to make an appointment to meet the TA at another time.

Programming Assignments

Extensions

Programming assignments must be turned in on the day they are due. Students are urged to plan ahead to avoid problems such as congestion or failure of computer facilities at the last minute. If your assignment is incomplete or is not working by the due date, turn in whatever you have. Note due to limited resources for grading, programs which do not compile or run for testing may not be graded. If some sort of emergency prevents you from submitting your assignment on time, supplying me with suitable documentation and notification prior to the assignment deadline will be considered.

Course Schedule

Following is a tentative schedule for the class topics:

Week/Day	Lecture Topics	Readings	Tests/Vids
W1: 2/22	Course Overview		
2/24	Basic Unix Commands	Ch 1	C references
W2: 3/1	Independence Movement Day [No classes]		
3/3	More Basic Unix Commands		
W3: 3/8	C programming I		
3/10	C programming II		
W4: 3/15	Advanced C Programming / Using gdb		
3/17	Assembler programming		
W5: 3/22	Dynamic Memory Allocation I	Ch 9.9-9.11	
3/24	Dynamic Memory Allocation II		
W6: 3/29	Exceptional Flow Control	Ch 8.1-8.4	
3/31	Midterm I		Midterm I
W7: 4/5	Signals	Ch 8.5	
4/7	System APIs	Ch 10	
W8: 4/12	Network: Architecture	Ch 11.1-11.3	
4/14	Network: Socket API	Ch 11.4-11.6	
W9: 4/19	Concurrent Programming: Threads	Ch 12.1-12.4	
4/21	Concurrent programming: Semaphores I	Ch 12.5	
W10: 4/26	Concurrent programming: Semaphores II, shared resources, deadlocks	Ch 12.6-12.8	
4/28	Memory Hierarchy		
W11: 5/3	Cache	Ch 6	
5/5	Children’s Day: No classes		
W12: 5/10	Review: Midterm II		
5/12	Midterm II		Midterm II
W13: 5/17	Virtual Memory I	Ch 9.1-9.5	
5/19	Virtual Memory II	Ch 9.6-9.8	
W14: 5/24	Memory Mapping and Linking I	Ch 7	
5/26	Memory Mapping and Linking II		
W15: 5/31	Performance		
6/2	Review for Final		

Academic Dishonesty

You may *discuss* the assignments at a 'high level' with anyone you like, however each student, when they submit an assignment, asserts that the assignment is **their own work, and only their own work. Any evidence that source code or solutions have been copied, shared, or transmitted *in any way* (this includes using source code downloaded from the Internet or written by others in previous semesters!) will be regarded as evidence of academic dishonesty.**

Guidelines for Assignments

Working together to find a good approach for solving a programming problem is cooperation; listening while someone dictates a solution is cheating. You must limit collaboration to a *high-level discussion of solution strategies*, and stop short of actually writing down a group answer. Anything that you hand in, whether it is a written problem or a computer program, must be written in your own words. If you base your solution on any other written solution, ***you are cheating***

Guidelines for Taking Exams

When taking an exam, you must work completely independently of everyone else. Any collaboration here, of course, is cheating. All examinations will be closed-notes and closed-book. No electronic devices of any kind will be permitted to be used during exams. All cell phones must be silenced or turned off during exams. You will be allowed one sheet of notes, both sides (8.5 x 11 or A4). [This policy may vary slightly for online exams given due to COVID-19 conditions].

General Guidelines

Be advised that any evidence of academic dishonesty will be treated with utmost seriousness. *We do not distinguish between cheaters who copy others' work and cheaters who allow their work to be copied.*

If you cheat, you will be given an F on the assignment. Any incidence of cheating will be reported to Academic Affairs. If you have any questions about what constitutes cheating, please ask.

Students with Disabilities

If you have a physical, psychological, medical or learning disability that may impact your course work, please let the instructor know. Reasonable accommodation will be provided if necessary and appropriate. All information and documentation are confidential.

Critical Incident Management

The University expects students to respect the rights, privileges, and property of other people. Faculty are required to report to the Office of Judicial Affairs any disruptive behavior that interrupts their ability to teach, compromises the safety of the learning environment, or inhibits students' ability to learn.