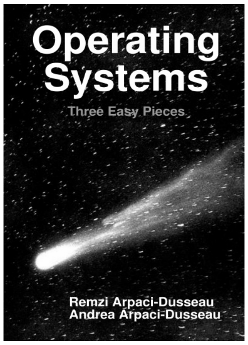

CMPE320

Operating Systems

Section 01: MW, 11a - 1:50p @ MCT158

Section 02: MW, 2p - 3:50p @ MCT158

Dr. Chen Huo

chuo@engr.ship.edu
Office Location: MCT150
Office Hours: `https://web.engr.ship.edu/~chuo/`
Zoom meetings by appointments

The content of this schedule may change during the semester.

Course Description:

A study of operating systems concepts and interfaces, with a special emphasis throughout the course on the concept of abstraction and separating mechanism from policy as a design technique. Topics include UNIX shells and common commands, writing shell scripts, important system calls, performance benchmarking, OS impact on program design and performance, processes, multiprogramming, multiprocessing, threading, scheduling, process isolation, inter-process communication, mutual exclusion, deadlock detection and avoidance, file system design, permissions and protections, and RAID. At the end of the course, students will be able to describe the importance of abstraction as a design pattern, and use it to explain the organization of OS components, interact with the UNIX shell and write shell scripts, and write programs using important system interfaces, understand the performance impact of making system calls, and independently find sources to guide their future development.

Prerequisite(s): C or better in CMPE220

Credit Hours: 4

Text(s):

1. *Operating Systems: Three Easy Pieces*[1], ISBN-13: 978-1985086593, Required

Websites and Tools:

1. D2L: Course materials, grades
2. Repl: Exercises, Playground

[1] It's a free book. For details, see in `https://pages.cs.wisc.edu/~remzi/OSTEP/`

3. Piazza: Discussions
4. GradeScope: Grading Projects

Grade Distribution

Quizzes	20%
Labs	15%
Projects	25%
Reading Papers	10%
Midterm	15%
Final Exam	15%
Extra Credits	0 – 5%

Letter Grade Distribution

>= 93.00	A	70.00 - 76.99	C
90.00 - 92.99	A-	60.00 - 69.99	D
87.00 - 89.99	B+	<= 59.99	F
83.00 - 86.99	B		
80.00 - 82.99	B-		
77.00 - 79.99	C+		

Coding Projects

There are two coding projects, *Process Shell* and *Wzip*. Both come from the book project repository. I modified them to make 5 stages, with each counts for 5% of the course. For each stage, you will submit your code on Gradescope. Each submission will be graded automatically against the book author's tests. There will also be manual code review, a small number of my tests, as well as *valgrind* inspections for some stages. See more details in lab 2.

Stage No.	Name	Note
1	Shell - stage 1	Fork, built-in commands
2	Shell - stage 2	Redirection, etc.
3	Shell - stage 3	Parallel commands, etc.
4	Wzip - single	Simple compression algorithm
5	Wzip - threads	Using threads

Labs

There are 5 labs. Each worths 3% of the course grade. An entire class time will be allocated for a lab. You are required to complete a portion of the lab in-class to obtain 1% "in-class" grade (out of the 3%). If you cannot complete the lab in-class, you will have a week to complete it. You'll need to get it checked during my office hours or stopping by (Zoom is also fine). You can submit a lab by handing in the hardcopy. In some cases, you can also "submit" it after checking via Zoom. You can submit an incomplete lab by the deadline to get partial credits.

Lab No.	Note
1	Reviewing C basics
2	Coding project how-to
3	Fork on an echo server
4	Using threads on a web server
5	Using a thread pool

Quizzes

There will be weekly D2L online quizzes after Wednesday lectures. They are due on Sunday nights. You are given 5 to 9 attempts for each quiz. The quiz is individual and is open-book. For each attempt, you will see the score but you will not see which ones are correct and which ones are not. Late quiz submissions are not accepted because the answers will be automatically released after the due date. The lowest quiz score will be dropped.

Exams

There will be one mid-term exam and one final exam. Both are on D2L[2] and are closed-book. (There will be an official information sheet for the final.) The final is not cumulative. The exams are based on the quiz questions — some of them are the same, some with numerical modifications if calculations are involved. In addition, the exams will ask for short C programs which are related to the projects.

Grace Period Policy

Everyone has 5 days of grace period for any lab or project stage. The days will be automatically applied for late submissions. When there is not enough days left to cover the lateness of a submission (at the moment of the submission), the submission will receive a zero. Everyone can earn up to 5 extra days by: 1) asking questions about labs or projects during my office hours face-to-face, one day for each lab/project-stage, 2) completing 3 stars of Hackerrank C challenges (`https://www.hackerrank.com/domains/c`) between Jan 1 and Jan 27 5p (3-day award). I'll keep the log of grace-period days private. You can ask me about your own days at anytime.

Course Policies

- **General**
  - Slides will be released as notes on D2L after lectures.
  - You are expected to take notes. Some contents in the slides and written on the board are not from the textbook.
  - We will use Piazza for discussion. You cannot post any *code* related to homework or labs as public posts.
  - Quiz questions will be given to you prior to the quizzes. You will do the quies in class, close-book.
  - Most materials of the course will be managed on D2L.
- **Grades**

[2] In rare cases, if I cannot get computer rooms for the final, I may turn the final to paper-based.

  - Grades in the **C** range represent performance that **meets expectations**; Grades in the **B** range represent performance that is **substantially better** than the expectations; Grades in the **A** range represent work that is **excellent**.
  - Grades will be maintained on D2L. Students are responsible for tracking their progress by referring to the online gradebook.
  - You can dispute your grade for a certain item within **1** week after the grade is released. For example, you cannot dispute your grade of the first homework at the end of the semester.

- **Attendance and Absences**
  - Students must provide valid reasons for excused absences.
  - Students are responsible for all missed work, regardless of the reason for absence. It is also the absentee's responsibility to get all missing notes or materials.

Title IX

https://www.ship.edu/academics/colleges/engineering/title-ix/

Academic Honesty Policy Summary

Those students found violating Shippensburg's academic dishonesty policy [3] will be dealt with on a case by case basis. Minimal punishment should it be a first offense is a zero for the assignment and signing a form admitting to the offense [4]. Second offenses are handled directly by the office for the Dean of Students. See additional information about academic Dishonesty within the School of Engineering[5].

[3] https://www.ship.edu/life/dean-students/student-conduct/academic_policies/

[4] https://www.ship.edu/globalassets/provost/academic-dishonesty-form.pdf

[5] https://web.engr.ship.edu/media/2012_AcademicDishonestyPolicy.pdf

Tentative Course Outline

The weekly coverage might change as it depends on the progress of the class. The students will be advised when to read the corresponding material.

Week	**In Class**	**Notes**
Week 1	MLK Day: no class	
	Course Intro	Read Ch2
Week 2	Process	Read Ch4
	Lab 1	
Week 3	Process API	Read Ch5
	Direct Execution	Read Ch6
Week 4	**Lab 2**	
	CPU Scheduling Intro	Read Ch7
Week 5	MLFQ and Lottery Scheduling	Read Ch8 and Ch9
	(spare)	
Week 6	Address Space	Read Ch13, 14, 15
		Project 1 stage 1 due
	Segmentation and Paging	Read Ch16, 18
Week 7	TLB and Smaller Page Tables	Read Ch19, 20
	Swapping	Read Ch21, 22
Week 8	Spring Break, no class	
Week 9	**Lab 3**	Project 1 stage 2 due
	Midterm Exam	
Week 10	Threads	Read Ch26, 27
	Locks	Read Ch28
Week 11	(spare)	
	Lab 4	
		Project 1 stage 3 due
Week 12	Condition Variable	Read Ch30
	Semaphore	Read 31
Week 13	**Lab 5**	
	Common Concurrency Problems	Read Ch32
		Project 2 stage 1 due
Week 14	I/O Devices	Read Ch36
	Hard Disk Drives	Read Ch37
Week 15	RAIDs	Read Ch38
	FFS	Read Ch40
		Project 2 stage 2 due
Final	**Final Exam**	