CMSC 421: Principles of Operating Systems

UMBC 2018 Spring

Standard Wise Saying: "One of the major differences between school and real life is that school first teaches you a lesson and then gives you a test; life gives you a test that teaches you a lesson" Chris Hadnagy

"Bring me a puddle, not an ocean". I don't know who originally said this.

Instructor

Gerald S. Tompkins

Final Exam:

https://registrar.umbc.edu/files/2018/01/Regular-Final-Exam-Schedule-Spring-2018.pdf

Course Information

Course Title: CMSC 421: Principles of Operating Systems
Time & Place: Monday and Wednesday, 1730-1845
Classroom: Sherman 015
Office Hours: I will be in my office or around our classroom one hour before class. My office is ITE 328. You are welcome to stop by. If you need to meet me at another time, let me know and we can make arrangements.

I can be reached at my email address, Gerald@umbc.edu
Teaching assistant: Ethan Goldschen
Email address: egold2@umbc.edu
TAs office hours and location to be announced.

First question on any exam in this class

1 == 1
No, I am not kidding. I tell each class this and at least seven people have missed it on the midterm. Others have missed it on the final. Names must be spelled correctly. Attention to detail is important.

1. Instructors name
2. TA's name
3. Instructors email address
4. TA's email address

Prerequisites

This class has the following prerequisites:

- CMSC 341, and
- One of the following:
  - CMSC 313, or
  - CMPE 212 and CMPE 310

Students must be a programmer and consider themselves a computer scientist to be successful in this course. Students must already be familiar with the C computer language. Homework and projects will be in C. You must be comfortable writing C programs. If you do not meet these prerequisites, drop the course.

Notices:

Students should read their UMBC email before coming to class. This is the way I will notify you if I am unable to make a class. Repeat: before coming to each class, check your mail.

Textbook

The main textbook for this course is *Operating System Concepts, Ninth Edition* by Silberschatz, Galvin, and Gagne, colloquially known as the "Dinosaur Book". It is also in its ninth version which indicates the book is a classic. It is possible to purchase an electronic copy of this book and it can be read on a tablet. I used this format instead of a physical book when developing the course.

The following books are also recommended. The last two are podcasts that I listen to regularly. They are all excellent references and may prove useful in your future endeavors:

- *"Advanced Programming in the Unix Environment"* by Richard Stevens - a very hefty book, good for generic Unix/POSIX programming
- *"The Linux Programming Interface"* by Michael Kerrisk - similar to the above APUE, but tailored to Linux programming
- *"Linux Kernel Development"* by Robert Love - get the latest version available, though be warned that the third edition is somewhat outdated.
- *"Security Now":* Podcast by Steve Gibson. https://twit.tv/shows/security-now
- "*Paul's Security Weekly*": Podcast by Paul Asadoorian

Course Description

An introduction to the fundamentals of operating systems. Formal principles are illustrated with examples and case studies of one or more contemporary operating systems.

This course requires several significant programming projects that teach how operating systems work by allowing students to write portions of a simple operating system. This is kernel code which is not the same as user code. Different function calls are used. Thus, students should ensure that they've had a strong programming background and are very comfortable with C and not just Java and C++ before taking this class. Taking CMSC 421 at the same time as CMSC 341 is not recommended.

The most difficult part of this class are the projects. Writing kernel code is non-trivial start early.

Teaching Method

I plan to primarily teach from the Silberschatz book. The slides I use will be taken from: http://os-book.com. I will mail each student a copy of the slides I use before each class and they will be posted on Blackboard. The slides I use from the website will be slightly different and I will deviate from them in class. I have been a computer scientist for over 32 years, and I have some stories that apply to the slides.

There is debate on whether to teach computer security as a separate topic in your senior year, or have it woven throughout your four years as an undergraduate Computer Science major. With the seeming blizzard of break-ins (Target, Home Depot, Anthem, OPM) and my background, I think both are important. This class will have a security component. The projects will have also security component.

Tentative Topics to be covered this semester

- Operating-System Structures
- Processes
- Threads
- Caching
- Process Synchronization
- CPU Scheduling
- Deadlocks
- Memory Management, Main and Virtual
- Virtualization

- Mass Storage
- File System Interface
- File System Implementation
- I/O Systems
- Security
- Protection
- Types of OS (Linux and Windows)

Grading

Ethan Goldschen will be the teaching assistant this semester. He will be responsible for developing and grading all homeworks and the projects. The instructor will grade the midterm and exams.

If you have any questions about the homeworks or projects, please contact the teaching assistant first. If he cannot resolve it, please contact me.

Your final grade will be computed from the following components:

Homework (~4)	20
Projects (~3)	30
Midterm	20
Final Exam	30
Total =	100

Your final letter grades will be based on the standard formula: $0 \le F < 60$, $60 \le D < 70$, $70 \le C < 80$, $80 \le B < 90$, $90 \le A \le 100$.

Grades are given for work done *during* the semester; incomplete grades will only be given for medical illness or other such dire circumstances.

If there is a curve in the grade, attendance, class participation, and passing grade on all work is required. The instructor appreciates students who attend class and are not constantly monitoring their mobile devices for updated Facebook, Twitter, and other social networking applications during class.

For best results in this class, please store all mobile devices during class.

Attendance

You are expected to attend all classes. If you miss a class, you are responsible for getting the notes and any verbal information given during class from a fellow classmate. Just like in life, make friends and network. I do not believe you can miss many class periods and be successful in this class or life. Part of your job is showing up.

As an encouragement to attend class, I will be stressing certain topics I consider to be more important than others. These tend to be on the tests and midterms.

Also, we will cover the homework and projects in class.

Homework and Projects

Programs are graded not just on correctness (producing the correct output) — but neatness counts. Also, you must be consistent. If you start off indenting 3 spaces and switch to 5 spaces, you will be penalized. It is much easier to originate a program than picking up another person’s code and attempting to modify it. You must make it easy for another person to understand your code. Here *neatness* means that your program is well formatted (see Coding Standards and Indentation Standards), the output from your program is nicely presented, and that the logic in your program is straightforward. Sloppy code *will* be penalized.

If you cannot complete a programming assignment, you should still submit your code. Explain what is missing and what is remaining. Partial credit will be given for reasonable effort. Late work will not be accepted without a very good reason.

You will be submitting your programs electronically to BLACKBOARD or something else determined in class. Assignments are normally due on midnight of the assigned day, not several minutes afterwards. The deadline is *strictly* enforced by the BLACKBOARD.

Once per semester, you may have a late submission, albeit with a 20% late penalty. Additional late submissions will not be graded and will be ignored.

Be aware that the GL system may go down from time to time. You are given ample time to complete your programs, so system downtimes are not necessarily an excuse for late submission.

Exams

There are two exams scheduled for this class. We will determine the mid exams dates when we get closer. The final is the UMBC scheduled exam day. If I parse the spread sheet correctly, it will be Monday, 18 December at 800-1000. Unless you notify the instructor and there is a documented Letter of Accommodation, students are not allowed to use any electronic device during the testing period. Testing is to be the work of each student individually. No collaboration.

Collaboration

Professor Lawrence Sebald and I are working together with tests and projects this semester. We will have similar mid terms, homework, projects and final exam. This means you can also use Professor Sebald's TA and ask him questions.

Notes:

1. Resume
2. Hackathon
3. How to get a job

Academic Integrity

When you submit your homework, project, and classwork, you are stating that the work was created by your own individual effort, or in the case of a group assignment, created solely by the effort of members of your group.

Receiving help from the instructor or from the teaching assistant does not violate this academic integrity policy.

You may also receive help from other sources. However, this help must be limited to:

- Discussions about the meaning of the assignment.
- Identifying syntax errors in your program.
- Identifying simple logic errors in your program.

The following is a non-exhaustive list of actions that clearly violate this academic integrity policy:

- Someone else is typing code in your program.
- You are cutting and pasting more than a single line of code (from a program that was not distributed by the instructor).
- You are looking at someone else's program while you are typing in your code.
- You receive someone else's program by email, hard copy, text message, instant message, ...
- You make your program available to another student in CMSC 421 directly or indirectly by email, hard copy, text message, instant message, ...

This policy recognizes that students can learn productively from many sources including from other students in the class. Thus, this policy allows small amounts of help but prohibits outright copying. Although, this leaves a gray area between "small amounts of help" and "outright copying", it is better that we live with some ambiguity than to have a clear-cut policy that deprives the students of productive learning opportunities. Students who have doubts about the propriety of an activity should consult the instructor.

Students who violate this academic integrity policy will receive a grade of 0 for that assignment. A second violation will also result in a reduction of one full letter grade in the student's final course grade. In the case where one student copies the program of another student, both students are considered to have violated this policy. Here, copying includes not just programs that are verbatim copies, but also programs that are substantially similar and could not have been produced independently.

Furthermore, all parties concerned will have their prior homework and programs checked.

Violations of this policy may be reported to the University's Academic Conduct Committee for further action. Egregious cases of cheating will be written up as a **more serious** infraction. In this case, you will not be allowed to drop the course. Also, a **more serious** infraction would appear as a permanent part of your student record and would be seen by potential employers when they ask for an official copy of your transcript.

For a more complete description of academic dishonesty, refer to the UMBC Undergraduate Student Academic Conduct Policy.

Email

In order to facilitate email communication, please observe the following guidelines for email sent to the instructor.

- Make sure that the subject line of the email message clearly identifies its content (e.g., mention CMSC 421).

- Use your UMBC email account. (I really shouldn't discuss your grade with some random person on the internet just because he has an email address that resembles your name.)
- Use your full real name.
- Submit your program instead of attaching it to your message.
- My email address is Gerald@umbc.edu

Class Hardware and Software Requirements

For this class, I **strongly** recommend using your own computer if at all possible. Either Microsoft, Apple, or Linux is fine. You will need either VMWare or Virtual Box. Virtual Box is free and has more features than the free version of VMWare. Either will be fine. You will need to know Unix/Linux and especially the command line. The class will focus on Linux and not Microsoft. This is because we have the source code for Linux, we can modify it, and run a new Kernel. It is also free!

You must know how to program in C. This cannot be the first time you have written a C program. We have four homework assignments plus two longer projects, all of which are to be written in C.

This class is using C because it is as close to the silicon as we can get without using assembler language. When other languages especially OO languages are used, you start to lose track of what code is generated. I do not want this to happen.

Class Lab

The ITE 240 Lab may be used by students enrolled in CMSC 421 for their project work. Students that are interested in using the lab should visit the Computer Science Department (ITE 325) to request access. The lab hours are Mon-Fri, 9:00am-9:00pm.

You will be able to access the lab using your UMBC student ID swipe cards. You must observe all the rules below, regarding usage of the lab:

- Observe all the University and Departmental policies regarding the use of the University's and the Department's computer and laboratory facilities.
- No Food or Drinks Allowed in the Lab. Absolutely! Never!
- Do not log in to multiple machines, especially when there are students waiting for access to the lab.
- Be respectful of others working in the lab. DO NOT create disturbances (e.g. don't be noisy, don't play audible music, etc).
- Use the laboratory ONLY for CMSC 421 related activities, NOT for general other coursework. There are designated OIT Labs across campus to serve this purpose.
- DO NOT ABUSE the superuser privileges from your installed kernel.
- Observe the operating hours of the lab. You must leave the lab by its closing time - no excuses, no exceptions!
- Report offenders of lab rules to the instructional staff.
- Report any suspicious activity to the instructional staff or the Campus police, as appropriate.
- The lab is provided to you by the University for your benefit. Take good care of it! Do not abuse it!
- Repeated offenses might result in shutting down the laboratory for everybody.

Failure to observe all the lab rules will result in suspending your lab access, as well as further disciplinary actions as determined by Departmental and University policy.