ECE357 Computer Operating Systems Syllabus -- Fall 2019 -- Prof. Jeff Hakner

Overview

This course in Computer Operating Systems is a required 3-credit class for all ECE majors taking the Computer Engineering track who were admitted Fall 2016 or later, and an elective for those admitted earlier, or for those from a different major or track, who are welcome as long as they have the prerequisite course work or equivalent knowledge and skills. The course provides a deep technical understanding of how modern computer operating systems function.

Classroom rules

No "devices" are to be used by students or in view during class. The only thing on your desk should be paper and a writing implement. This rule may seem stringent -- after all this is a technology class -- but it is based on years of classroom experience and numerous studies from across academia. The use of electronic devices in class creates a climate of distraction which not only detracts from your own learning but also impacts the learning environment of others. This ban includes "smart watches" (there is a clock in the room). Keep all these devices out of sight such as in your bag and turned off or set so they don’t make noise. Feel free to use your devices during breaks, before class begins and after class ends.

Likewise, I ask that you **refrain from side conversations** during class that are audible to others in the room, raise your hand prior to speaking, and generally follow rules of classroom etiquette. Since this is an evening class, I am very tolerant of food and non-alcoholic drink being consumed during class so long as it is not malodorous.

Attendance of classes is required by Cooper Union academic policies. I do not have a mechanism to enforce this. I do not take or record class attendance. You do not need my advance permission to be absent from class. However, please note carefully under "academic responsibilities" below! I believe that attending class adds value to your learning experience and I recommend that you attend all classes in their entirety.

Since our class is a 3-hour block, I elect to give one break, approximately at the mid-way point, not to exceed 20 minutes. Please return from break promptly. Avoiding delay in getting the class resumed after a break, or disturbance from students entering long after class has resumed, is a major factor in the success of the class.

Our class is the last of the day, and we are missing at least one regular meeting because of religious observance. Therefore I will be making up some of this time by ending class a few minutes beyond 9:00 PM, as needed.

Office Hours / Contact

I generally maintain office hours at 41CS, Rm 810 Mon/Tue/Wed 6PM until close. However, it is best to email me and make an appointment. `hak@cooper.edu` I will try to answer student email questions within 12 hours, including weekends. Lecture notes, problem sets and other material will be distributed via a course web site: `faculty.cooper.edu/hak/ece357`

Prerequisite Topics

The following academic topics and skills are assumed upon entry to the course and will be important for your success:

Computer Architecture course: the basic instruction cycle. register and memory operands. memory addressing. I/O devices. interrupts.

Programming Languages course: The C language! structs and unions. local vs global variables. memory allocation with malloc. pointers and arrays. strings.

Data Structures course: Linked lists, trees, hashing.

UNIX environment

Much of this course will be conducted with reference to a "UNIX" programming and operating systems environment. You will need to gain familiarity with using the operating system from the "command line" including compiling and executing programs with gcc. We will be studying the Linux kernel as a reference kernel. Many of the homework assignments will work just as well on other flavors of UNIX such as Mac OSX but the assignments will be written with the assumption that you are using some Linux distribution. (Note that using "CYGWIN" or the "Ubuntu Subsytem" under Windows will generally not be sufficient since it is not a true and full emulation of the UNIX systems programming environment and since the underlying kernel is not a UNIX kernel).

Since we aren't working with the GUI, your choice of "distro" is not that critical. One recommendation is to download a "LIVE CD" (really a DVD-ROM or a bootable USB drive) so you can run UNIX on your computer without installing it permanently. "KNOPPIX" is one such distribution which works reasonably well and has most of the common tools that you'd need. There are certainly other usable choices such as Debian or Ubuntu live media. All of these "live" systems will allow you to "mount" your existing hard drive to store files there if you wish, or you can keep files on a "persistent" partition within the USB drive. Of course, I recommend that you back up your work.

Academic Responsibilities

Students are academically responsible for studying material that appears in or is covered in any of the following: (This material is fair game for exam questions.)

> a) In-class lecture presentations and class discussions. If you are absent please get notes from a classmate. Not all material in the course will be presented in lecture! See below
>
> b) The "Lecture Notes" that are distributed at the conclusion of each course unit
>
> c) The homework assignments ("problem sets"). Exception: material derived solely from homework problems that are marked as "extra credit" is not going to be on any test.
>
> d) Any additional supplementary material (such as "man pages" documentation) that I might choose to distribute or ask you to obtain and indicate is required study material.

Homework Assignments

Associated with most of the syllabus units in this course are "problem sets" which generally involve one or more of the following:

i) writing code that runs under the UNIX environment and which illustrates or exercises kernel interfaces
ii) obtaining experimental results such as performance benchmarks
iii) exercises and problems/questions to be filled out.

The problem sets are designed to reinforce the material presented in lecture and lecture notes and are mandatory. They form a crucial part of the pedagogy of this course. It is very important that you do this work, and do so promptly.

Generally speaking, a problem set will be distributed at the conclusion of a lecture which concludes an academic unit, and will be due 2 weeks later. At each class meeting following distribution of a problem set, a period of time is set aside for discussing questions that students have which arise during the course of implementing the assignment. It is therefore very important that you begin work on a problem set within this initial 7 day period. If you wait until the last minute, you will not have the benefit of meaningful participation in the class discussion. In some cases, parts of a given problem set will be due at different times, especially the programming part vs. the short-answer part. Please pay close attention to the deadlines!

In some cases, problem sets will overlap and at other points, there may be no active problem sets. This will be dictated by class pace and schedule issues. If due dates change, you will be notified by email.

Please read each problem set description carefully! It contains many details on the requirements of the solution as well as hints to common implementation problems. Although some of the problem sets may be similar to ones used in previous instantiations of this course, do not assume that they are identical!

Submission: The submission of your problem set must be reduced to paper form! Although I will answer email questions and review code online, the final submission must be on paper. **Please read this carefully:**

* Code must be printed out in a fixed-width font no smaller than 10 pt. (Don’t cut-and-paste your code into MS-Word unless you are very careful. You want your indentation to line up consistently. This means a monospaced font.)

* If your code or sample output lines are long, make sure they "wrap" when printed out. Check the work that you submit! Make sure parts of your code are not cut off. Make sure that all of your code is included!

* When you are asked to attach sample output, the code you submit must coincide with the output you submit. Don’t make hand-written "corrections" to either your code or your output.

* Print your code in in black & white. (Remember, font changes and color are not part of the C language standard. If your code is unreadable when printed out this way, then it is unreadable, period.)

* Don’t take screenshots of text editor screens to submit your code. Such practice generally results in unreadable paper.

* If screenshots are requested, e.g. to demonstrate that your program ran and what output was obtained, please make sure the size of the printout is adequate such that the text which was on your screen can be read clearly, at a distance of 12 inches, without the use of an electron microscope!

* Any additional write-up which accompanies your code, such as discussion of experimental results and narrative, may be in any font or format of your choosing that is reasonable.

Homework Lateness & Review

Unless otherwise stated, the homework will be due at the start of class. This means when I start class, not when you arrive at the classroom. If the homework is late, I will assess a penalty of 1 point (equivalent to one letter grade). I may choose to not accept a late homework at all after a certain period of time has elapsed.

In some cases, I will randomly call upon students to either explain their homework solutions, and/or to come to the board to present their solutions. If you are called upon, you may consult your homework papers for reference.

Academic Honesty

The homework assignments are intended to be individual submissions of your own work. Group submissions are **not permitted.**

I understand that there are only so many ways to write certain programs, especially simple ones. However, **you** must actually write them! I also understand that asking questions among your peers or collaboration is a natural part of the learning process and I absolutely encourage that, as well as doing research on the internet (I caution you however that much of the information that you may find on the internet is WRONG) However, in the end, the program must be your own work. If you derived an approach or concept from another source, you are required to give proper attribution to that derivation as a comment within your source code. You are also required, if you obtained a snippet of code from another source, to **re-type** that code. Do not cut-and-paste it! Re-typing it forces you to stop and think about it. *You are permitted to re-use snippets of code from other sources, including other sources, with attribution.* **You are not permitted** to grab entire programs, or large portions thereof, even with attribution.

Submitting code that you didn’t write and attempting to pass it off as your own work (i.e. without proper attribution) is **plagiarism** and per Cooper Union’s academic honesty policies I am required to report such to the office of the Dean.

Although in-class exams are designed to thwart "cheating," to be clear, it is a violation of academic honesty rules to obtain or attempt to obtain, during the exam, the answer to any test question from any other person or any source other than explicitly permitted sources, or to disseminate your answers to others by any means or in any form during the exam.

Grading

NOTE: The deadline for dropping a class this semester is Wednesday, October 30, 2019. After that date, dropping will require the approval of the Dean’s office, who has advised that permission will only be granted for "exigent circumstances" and that "poor performance in a course does not rise to the standard of exigent circumstances." Therefore, I will endeavor to give you adequate feedback about your performance in this course prior to that date. If you have any questions or concerns about your class performance, I encourage you to have a conversation with me.

The numerical or objective basis for grade determination is:

45% homework average (normalized to a 100 scale)

20% midterm

35% final exam

Subjective factors such as quality of class participation, effort and improvement are also weighed.

I do not "curve" grades to a Gaussian distribution because the sample size is far too small but I will adjust my grading cutoff points subjectively based on class performance. Historically I have an "A" cutoff in the mid-to-high 80s, and a "B" cutoff mid-70s. Grades of C are not common and D and F are very rare.

Grades are given for each problem set on a 10-point basis. When I return an assignment, I will have a breakdown of things that you misunderstood or implemented incorrectly. This is not intended as an itemized tax return of how I arrived at your grade! Your homework grade is ultimately a *subjective* assessment of how I felt it demonstrated your understanding and mastery of the topic material involved, and the quality and effort on your part which I felt it demonstrated.

In some cases there will be extra-credit portions of an assignment. To receive full extra credit, you must do all the work that is specified as meeting the extra-credit criteria. The award of partial extra credit is entirely at my discretion.

Exams

There will be one mid-term exam designed to be covered in a maximum of one hour, and one comprehensive final exam which is designed for the full three-hour session. The midterm will be reviewed immediately following a class break. The final will be reviewed at a time and location TBD. The likely date for the midterm is Oct 16. The final exam will probably be given on Wednesday, December 18. This is subject to change.

All exams are open-notes and open-book. You may bring any printed material with you into the exam. All electronic devices are prohibited during the exam. This includes calculators, for which there will be no need in these exams. If you need to leave the exam room, e.g. for a bathroom break, you are not permitted to take your phone or other device with you.

Return of work

I endeavor to return all homework assignments by the class meeting in the week following the submission due date and all exams within one week. If required by ABET or Middle States, I may retain the original work and return a graded copy for a small sample of the class.

Textbook

There is no required textbook for this class. However, the following reference book is considered a "bible" of UNIX systems and network programming and may be helpful for at least some of the assignments:

Stevens, W. Richard & Rago, Stephen A. *Advanced Programming in the UNIX Environment*. 3rd ed. Addision-Wesley, 2013.

Course Outline

Below is the intended list of topics and the time allotted to each. This is subject to change depending on how the class progresses and scheduling issues.

Unit 1: Introduction to operating systems (1 wk)

Unit 2: Introduction to filesystem (2 wks)

Unit 3: Processes and I/O redirection (2 wks)

Midterm Exam

Unit 4: Signals and pipes (1.5 wk)

Unit 5: Virtual Memory (2.25 wks)

Unit 6: Concurrency and Synchronization (1 wk)

Unit 7: Introduction to the Kernel (1.25 wk)

Unit 8: Task switching in the Linux kernel, scheduling (1 wk)

Final Exam

Final Exam review & optional material such as networking, device drivers, virtualization.

Note: Class will not meet on Wednesday, October 9 in observance of Yom Kippur. Class will not meet on Wednesday, Nov 27 (Friday classes meet this day).