Syllabus: CIS 310 Operating Systems

Brian C. Ladd

Fall 2015

Introduction

Catalog Description

Principles of operating systems, concurrency, scheduling, virtual memory, device management, security and protection, deadlocks, introduction to networking. Prerequisite: CIS 203.

Overview

What is a computer? It can be viewed at different layers of abstraction. For example, when writing a Java program, the computer exposed by the language is a virtual computer that executes Java statements. Once the Java is compiled, there is a virtual computer running JVM bytecode. The layers keep going on and on, kind of like a parfait. At some point we reach the “naked iron”, the hardware realization of a computing device. The software that runs directly on the hardware is the operating system.

The operating system is the sergeant in charge of taking care of the higher-level software: it protects the hardware from stupid mistakes of the software and protects the software from stupid design mistakes in the hardware. It can also be described as a resource manager, software that manages what any other program can do with the hardware.

This semester’s textbook lists three aspects of an operating system: virtualization, concurrency, and persistence. You should expect those three pieces to guide our exploration of operating systems.

Learning Objectives and Outcomes

The Computer Science program of the CS has adopted ten learning objectives for students completing an undergraduate major or minor in the department. Each course in the curriculum is designed to address particular objectives so that the overall curriculum fulfills our goal of producing capable computer scientists. The objectives (and broad outcomes) serve as a rubric for evaluating the learning that happens in this class. You may wish to review this section of the syllabus when filling out the course evaluation and especially when studying for midterm and final exams.
Objectives

Students in the Computer Science program at SUNY Potsdam are expected to graduate with a:

1. knowledge of discrete and continuous mathematics – including elementary probability and statistics – and the ability to apply logic and mathematical proof techniques to computing problems.
   
   *Not applicable to CIS 310.*

2. knowledge of basic theory of computability and complexity of computation.
   
   *Not applicable to CIS 310.*

3. knowledge of and the ability to apply programming fundamentals in at least two programming languages.
   
   *Students will be able to write and compile an ARM assembly program for a RasPi. They will be familiar with writing a C program and compiling it to ARM assembly.*

4. knowledge of fundamental data structures and algorithms – including analysis of their correctness and complexity – related to various fields of computer science, and the ability to apply this knowledge to problems through the use of appropriate programming languages.
   
   *Not applicable to CIS 310.*

5. knowledge of computer architecture and organization, computer operating systems, and computer networks, and the ability to apply this knowledge to problems through the use of appropriate programming languages.
   
   *Students will be able to articulate the "three easy pieces" of a modern operating system as identified by our textbook. They will be able to list examples of the use of virtualization, concurrency, and persistence and explain how the example fits into a modern operating system.*

6. competence and effectiveness in technical oral, written, and visual communication, particularly as they apply to the dissemination of technical information on subjects dealing with computing technology and applications.
   
   *Not applicable to CIS 310.*

7. knowledge of and skill in applying good practices in software engineering.
   
   *Students use the git version control system to fetch source code for the course and to submit their programming assignments. Students can explain version control in database terms.*

8. the ability to function effectively in teams to accomplish a common goal.
   
   *Not applicable to CIS 310.*
9. an understanding of professional, ethical, legal, security, and social responsibilities and issues, including an awareness of impact of computing on individuals, organizations and society.

*Not applicable to CIS 310.*

10. a commitment to continuing professional development.

*Not applicable to CIS 310.*

**Grading**

This is a high 300-level course in one of your major fields. You will be evaluated on both your thinking about the course content and your ability to communicate your thinking. This means that your writing must be clear, concise, and in proper English (the language of the course). If you are not a native speaker of English, use friends who are or the Writing Center to make sure you can meet this requirement.

<table>
<thead>
<tr>
<th>Component</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Homework and In-Class</td>
<td>15%</td>
</tr>
<tr>
<td>Programming Projects</td>
<td>55%</td>
</tr>
<tr>
<td>Exams</td>
<td>15%</td>
</tr>
<tr>
<td>Final Exam</td>
<td>15%</td>
</tr>
</tbody>
</table>

**Homework:** Standard written homework drawn from the textbook and similar sources. A couple of the smaller programming assignments (at the beginning of the semester) will be graded as homework.

**Programming Projects** A collection of programming assignments, to be done individually. The assignments count for a high percentage of the grade because operating systems is best understood after implementing a program or two using various features.

**Exams:** There will be three midterm exams over the course of the semester. Each focuses on the section of the just covered.

**Final Exam:** There will be a cumulative final at the end of the semester.

**Textbook and Readings**

Arpaci-Dusseau and Arpaci-Dusseau, *Operating Systems: Three Easy Pieces* is our textbook. The chapters are short but there are a lot of them. Most reading assignments will be two or three chapters. In addition, reading from the Internet will be assigned, generally information on assembly language programming and the Raspberry Pi.

This semester we will practice some programming on the Raspberry Pi. Note that there are multiple versions of the RasPi platform out there: Pi 1 and Pi 2; Models A, B, and B+. Operating systems are intimately tied to the hardware on which they run; you need a Raspberry Pi 1, Model B+. The Pi 1 is simpler than the Pi 2 (single-core processor). The Model B+ has more memory than the A (512 MB), a different pinout, and uses a micro SD card as a boot device. You will need at least one micro SD card of 4GB or more.
General Rules

Your choices make your fate. Information arms you to make your best decisions and enjoy the best fate. This section describes the class expectations so that you can meet them (or decide not to).

Instructor Expectations of Students

Communication

Read/respond to e-mail. Read/respond to the course Moodle site. These are the two primary means of communication in the class. You should make use of them.

Moodle is the communications hub of this class. It is where readings, writings, programs, presentations, and homework deadlines appear. If you find any discrepancies or have questions about due dates or the meaning of assignments ask in person, in class, or in e-mail. My office hours are listed in my Moodle profile.

Students are expected to have a copy of the textbooks (if any) and are expected to complete reading assignments before the beginning of class the day they are due. In-class participation requires you to have engaged the reading.

Students are expected to listen actively in class. They are also expected to take notes in class. These two activities (which are linked) correlate strongly with understanding the material presented. Do not copy slides or board notes directly: digest them and restate them in your own words. Mark where you expect test questions to lurk (sometimes the instructor will even help with this).

Advanced courses often use a version control system (such as git) to retrieve and submit source code and programming assignments. This requires you to understand how to use git, to understand the difference between the class repository (which belongs to the instructor) and assignment repositories (which belong to you).

Attendance

American undergraduates are old enough to join the army, vote, and even get married. You are each mature enough to make your own decisions about attending class. Be aware that you decisions have consequences.

Students are expected to attend every class. Students are responsible for all material covered in every class meeting. There is no taking of attendance in this class (see 'old enough' in the previous paragraph).

The study of computer science is cumulative; past experience shows a strong correlation between high absences and low grades.

The registrar schedules the final exams every semester; I am not able to move them and do not give early finals. Take-home final exams are due before the end of the scheduled exam period (it will be accepted early).

Do Your Own Work

Do your own work. That should go without saying but it appears that for some students it bears repeating.
Code you plan to turn in for an individual grade should not be shared with any other student; you should never look at another student's project code, not even for 'guidance.'

I want you to discuss assignments and show off results of your playing with programming different things. Discussion at a high-level, even if it includes details on data structures is fine; debugging assistance is also acceptable though the depth of code analysis necessary for debugging skates on thinner ice; directing someone else on exactly what methods a given class should have or the exact format of a programmer-controlled file should take is beyond the pale.

This does not apply to group work: in a group project, all members of the team are expected to see and understand all parts of the project. Cross-disciplinary understanding is a highly valued skill in the real world and a major reason for group work in our curriculum.

Turning In Work

Start early, everything takes longer than you think.

As mentioned earlier, reading is to be done before class. Homework assignments are also due at the beginning of class so please complete them before class.

Assignments have a due time: a date and a time when they are due. If they are due electronically (they all are), make sure you submit them the expected way by the expected hour.

Make sure you check the spelling and arithmetic in your assignments. It does not help your grade (or my mood) to turn in sloppy work.

Attitude

Learning is not always easy. Learning is not always comfortable. Students must actively engage the material and be able to ask for help. This does not mean look at the problem for fifteen seconds and throw up your hands because it is 'hard.' It does mean beating your head against it for a bit and then asking me for guidance.

Be respectful. Trust me that I actually have a plan and I know where we are going (okay, I don't always know exactly where we are going at every moment but I do have a plan). The readings, the assignments, the lectures, they all go together.

Start early! Everything takes longer than you think.

Student Expectations of the Instructor

Communication

You may expect me to state and keep office hours (you know where my office is, right? 301 Dunn Hall) They are part of my Moodle profile.

You may expect fairly prompt answers to e-mail (next morning is typical).

You may expect confidentiality: I will not discuss your grades with anyone except my departmental colleagues (and then only to make sure that I am doing my job correctly). I will not share work you have done without your permission.
**Attendance**

You may expect me to be prepared. This means for class every day and it means with assignments, too. You should expect adequate time to complete an assignment after it is published. You may expect some flexibility (with in the limitations of getting through the course materials) in due dates.

**Assignments**

Expect assignments to be clear (correct spelling, correct grammar). Expect most sample code to compile and run. Expect assignments to be accompanied by a clear system by which they will be evaluated. Expect me to follow that system.

Students should expect that their work will be evaluated on its own contained merits and not based on how closely it follows the instructor's personal/political views (unless I put parroting my political views into the rubric).

**Student Expectations of One Another**

Wait, students have expectations of one another? Yes, they should. The most important expectation we should all have of each other is to create a safe learning environment. By safe I mean we should all feel safe to ask questions, to admit mistakes, to try and to be wrong.

Students should be willing to help one another and should all be aware of the limits.

In group work, all students in the group should expect equitable division of labor and should expect to evaluate every member of the group.

Students should expect to evaluate other students' work (and to be evaluated by other students). Both parts of this process have valuable learning outcomes.

Students should expect to feel safe being different. What does this mean in computer science? I actually am not sure. It means respecting each other, evaluating each other in openly communicated terms related to computer science (since that is the topic of the course), and being flexible, accommodating each other whenever possible.

**Grading Grievances**

When graded work is returned you may have questions about how it was graded. If the question pertains to process (e.g., addition, number of points per subsection), by all means ask me as soon as possible. Typically, if the grade is changed, you will be asked to e-mail me the grade update so that the Moodle grade book is current.

If, however, you feel that your grade fails to reflect the quality of your work (you want to argue that your answer is, in fact, correct, rather than incorrect), take a deep breath and wait. You must wait no less than one and no more than seven days after the work is returned to you to raise the issue with me.

After the cooling-off period, resubmit the work along with a written explanation of your concern. This is to assure that your concern gets a cool, thoughtful consideration. If I agree to regrade the work, I will regrade the entire project, replacing the old grade with the new one.