

CS180: Operating Systems I

Fall 2020 Course Syllabus

Contact Information

Instructor: Matthew Mead
Office Hours: By arrangement (I'm usually here M-F all day and can usually meet anytime I'm not in class.)
Office Phone Number: 895-4441
Email address: mmead@digipen.edu
Home page (DigiPen): <http://azrael.digipen.edu/~mmead/www/courses/2020/winter/cs180/index.html>

Day and Time	Room
M/Th, 10:30 am - 11:50 am	Michelangelo

Prerequisites

It will be assumed that the student has successfully passed CS 100 (or CS 101), CS 120, and CS 170. The primary language used in this course is C, although some C++ may be used.

Course Description

This course provides an understanding of the core abstractions of modern computer systems: concurrency, memory management and file systems. Topics covered include processes and threads, context switching, synchronization, scheduling, virtual memory management algorithms, file systems, and disk management.

Objectives and Outcomes

In addition to learning the terminology and concepts associated with the various components of an operating system, the student will acquire practical knowledge, specifically, how to

1. create child processes
2. communicate between processes
3. write simple shell scripts
4. create and manage threads
5. synchronize threads and processes with mutexes and semaphores
6. schedule tasks using different algorithms
7. effectively manage memory
8. create and use virtual machines

Required Textbook

- *Operating System: Three Easy Pieces (August 2018, Version 1.00)* by Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau. This is a FREE textbook that can be downloaded (PDFs) from here: <http://ostep.org/> You can also buy a

Optional Textbook

- *Operating System Concepts, Eight Edition* by Silberschatz, Galvin, and Gagne. Published by John Wiley & Sons. Copyright ©2009 (ISBN: 978-0-470-12872-5). Older or newer versions may be substituted.

Additional References

- *The World Wide Web*. Quite possibly the greatest asset to learning since the teacher and the textbook.

Grading

Grades will be derived from homework assignments and exams. The detailed weightings and letter grades are as such:

Quizzes 20%
Homework 35%
Final exam 45%

You must receive a score of 60% or better on the final exam to pass this course, regardless of your homework and quiz scores.

$x\%$	Grade
$x \geq 93$	A
$90 \leq x < 93$	A-
$87 \leq x < 90$	B+
$83 \leq x < 87$	B
$80 \leq x < 83$	B-
$77 \leq x < 80$	C+
$73 \leq x < 77$	C
$70 \leq x < 73$	C-
$60 \leq x < 70$	D
$x < 60$	F

Attendance is Mandatory

There are no makeup exams or quizzes. Also, for every lecture that is missed, you will lose one point from your final grade (e.g. a 90 becomes an 89). The only exceptions are if you notify me prior to your absence with a valid reason. (Sleeping, studying for another class, working on your game, etc., are not valid reasons for an absence.) However, this is not unlimited. For freshmen there is a school-wide Digipen Freshmen Attendance Policy that I must also follow. Read it on the website. If you do have to miss a class (for any reason), it is your responsibility to contact other classmates to find out what was covered.

Disability Support Services

If students have disabilities and will need formal accommodations in order to fully participate or effectively demonstrate learning in this class, they should contact the Disability Support Services Office at 425-629-5015 or dss@digipen.edu. The DSS Office welcomes the opportunity to meet with students to discuss how the accommodations will be implemented. Also, if you may need assistance in the event of an evacuation, please let the instructor know.

Tentative Schedule (This is a very coarse guideline for the semester and is subject to change.)

Below is a list of topics that will be covered this semester. Depending on time, I may add additional topics or only briefly cover some of the ones listed. The right-hand column corresponds to the chapter titles in the optional text book. Since this is an introductory course on operating systems, there isn't a lot of programming required in this course. A more advanced course would entail larger and more detailed programming projects. In the schedule below, *Online* refers to the CS180 website, and *Chapter* refers to the required textbook. If *Chapter* is in square brackets (e.g. [Chapter 2]), it refers to the optional textbook.

Tentative Schedule (Topics and ordering subject to change)

<i>Week</i>	<i>Topic</i>	<i>Reading</i>
1	Introduction, Virtualization (Virtual Machines)	Online
2	More Virtualization	Online
3	OS Overview	Chapter 1
4	System Call API and Programming	Online, Chapter 3 [Chapter 2]
5	Processes	Chapters 4,5 [Chapter 3]
6	Threads	Chapters 26,27 [Chapter 4]
7	Memory	Chapters 13,14,15,16,17 [Chapter 8]
8	More Memory	Chapters 18,19,20,21,22
9	Process Scheduling	Chapters 6,7,8,9,10 [Chapter 5]
10	Synchronization, Semaphores and Mutexes	Chapters 27,28,31,32 [Chapters 6,7]
11	Storage (disk structure)	Chapters 36,37,38,44 [Chapter 12]
12	File Systems	Chapters 39,40,41,42,43 [Chapters 10,11]
13	TBD	
14	TBD	
15	Finals	

Workload

During the semester there will be several quizzes and a final exam. There will also be a few programming assignments to work on outside of class. These are not large and you will usually have one to two weeks to complete them (although they generally take no more than a few hours to complete). In addition to attending the lectures, you should plan to spend at least 6 hours per week reading, studying, and programming for this class, with a major portion dedicated to experimenting with code samples.

This course requires the student to do good amount of self-study. Reading material will be provided (usually on the website via URLs) as well as sample code for the student to experiment with outside of class. The major reason for students to fail the course is primarily due to lack of time spent reading and experimenting. Simply showing up to the lectures and trying to "memorize" the concepts will likely lead to failure. Please keep this in mind during the course.

Submitting Homework

Programming assignments will use the C programming language. More specifically, all programs must adhere to Standard C. There will be exceptions to some of these standards and you will be instructed how to deal with them. Because much of operating systems implementation is, well, operating system dependent, we may use different compilers in this class. A few examples will require Microsoft-specific code and will necessarily require Windows.

Programming assignments are due at the time/day specified and **NO LATE ASSIGNMENTS WILL BE ACCEPTED**, (even if it's 1 second late). Additional detailed instructions will be provided with each assignment. This usually includes a handout as well as supplemental material available on the web site. **Note that we will be using a submission server as was done in CS120/CS170.** By now, all students should be able to produce code that can at least compile. Code that fails to compile cleanly will not be accepted and will receive a 0.

Code Documentation and Conventions

Unless otherwise specified, all programs must be documented using Doxygen tags just as was done in CS170. There is an extensive help document included with Doxygen which shows examples of each tag. In addition to the “normal” documentation (file and function header comments), you must also put comments next to each header file that you include. The comments must list the functions, types, variables, symbols, etc. that you are using from that library. The purpose of this is for you to demonstrate to the graders and myself that you know why you are including these libraries in your project. Here are a few examples:

```
#include <stdio.h>  /* NULL, printf */
#include <stdlib.h> /* atoi      */
```

Studying Operating Systems

To really get the most out of an operating systems course, it is necessary to have access to various operating systems. Specifically, operating systems that give the user access to everything, including the source code. This is not possible with Windows, as it is a proprietary system, meaning that it’s code and much of it’s internals are protected or hidden from outsiders (users). The most popular operating system that has virtually unlimited access is Linux. Although modifying the source code to the Linux kernel is well beyond the scope of this introductory course, Linux does provide a lot of programs and utilities that can aid in the study of the operating system. Fortunately, all of these tools are readily available and free of charge. We will be discussing both the Windows and Linux operating systems, with a focus on POSIX. Students will be able to install a version of Linux on their computers and use it to help them understand the operating system. There are several ways to install multiple operating systems on a single computer. These will be demonstrated during the first week of the semester so that the students will have sufficient time to familiarize themselves with the system.

Required Software

You will require a 64-bit Unix-like programming environment in this course. If you don’t have access to a true Unix environment (most people do not) you can use Linux, BSD, or Mac OS X. These are freely and readily available to DigiPen students (with the exception of Mac, which is not free unless you have a Mac computer). Note that the Apple Macintosh’s latest operating system (OS X) is based on BSD, which is a Unix-like operating system. All of the homework assignments must compile and run on a 64-bit Linux system. Getting started with Linux will be demonstrated during the first week of class.

Academic Honesty

All homework assignments and exams must represent your own, individual work. It is permissible to discuss assignments (not solutions) with other students in the class, but the solutions must be recognizably your own. Cheating of any kind (copying someone else’s work, allowing others to copy your work, collaborating, etc.) will not be tolerated and will be dealt with SEVERELY (at the discretion of the instructor, which likely will include removal from the class with a grade of F.) Please keep in mind that discussing solutions to exams, quizzes, homework, etc. with students that haven’t taken the exam or turned in the assignment is also prohibited. Ultimately, you are only wasting your time (and money) because if you can’t master the fundamentals covered in this course, you have little hope of succeeding in other courses or as a programmer in the Real World.

Religious Accommodation

DigiPen Institute of Technology provides reasonable accommodations to students who may be absent from activities or incur significant hardship due to religious holidays or observances. These holidays or observances must be part of a religious denomination, church, or religious organization, and the course instructor must be notified in writing during the first two weeks of the course. The institute’s policy for grievances is published in the course catalog.

From the “It-shouldn’t-need-to-be-said-but...” Department

During class, all electronic devices must be turned **OFF**. This includes cell phones, game consoles, digital cameras, laptop computers or any other devices. If you absolutely must have a cell phone on, you must first clear it with me **BEFORE** class begins.

In addition to showing up for class on time, other student responsibilities include proper behavior during class, learning the material, completing assignments correctly, submitting assignments properly and on time, studying for the exams, and participating in class by asking or answering questions during the lectures. **All students are required** to bring a pencil (or other writing instrument) and paper to class to take notes and perform other tasks. Finally, no food is allowed in the classroom.