CS180: Operating Systems I: Man-Machine Interface
Fall 2020

Prerequisites: CS100, CS120 and CS170

General Information:

Class Schedule:	CS180F20-A Mon/Wed 11:00am-12:40am
	CS180F20-B Mon/Wed 2:0pam-3:40pm
Class room:	CS180F20-A Plato (Wed) / Edison (Mon)
	CS180F20-B Plato (Wed) / Newton (Mon)
Professor**:**	William Zheng
Contact**:**	Phone: 6577 1745
	Email: william.zheng@digipen.edu
Class web Page**:**	https://distance.digipen.edu
Office Hours**:**	Thu/Fri 11:00am-12:00am.
	Other times appointment by email/phone.

Description:
This course presents an overview of modern operating systems as implemented on personal computers. It presents an overview of what an operating system is and does, with emphasis on the following topics: organization and design, process management, threading, interprocess communication, process synchronization, and memory management.

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

- create a child process
- set up a pipe between processes
- write a user-level thread library
- create and manage threads
- synchronize threads and processes with mutexes and semaphores
- schedule tasks using different algorithms
- manage a memory pool

Textbooks:
Operating System Concepts, eighth edition, by Abraham Silberschatz, Peter Galvin, and Greg Gagne; published by John Wiley & Sons, 2014. ISBN: 978-11180-9375-7.

References:
Course materials and activities, such as lecture notes, assignments, forums, announcements, and calendar will be made available through distance.digipen.edu site.

Outline and Tentative Dates:
Please note that this is a tentative organization of the course and may be subject to change. Below is a list of topics that will be covered this semester. Depending on time, I may add additional topics or skip some of the ones listed. The right-hand column corresponds to the pages within the text book that cover the corresponding topic. The readings should be done before the lecture is given.
The midterm and the final exams are comprehensive, covering all topics taught up to the point of the examination. There will be 4 or 5 programming assignments during the semester. The programming assignments may require up to 8 hours per week, depending on your ability. Understanding the material covered in class, readings from the text book, and doing practice exercises will require about 4 hours per week, depending on your grasp of the subject matter.

Week	Week of	Topic	Textbook references
1	7 Sept	Introduction to OS. Why study OS. Booting. History of Computing. Multiprogramming and Spooling. Interrupts and Polling. Interrupt Handling. System Calls. OS Architectures– the monolithic kernel, microkernel and hybrid. Pros and cons.	Chapter 23.1 – 23.2 Chapter 1.1 – 1.8 Chapter 2.1 – 2.7
2	14 Sept	Processes– Process states, Process Control Block, Context Switching of a process, Process Creation and Termination.	Chapter 3.1 – 3.3
3	21 Sept	Inter-process communication: pipes. Named Pipes and Anonymous Pipes. Pipes Creation. Windows and Linux Pipes Demo. Threads Introduction.	Chapter 3.6.3 Chapter 4.1 – 4.4
4	28 Sept	Multithreading Models. Introduction to Process Scheduling. Metrics for scheduling algorithm. Non preemptive scheduling algorithms. Pre-emptive scheduling algorithms.	Chapter 3.6.3 Chapter 4.1 – 4.4 Chapter 5.1 – 5.2
5	5 Oct	Inter-process communication: message queues, shared memory. Comparison between the IPC methods.	Chapter 5.3 – 5.4 Chapter 3.4 – 3.5
6	12 Oct	Race Condition, Process Synchronization, The Critical Section Problem, Criteria for Critical Section Problem, Software Solutions, Hardware Solutions, Solutions requiring OS Support, Priority Inversion Problem, Busy waiting versus Sleep and Wakeup Midterm Review Midterm Examination	Chapter 6.1 – 6.4
7	19 Oct	Study break – no classes	
8	26 Oct	Semaphores Intro, Semaphores and Synchronization, Implementation of Semaphores, Classical Synchronization Problems	Chapter 6.5-6.6
9	2 Nov	Deadlocks – What is Deadlock, Banker's Algorithm, Resource Allocation Graph Physical and Logical Address Spaces, Memory Management Unit (MMU), Address translation	Chapter 8.1 – 8.3, Chapter 8.6
10	9 Nov	Contiguous Memory Allocation (Different Algorithms), Segmentation Scheme(Segment logical address, Segment Table, Address Translation, Segmentation Fault, Recovering from segmentation fault)	Chapter 8.1 – 8.3 Chapter 8.66
11	16 Nov	Paging Scheme – 1 level and 2 level paging (Internal and external Fragmentation, Paging logical address, Page Tables, Size of Page Tables, 2-level paging, Comparisons between 1-level and 2-level paging)	Chapter 8.4, 8.7
12	23 Nov	Demand paging and Page replacement algorithms Implications on programming and thrashing Disk Structure, Disk Scheduling, RAID structure File Systems Introduction Allocation Methods (FAT, NTFS, EXT), Free Space Management Finals Revision	Chapter 9.1 – 9.4, 9.6 Chapter 12.1 – 12.7 Chapter 11.1 – 11.5
13	30 Nov		
14	7 Dec		
15	14 Dec	Finals week	

Please note that this is a tentative list of exam, assignment submission dates and holidays. Note that all exams and submission dates are subject to change. The specific dates will be published in the course web page.

Week Of	Topic
Week 1	Publication of specification for programming assignment 1
Week 2	Quiz 1

Week 4	Programming assignment 1 deadline
Week 4	Publication of specification for programming assignment 2
Week 4	Quiz 2
Week 6	Programming assignment 2 deadline
Week 6	Quiz 3
Week 8	Midterm Publication of specification for programming assignment 3
Week 10	Programming assignment 3 part 1 – deadline
Week 10	Quiz 4
Week 11	Quiz 5
Week 12	Programming Assignment 3 part 2 – deadline Publication of specification of programming assignment 4
Week 13	Programming assignment 4 deadline
Week 13	Quiz 6
Week 14/15	Finals

Grading Policy:

- Participation/Attendance: 2%
- Quizzes: 18%
- Programming Assignments: 40% (7 + 7 + 12 + 7 + 7)
- Midterm: 15%
- Final exam: 25%
- Final letter grade algorithm: A: 93 – 100%; A–: 90 – 92.99%; B+: 87 – 89.99%; B: 83 – 86.99%; B–: 80 – 82.99%; C+: 77 – 79.99%; C: 73 – 76.99%; C–: 70 – 72.99%; D: 60 – 69.99; F: < 60%
- You must receive an average score of 60% on both the midterm and final exams to pass this course, regardless of your quiz/assignment scores.

Rubrics and Assessment:

There will be two major exams during the semester – midterm and final. The subject matter for these exams will be specified to you well in advance of the examination date. We would be having 6 quizzes this semester. In addition, there would be 4 or 5 programming assignments.

Every quiz consist of the following:

- Quizzes are given at the beginning of classes.
- In preparation for quizzes, students must be seated as far apart from one another as possible.
- All quizzes are closed-book quizzes. All bags should be in front of the room during the quiz.
- There should be no notes or textbooks with the students during the quiz.
- The students shall need no more than writing tools such as pencils or pens during the quizzes.
- A quiz takes about 20 minutes, covering topics taught between the last quiz and this one.

Every assignment will consist of the following sections

- **Topics**: Specifies the list of topics covered by the assignment.
- **Programming Statement**: Specifies the assembly program to be implemented or C statements to be translated into assembly program.
- **Non-Programming Statement**: Specifies the questions to be answered.
- **Deliverables**: Specifies the list of files and documentation to be submitted.
- **Submission Deadline**: Specifies the time and date at which the assignment is due.
- **Objectives**: Specifies the detailed list of objectives to be completed and the grade associated with each objective.
- **Declaration**: Student affirmation about sole ownership and non-plagiarism in the implementation and completion of the assignment.

The objectives section – in plain language - specifies the assessment plan for evaluating and grading your assignment submission. Please read this section carefully. Every assignment submission must be accompanied by your signed declaration. *You will not be given credit for the assignment if you have not signed and attached this declaration to your submission.*

Mechanisms and Procedures:

- **Attendance:** The duration of this semester is 15 weeks. During the first 14 weeks of the semester, the class will meet bi-weekly (2 lectures) for a total of 28 lecture sessions. There will be a three hour long final exam scheduled for the final week of the semester. Attendance is mandatory. There are no makeup quizzes or exams. You will be penalized for unexcused absences from class according to the following scale:
  - Three (3) or more unexcused absences will result in a 10% reduction of your overall course grade.
  - Six(6) or more unexcused absences will result in a 20% reduction of your overall course grade.
  - Nine (9) or more unexcused absences will result in a 30% reduction of your overall course grade.
  - Twelve (12) or more unexcused absences will result in your automatic failure in the course irrespective of your performance on homework, assignments, quizzes, and exams.

  Medical leave, family emergencies and national duty –accompanied by appropriate documents
  will be the only valid reasons for this policy. Sleeping, studying for another class and/or exam, working on your game, etc., are not valid reasons for an absence.
  On the other hand, Class participation will boost your grade if you are on the border – for example, it is possible to get an A- with an overall grade of 89%.

- **Workload:** There will be two major exams during the semester – midterm and final. There will be 5 programming assignments to work on outside the class. In all cases, you will have ample time to complete them. Plan on spending at least 8 hours per week on these assignments. In addition to attending the lectures, you should plan to spend at least 4 hours per week reviewing, reading, and studying for this class. Additional, optional readings may be assigned. These will be readings that are related to course materials but are purely to help increase your understanding of the concepts presented. Any material on these readings that is not specifically covered in the assigned readings will not be on the test.

- **Submitting Assignments:** Submission procedures for programming assignments are detailed below. ***Failure to follow any of the following submission procedures and guidelines will result in deductions (10%) from your assignment grade.***
  - The programming assignments could either be Win32-based or Linux/Unix based.
  - For Win32-based assignments, your programming submission must be a compile-able C program under Visual Studio 2015 or newer version.
  - For Linux/Unix based, your programming submission must be compile-able using GNU gcc or otherwise specified in the assignment statements.
  - **ANY** Source, header, data, or README files submitted must start with the following header:

```
/*  Start Header
*****************************************************************/
/*!
\file <put file name here> (e.g. main.cpp)
\author <provide your name, student login, and student id>
(e.g. Joon Edward Sim, esim, 60001906)
\par <provide your email address> (e.g. email: esim\@digipen.edu)
\date <date on which you created this file> (e.g. Jan 09, 2017)
\brief
Copyright (C) 20xx DigiPen Institute of Technology.
Reproduction or disclosure of this file or its contents without the
prior written consent of DigiPen Institute of Technology is prohibited.
*/
/* End Header
*******************************************************************/
```

- To submit your programming assignment, organize a folder consisting of ALL relevant source code (including source files, header files, data files), documentation files. In other words, *your* *submission must be ready for compiling and linking by us*. For example, if your login is *foo* and assignment 2 is being submitted, your folder would be named **cs180_foo_2**. The folder naming convention is:

<class>_<student login name>_<assignment#>

- Zip this folder and name the resulting file using the following convention:

  <class>_<student login name>_<assignment#>.zip

  For example, if your login is foo and you are submitting assignment 2, your zipped file would be named as: **cs180_foo_2.zip**
- Next, upload your zip file after logging into the course web page using the link https://distance.digipen.edu.
- Finally, perform a sanity check to determine if your programming submission follows the guidelines by downloading the previously uploaded zip file, unzipping it, then compiling, linking, and executing your submission.

Late Policy:

Assignments (both programming and non-programming work) are due at the specified time on the specified due date. More precisely, any programs and/or non-programming work submitted ***after the specified time on submission day will be considered late. Late assignments will be penalized 10% for each day (including weekends and holidays) beyond the deadline.***

Last Day to Withdraw:

The final date to withdraw from this course is shown on school calendar. In order to withdraw from a course it is not sufficient simply to stop attending class or to inform the instructor. In accordance with policy, contact your advisor or the Registrar to begin the withdrawal process. The last day for withdrawal from this course is cited in the official catalog.

Academic Integrity Policy:

CS 180 programming assignments are NOT group projects. They must represent a student's own individual work. It is reasonable for students to consult or discuss general solutions to an assignment. However, it is unreasonable for students to collaborate on detailed solutions, to copy code, or to give away code.

Cheating, or academic dishonesty in any form will not be tolerated in this course. Cheating, copying, plagiarizing, or any other form of academic dishonesty (including doing someone else's individual assignments) will result in, at the extreme minimum, a zero on the assignment in question, and could result in a failing grade in the course or even expulsion from DigiPen.

Academic dishonesty or cheating occurs when a student represents someone else's work as his/her own, or assists another student in doing so. This can happen on exams, quizzes, homework, or projects. Academic dishonesty may also occur when a student uses any prohibited reference or equipment in the completion of a task. Examples include using a calculator, or notes, or books, or the internet when such sources are prohibited for that task. Plagiarism is a common form of academic dishonesty. This takes the form of copying and pasting excerpts from the web and representing them as original work. The type and severity of any occurrence, as well as the legitimacy of any claim of academic dishonesty will be judged by the instructor and the disciplinary committee. All students are asked to help in promoting a culture of academic integrity by discouraging cheating in all forms. Please consult your student handbook for additional information and details on DigiPen Singapore's academic integrity policy.

External Preparation:

It is expected that the students in this class spend 6 hours on average per week for outside classroom activities through the trimester, including, but not limited to, homework, reading assignments, project implementation, group discussions, preparation of examinations, etc.

Disability Support Services:

Students who have special needs or medical conditions and require formal accommodations in order to fully participate or effectively demonstrate learning in this class should contact the Student Life & Advising Office (studentlife.sg@digipen.edu) at the beginning of each semester. A Student Life &

Advising Officer will meet with the student privately to discuss how the accommodations will be implemented.