Operating Systems
CS 421 - Fall 2017

Instructor: George Thomas **Office:** Halsey 218
Email: thomasg@uwosh.edu **Phone:** 424-2069

Office Hours: MW 9:30-11:00
TR 11:30-12:30
Or by appointment

Lectures: TR 9:40-11:10 Halsey 208

Prerequisites: Computer Science 271 and Computer Science 212, each with a grade of C or better

Textbooks:

- **Required:** *Operating Systems: Principles and Practice*, Thomas Anderson and Michael Dahlin, 2nd edition, 2014, Recursive Books.
- **Recommended:** *Operating Systems: Three Easy Pieces*, Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau. Available at *http://pages.cs.wisc.edu/~remzi/OSTEP/*

The lectures will be based primarily on the required textbook, with significant portions taken from the recommended textbook.

Course Website: UWO D2L

Note: If you have special needs, please do come and talk to me at the end of the first class.

Current Catalog Description:
An introduction to operating systems concepts. Topics covered include: interrupts, memory allocation, virtual memory techniques, process scheduling and synchronization, deadlocks, resource allocation, and file systems. A major programming project will be assigned to provide experience with operating system design.

Course Outcomes:
Outcomes #1

1a. Demonstrate by explaining or describing objectives and functions of modern operating systems

1b. Demonstrate sound knowledge of the user and system views of the operating system and able to differentiate between these two viewpoints.

1c. Adequately explain the structure and mode of operations of today's operating system as they relate to such concepts as multiprogramming, time-sharing, swapping, interrupt, and dual mode operation.

1d. Demonstrate intimate knowledge of the issues of design and implementation as well as the desired user and system properties.

1e. Show understanding of the influences of security, distributed and special-purpose systems, and different computing environments on modern operating system design.
1f. Demonstrate the ability to discuss the tradeoffs inherent in operating system design and able to identify potential threats and their safeguards in designed systems.

Outcomes #2

2a. Demonstrate ability to compare and contrast the many possible methods (e.g., simple, layered, modular, and microkernel) of structuring the operating system.
2b. Show sound knowledge of the difference between asynchronous and synchronous interrupts and of the relative advantages of interrupts over polling.
2c. Able to discuss the system call concept and able to differentiate between the needs for the system call interface and the application program interface.
2d. Show how system programs manage the resources used by applications.
2e. Demonstrate the ability to illustratively explain why the services provided to users constitute a different set of functions than that provided to the system.

Outcomes #3

3a. Able to explain process state and process control block using appropriate examples to highlight their components.
3b. Able to illustratively describe the creation and termination, scheduling, and interprocess communication features of a process.
3c. Show the ability to compare and contrast the various types of multithreading models.
3d. Demonstrate a deep understanding of some of the issues encountered with multithreaded programs (e.g., `fork`() and `exec`() system calls, cancellation, signal handling, thread pools, thread specific data, and scheduler activations).
3e. Able to describe possible run-time problems arising from the concurrent operation of many separate tasks.
3f. Demonstrate the ability to explain how software and hardware are individually used to solve mutual exclusion problem.
3g. Able to discuss the evaluation criteria for selecting a particular system's CPU scheduling algorithm and the various available criteria comparing CPU scheduling algorithms.

Outcomes #4

4a. Able to compare and contrast paging and segmentation techniques.
4b. Know how to evaluate the tradeoffs of the components parts in the memory hierarchy in terms of size, cost, and access time.
4c. Able to compare and contrast demand paging and copy on write techniques.
4d. Show the ability to analyze such techniques or policies as page replacement, swapping, and thrashing.

Outcomes #5

5a. Able to discuss file system design tradeoffs including access methods, file sharing, file locking, directory structures, and protection.

5b. Demonstrate sound knowledge and deep understanding of a file's attributes and operations.

5c. Demonstrate a deep understanding of the details of a local file system and directory structure implementation.

Course Grading Policy:

Your final grade for this course will be based on four components, namely exams, programming projects, homework and class participation. Your overall numerical grade for the course will be computed as the weighted sum of the component grades using the following weights:

Component	Weight
Exams (3)	50%
Projects	25%
Homework	20%
Class Participation	5%

Tentative Exam Dates are as follows:

- **Exam 1 – Thursday, 10/05**
- **Exam 2 – Thursday, 11/02**
- **Exam 3 – Thursday, 12/14**

Your letter grade for the course will be computed as follows:

Numerical Score	Grade	Numerical Score	Grade
>=92	A	72-78	C
90-92	A-	70-72	C-
88-90	B+	68-70	D+
82-88	B	62-68	D
80-82	B-	60-62	D-
78-80	C+	<60	F

While this overall grading scheme is fixed, I will be happy to discuss any issue you may have with individual grades. If you notice a mistake or have a question regarding a specific grade, please come and talk to me *as soon as possible*. Do not wait until the end of the semester to bring up grading issues.

Project and Homework Deadlines:

Each homework will come with a deadline (day and time) by which it must be submitted. Late homework submissions will NOT be accepted.

Each project will also come with a deadline (day and time) by which it must be submitted. You are allotted *three* project credit days you can use through the semester. A credit day is

exactly 24 hours or less. You can use unused credit days to submit a project after its deadline, without penalty. Any project submitted after the deadline, plus any credit days you have unused, will receive a zero.

For example, if you have 2 unused credit days available and a project is due on Tuesday at 5:00PM, you can submit it anytime by exactly Thursday at 5:00PM without penalty. Do note that if you submit your project on Thursday at 5:01PM, you will be penalized 100% of the score of the project and thus receive a zero! Note also that if you submit your project on Wednesday at 5:01PM, you will be charged two credit days (but no penalty, obviously).

Attendance and Participation:
While attendance at lectures is optional, there may be class participation exercises to reinforce the material presented during lecture. You will, obviously, have to be present during lecture to get any participation points. I do suggest that you not only attend **every** class meeting but also come **prepared** for and **participate** actively in it. I **strongly encourage you to ask any question** or raise any issue you have with the course either during or at the end of class, or during my office hours. I will also gladly meet with you by appointment.

Absences and Extensions:
Extensions on deadlines may be granted at the discretion of the instructor if you provide a valid justification (in the form of a written excuse from a medical doctor or the Dean of Students Office) **before** the due date.

If you miss a scheduled exam (tentative dates are provided), you **may** be able to take a make-up exam provided you give the instructor a valid justification (see above) ahead of time if possible. Only one make-up exam will be given. It will be a comprehensive exam scheduled at the end of the semester. Similarly, there will be no make-up quizzes unless the instructor is provided with a valid justification (see above) for your absence on the day of the quiz.

Collaborating versus Cheating:
Unless otherwise stated in the assignment or project, all submissions must be entirely your own work. While it is acceptable to discuss the assignments at a high level (for example, at the design level) with others, you must submit your own work. **You may not "borrow" any piece of code or design of any length from someone else, the internet, or any other source, unless you can live with a zero and the other potential academic sanctions of cheating** (see UWO Student Discipline Code 2007, Chapter UWS 14).