Introduction to Systems Programming & Operating Systems
CIS 3207 / CIS 5012
Spring, 2018

Professor: **Dr. Eugene Kwatny**

Office: SERC 324
Phone: 215-204-1679
EMAIL: gkwatny@temple.edu

Course Meeting Times:

3207 Section 1:
- Tuesday & Thursday 9:30 AM - 10:50 AM [Tuttleman 302]
- Laboratory: Monday 9:00 AM - 10:50 AM [SERC 204]

3207 Section 2 and 5012:
- Tuesday & Thursday 12:30 PM - 1:50 PM [Tuttleman 302]
- Laboratory: Monday 9:00 AM - 10:50 AM [SERC 206]

Office Hours: [appointments are encouraged]
Tuesday 2:30 PM – 3:30 PM
Thursday 3:00 PM – 4:00 PM
(**other times available by appointment**)

Teaching Assistants:

3207.001	**Chenglong Fu** Office hours: TBD Office: SERC 337 Email: Chenglong Fu Office phone #:
3207.002/ 5012	**TBD** Office Hours: TBD Email:

Prerequisites:

Grade of C- or better in: C+IN SC 1166 or Math 2196, C+IN SC 2107, and C+IN SC 2168.

- The 'C' programming language will be used for laboratory experiments, and students are expected to have some elementary experience with programming in C. [Online resources for programming in C are available through the class Canvas site]

Required Textbooks:

" Operating Systems: Internals and Design Principles, 9th Edition", William Stallings, Pearson, ISBN-13: 9780134670959, 2018
"Computer Systems - a Programmer's Perspective",3e, Bryant & O'Hallaron, 2016, Pearson [same textbook used in 2107, the 2011 edition is OK]

A helpful programming resource for the course: ""The Linux Programming Interface: A Linux and UNIX® System Programming Handbook" by Michael Kerrisk", Published by "No Starch Press" [see Canvas 'Additional Text Book References"]

Grading:

Course grade will be determined by :

Final Exam (30%) [the final exam will be comprehensive, covering all course material]
Midterm Exam (25%) [covers all materials up to the date of the midterm]
Laboratory Projects (25%)
Quizzes, homework, class participation (20%) (quizzes will generally not be announced in advance and there will be a quiz every 7 to 10 days).
Unexcused absences will result in a decrease in the course grade. There is no makeup for missed quizzes and the grade for a missed quiz is 0. The lowest quiz grade will be dropped.

We have many projects to complete this semester. If you delay in completing a project, you will end up behind, with greater delay, in each subsequent project. Except for your reading assignments, the lab projects are your primary out of class work. You should have time to complete them.
Penalty for late project submission:
up to 1 day late, maximum grade of 90% of project value
1 to 2 days late, maximum grade of 80% of project value
2 to 3 days late, maximum grade of 70% of project value
3 to 4 days late, maximum grade of 60% of project value
4 to 5 days late, maximum grade of 50% of project value
after the 5th day, the submission will not be accepted and a grade of 0 will be assigned.

Aim of Course

To introduce the basic concepts of operating systems; with a focus on understanding and evaluating operating systems and the most important computer architectural issues impacting operating system design and implementation. The laboratory portion of the course will be based on experiments and assignments implementing and applying operating system mechanisms and components. The course will be an opportunity to strengthen your software development abilities, your use of the C/C++ languages in problem solution, and introduce you to the use of operating system libraries and system calls.

Introduction to the Course

CIS 3207 is a course about Systems Programming and about Operating Systems. These are two very related topics. One, operating systems, is concerned with the internal software that transforms or manages the devices and resources that comprise the computer system. The other topic, systems programming, is concerned with writing programs that interact with and make direct use of the operating system services.

This semester we will explore the main features and principles of operating systems. In order to pursue this exploration, we will also concern ourselves with computer architecture. We must understand the architecture and devices that make up the computer system to understand software at the level of controlling a computer system,.

We will apply these concepts, in particular, to the study of two operating systems that are widely used: Microsoft Windows (actually the Windows NT family) and Linux. In the process, we will study the systems APIs (application programming interfaces) for the two OS'.

The 'C' (or C++) programming language will be used for laboratory experiments and students will have to enhance their own C programming capability (beyond the C programming introduced in CIS 2107. The Canvas '3207 course' includes many C programming references and training materials). The Lynda.com course "C Essential Training" is available through TUPortal and is a valuable resource.

Course Topics

- Overview of operating systems
- Operating system principles and computer architecture
- Operating system kernels
- Processes and threads
- Concurrency
- CPU Scheduling and dispatching
- Memory management and virtual memory
- Device management
- File systems

Learning Objectives

OS/Overview of Operating Systems
Describe and explain the objectives and functions of modern operating systems, the tradeoffs inherent in operating system design.

OS/Operating System Principles
Explain the concept of a logical layer and the benefits of building abstract layers in hierarchical fashion; describe the value of and demonstrate the use of APIs and middleware; contrast kernel and user mode in an operating system; describe the advantages and disadvantages of using interrupt processing; explain the use of device drivers and I/O queues.

OS/Concurrency
Define concurrency. Describe the need for concurrency within the framework of an operating system; demonstrate the potential run-time problems arising from the concurrent operation of many separate tasks; describe the range of mechanisms that can be employed at the operating system level to realize concurrent systems; explain the different states that a task may pass through and the data structures needed to support the management of many tasks; summarize and apply techniques for achieving synchronization in an operating system; and, describe reasons for using interrupts, dispatching, and context switching to support concurrency in an operating system.

OS/Scheduling and Dispatch
Compare and contrast the common algorithms used for both preemptive and non-preemptive scheduling of tasks in operating systems; discuss the types of processor scheduling such as short-term, medium-term, long-term, and I/O; describe the difference between processes and threads; explain and apply thread models and user and kernel level threads; discuss the need for preemption and deadline scheduling.

OS/Memory Management
Explain memory hierarchy and cost-performance trade-offs; summarize the principles of virtual memory as applied to caching and paging; defend the different ways of allocating memory to tasks, citing the relative merits of each; describe the reason for and use of cache memory (performance and proximity, different dimension of how caches complicate isolation and VM abstraction); and, discuss the concept of thrashing, both in terms of the reasons it occurs and the techniques used to recognize and manage the problem.

OS/Virtual Machines
Explain the concept of virtual memory and how it is realized in hardware and software; differentiate emulation and isolation; discuss hypervisors and the need for them in conjunction with different types of systems.

OS/Device Management
Explain the key difference between serial and parallel devices and identify the conditions in which each is appropriate; identify the relationship between the physical hardware and the virtual devices maintained by the operating system; explain buffering and describe strategies for implementing it; differentiate the mechanisms used in interfacing a range of devices; describe the advantages and disadvantages of direct memory access and discuss the circumstances in which its use is warranted; describe memory-mapped I/O.

OS/File Systems
Describe the choices to be made in designing file systems; compare and contrast different approaches to file organization, recognizing the strengths and weaknesses of each; construction of a file system; summarize how hardware developments have led to changes in the priorities for the design and the management of file systems; summarize the use of journaling and how log-structured file systems enhance fault tolerance.

Content

Reading and lecture material from Stallings ("Operating Systems) Chapters 1 - 12.
Reading and lecture material from Bryant & O'Hallaron ('Computer Systems') Chapters 6, 10, 13.

Lecture materials and supplemental documents and texts are available through Canvas.

DATES of IMPORTANCE:

First class: Tuesday, January 16
Last day to drop (tuition refund available): Monday, January 29
Spring Break (no classes) March 5 – March 11
Last day to withdraw (no refund): Wednesday, March 21 .
Last Class: Thursday April 26
Last Lab: Monday April 30
Study Days: Tuesday, May 1 and Wednesday, May 2.
Final Exam:
3207 Section 1 - Thursday May 3, 8 AM - 10 AM.
3207 Section 2/5102 - Thursday May 3, 10:30 AM – 12:30 PM

Student Responsibilities

Student's are responsible for reading all assigned text materials, handouts, and referenced sources. Students are responsible for participating in classroom discussions and discussions carried out electronically though Canvas or other class facilities.

The CIS laboratory computer systems are available for use in homework and laboratory exercises. Access to the computer systems in CIS labs is through Temple University AccessNet username and password. SERC laboratories 204, 206, 357, and 359 have dual boot Windows and Linux systems. We will be using the computer systems in room 204 and 206 for in class laboratory assignments, although the same resources are available in each of the CIS SERC labs. (Note that this semester, the TECH Center computers will also be available for work on lab assignments when CIS labs are not available). You can use your own computer systems for your project work. You are responsible for performing and completing all of the laboratory exercises. This includes becoming familiar with, and being able to use, all of the tools and software that are to be used in these exercises.

Much of the source code work in the course will require you to have familiarity with (or become familiar with) the C language and development environments for compiling and building C or C++ programs in both Windows and Linux. Reference material for the C Language is available via the CIS 3207 Canvas site.

Students are responsible for taking all quizzes and exams in the course. All work turned in for grading or review by the instructors of the course **must be the students own work**. The objectives of the course can only be met by your doing all of the work and presenting only your work for grading. **Presenting work that is not your own will result in disciplinary action and no credit.**

Student attendance to **each class** and **each laboratory** is **Mandatory**.

Students who miss the final exam and do not make alternative arrangements with me before the exam, will receive a grade of F.

Student and Faculty Academic Rights and Responsibilities

Freedom to teach and freedom to learn are inseparable facets of academic freedom. The University has a policy on Student and Faculty and Academic Rights and Responsibilities (Policy #03.70.02) which can be accessed through the following http://policies.temple.edu/PDF/99.pdf.

Accommodations for Students with Disabilities.

Any student who has a need for accommodation based on the impact of a documented disability, including special accommodations for access to technology resources and electronic instructional materials required for the course, should contact me privately to discuss the specific situation by the end of the second week of classes or as soon as practical. If you have not done so already, please contact Disability Resources and Services (DRS) at 215-204-1280 in 100 Ritter Annex to learn more about the resources available to you. I will work with DRS to coordinate reasonable accommodations for all students with documented disabilities. (http://www.temple.edu/studentaffairs/disability/accommodations/).

Collaboration and Cheating Policy [Please READ Carefully]

We encourage you to discuss the problem sets and programming assignments with your colleagues. We welcome discussions of possible interpretations of questions, solution approaches, and points of confusion. You are also welcome to use existing public libraries in your programming assignments (such as public classes for queues, trees, etc.) You may also look at operating systems code for public domain software such as Linux. Such activities qualify under approved collaboration practices and you are welcome to take advantage of them.

You may not look at any course project material relating to any project similar to this course's class projects. For example, you may not look at the work done by a student in past years' courses, and you may not look at similar course projects at other universities. If you are unsure about whether a particular source of external information is permitted, contact the instructor before looking at it.

Note that cooperation is not the same thing as cheating. **You** must understand and generate the solution, and you must not copy all or any part of someone else's solution. The project assignments and exams must be the work of the student turning them in. Students who violate University rules on scholastic dishonesty are subject to disciplinary penalties, including the possibility of failure in the course and/or dismissal from the University. Because such dishonesty harms the individual, all students, and the integrity of the University, policies on scholastic dishonesty will be strictly enforced.

It is generally OK to verbally discuss the concepts needed to do projects assignments. These discussions should focus on overall approach and understanding, not the detailed answer to the specific problem. These guidelines will help you keep on the right side of the line:

First, other than to the TA and instructor, it is never OK to look at the written work of another person or show another person your written work until after all grading on an assignment is completed. This includes looking at paper print-outs, sketching solutions on a white board or napkin, or looking at a screen to help debugging. It should go without saying that copying other people's code or solution sets is strictly prohibited.

Second, everyone in the class is expected to take appropriate measures for protecting one's work. For example, you should protect your files and printouts from unauthorized access.

Note that these guidelines are necessarily generalizations and cannot account for all circumstances. Intellectual dishonesty can end your career, and it is your responsibility to stay on the right side of the line. If you are not sure about something, **ask**.
For the in-lab parts of laboratory projects completed in teams, both team members should contribute equally and will be graded individually. The write-ups and out-of-class portions of labs must be completed independently.

In summary, when you are turning in an assignment with your name on it; what you turn in must be your work, and yours alone. Cheating will not be tolerated.

Cheating, looking at someone else's paper during a quiz or exam will not be tolerated. University policies on scholastic dishonesty will be strictly enforced. In a similar way, you should protect your exam papers from view of others during the exam.

Incomplete Grade Policy:

Incomplete grades will be granted only in rare circumstances and require the approval of the instructor. Students who have not completed all the requirements prior to the last day of class will need to talk with the instructor about how this could be rectified, if at all.

Resources

We will use the Canvas course management system in this course, https://templeu.instructure.com/ or access through the TUPortal (http://tuportal.temple.edu). In order to access Canvas you must have an Temple AccessNet account (all Temple students have one). Registered students will be automatically enrolled in the CIS 3207 Canvas course. But this can only happen if you have an AccessNet account and are registered for the course. Until you are registered for the course and have such an account, you cannot access the course materials in Canvas.

Your AccessNet account also gives you login capability for Astro (astro.temple.edu). All materials for the course will be available from Canvas. This includes a threaded discussion list, class notes and lecture slides. Students will be required to interact with themselves, and the instructor via this discussion list or other tools (e.g., a wiki) provided for the course. Canvas and the discussion lists are accessed through a web browser.

Announcements for the course will be provided via Canvas. You are expected to visit the Canvas course for CIS 3207 on a regular basis for updates to lecture notes, slides and other course information.

Project and homework assignments are to be submitted through Canvas in electronic form.

Office visits, Voice Mail and EMAIL are encouraged for communication with the instructor.

Last Modified: January 7, 2018