Operating Systems - CIS 5512
Fall, 2014

Professor: **Dr. Eugene Kwatny**

Office: Room 1014 Wachman (10th Floor) / SERC 324
Phone: 215-204-1679
EMAIL: gkwatny@temple.edu
Class Hours: Thursday 5:30 - 8:00 PM [Tuttleman room 302 / SERC 359]
Office Hours: Tuesday 11:00AM - 12:00 AM
Thursday 2:30 PM - 4:30PM
(other times available by appointment)

Required Textbook:
"Operating Systems", Deitel, Deitel and Choffnes, 3rd edition 2004, Pearson Education

Highly Recommended for Linux/Unix programming:
"Interprocess Communications in Linux: The Nooks and Crannies", Gray, Pearson Education, 2003
(This book is available electronically through the Temple Libraries)
[Also, see attached list of additional references]

Grading: Course grade will be determined by
Final Exam (35%) [the final exam will be comprehensive, covering all course material]
Midterm Exam (35%)
Homework, problem assignments and class participation (30%)

Aim of Course

To expand on the basic concepts for understanding and evaluating operating systems and the most important computer architectural issues impacting operating system design and implementation. This course will concentrate on theoretical and practical issues of kernels, threads and processes; concurrency, multi-processor management and virtual memory, and file systems.

Course Topics

1. Introduction to Operating Systems (OS): kernel: monolithic, microkernel, distributed systems
    a. Virtual Machines and the role of the OS in virtualizin computers
2. Processes and Threads
    a. Implementation of kernel level threads and user level threads
    b. POSIX threads and implementations
3. Interrupts and exceptions
4. Concurrency

    a. Theoretical introduction to synchronization and synchronization primitives
    b. Development of synchronization primitives: busy-waiting using spin-locks, ticket (Dekker) algorithm, bakery algorithm; semaphores, event counts, sequencers; monitors; message-passing
5. Single and Multiprocessor Management
    a. Examination of single CPU scheduling methods
    b. Development of a simple multi-CPU, multi-process scheduling algorithms
6. Examination of Memory management and virtual memory in single CPU systems
7. Introduction to multiprocessor architecture
8. Organizations of multi-processor operating systems
    a. Multiprocessor memory access architectures
    b. Multi-processor scheduling, process migration, load balancing
9. File Systems

Content

Reading and lecture material from Deitel ('Operating Systems') Chapters 1 - 6, 8 - 11, 15, part of 17 (related to synchronization and mutal exclusion), 19 and case studies in part 8 related to lectures; and Professor Kwatny's notes/slides on concurrency dealing with mutual exclusion, locks, semaphores, monitors and message-passing.

Deitel chapters 1 and 2 contain material that is a review of topics that you should be familiar with. Concepts in these chapters will be referenced in class discussions. The first three lectures will cover materials discussed in chapters 1 - 4 as well as additional material.
The subsequent 5 or 6 lectures will be concerned with concurrency and concurrency mechanisms. Chapters 5, 6 and part of 17 as well as Dr. Kwatny's notes/slides should be read for those lectures.
The next set of lectures will be concerned with Multiprocessor management. Chapter 15 and some handouts will be the reading for these lectures.
Memory Management and Virtual Memory will be the topics of the next set of lectures. Chapters 8 - 11 and Dr. Kwatny's notes/slides are the reading materials for these lectures.
The next set of lectures will concentrate on File Systems and Chapter 13 will be the basis for those lectures.

Lecture materials will also be available through Blackboard.

DATES of IMPORTANCE:

First class: Thursday, August 28
Labor Day Holiday: Monday September 1
Last day to drop (tuition refund available): Monday, September 8.
Last day to withdraw (no refund): Tuesday, October 21.
Fall Break (no classes): Monday November 24 – Wednesday November 26
Thanksgiving Holiday (no classes) Thursday November 27 – Friday November 28
Last Class: Thursday, December 4 .
Study Days: Tuesday, December 9 and Wednesday, Wednesday December 10.
Final Exam: Thursday December 11, 5:30 PM - 7:30 PM.

Student Responsibilities

Student's are responsible for reading all assigned text materials, handouts, and referenced sources. The Linux workstation systems (all CIS laboratories, are dual-boot Windows 7 and Ubuntu linux)), and the University's Astro (astro.temple.edu) system are available for student use with login using your AccessNet account. Astro access is available for all students using your AccessNet account. You can also use CIS-LINUX2.temple.edu, a system operated by the CIS department. You can get information about CIS-LINUX2 from consultants on the 2nd floor of Wachman Hall. AccessNet accounts are required for access to Blackboard.

Homework exercises will require use of a Unix or Linux system (although a Windows Xp/Windows 7 system could be used). Programs may be written in C, C++, as the systems services and thread libraries are probably more readily available in these languages). Since most example programs and texts use the C language, it is recommended that students become familiar with programming in C. Students will be required to learn to use the system services available in Unix and Linux, and will have to use multiple processes and threads, shared memory, pipes, mailboxes, event flags, 'shared libraries', etc. These services and features should be learned and experienced immediately at the start of the semester, since all homework assignments will require them. Students may use any of these systems available to them, as noted above. It is recommended that students use one of the Unix programming texts, in particular the text by Gray).

Students may be assigned tasks of developing questions and answering questions developed by other students for each lecture. These question/answer dialogues will be via the class discussion and/or a discussion board (see below).

Students are responsible for taking all quizes 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.

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

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 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 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.

Students who miss the final exam and have not made alternative arrangements with me before I turn in grades, will receive a grade of F.

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 will be rectified.

Resources

We will use the Blackboard course management system in this course (http://tuportal.temple.edu or directly through the TUPortal). In order to access Blackboard you must have an Temple AccessNet account and be registered for the course. Students registered for the course will be automatically enrolled in the CIS 5512 Blackboard course. But this can only happen if you registered for the course. Until you have such an account, you cannot access the course materials in Blackboard.

Your AccessNet account also gives you login capability for Astro. Astro is multi-CPU system and has all of the necessary programming resources for your assignments. All materials for the course will be available from Blackboard. 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. The discussion list and Blackboard are accessed through a web browser. When you select the '5512 Fall 2014 Operating Systems' course in Blackboard, you can access the discussion board from the button on the left side of the panel.
All course documents, lecture notes, problem assignments, announcements, etc. will be available through Blackboard.

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

Additional Textbook References

- '**Operating Systems in Depth: Design and Programming**', Thomas Doeppner, 1st edition 2010, Wiley
- '**Operating Systems', Deitel, etal, 3rd Edition, Pearson Education, 2004**
- '**Operating Systems', Gary Nutt, 3rd Edition, Addison-Wesley, 2004**
- **Operating Systems: Concurrent and Distributed Software Design**, Bacon & Harris, Addison Wesley, 2003
- **'Concurrent Systems', Bacon, 2nd Edition Addison-Wesley, 1998**
- 'Operating Systems', 2nd ed., W. Stallings, Prentice Hall, 2001
- **'Concurrent Programming', G. Andrews, Benjamin/Cummings, 1991 [This text is the basis for many of the lectures on concurrency.]**
- 'Practical Unix Programming - A Guide to Concurrency, Communication and Multithreading', Robbins and Robbins, Prentice-Hall, 1996
- **'Interprocess Communications in Unix, The Nooks and Crannies', 2nd ed., Gray, Prentice Hall, 1998**
- 'Unix Network Programming - Interprocess Communications, Vol 2, 2nd Ed., Prentice Hall, 1999
- 'Multithreaded Programming with PThreads', Lewis and Berg, Prentice Hall, 1998
- 'Programming with PThreads', Kleiman, Shah and Smaalders, Prentice Hall, 1996
- 'Programming Under Mach', J. Boykin, et al, Addison Wesley, 1993
- **'Advanced Programming in the Unix Environment', Stevens, Addison Wesley, 1992**
- 'UNIX Network Programming, Volume 1: Networking APIs - Sockets and XTI, 2/e', Stevens, Prentice Hall, 1998
- 'Unix Systems for Modern Architectures', C. Schimmel, Addison-Wesley, 1994

- 'Modern Operating Systems', A. Tannenbaum, Prentice-Hall, 1992
- **'The Logical Design of Operating Systems', Bic and Shaw, Prentice Hall, 1988**
- 'Operating Systems: Design and Implementation 2/e', Tanenbaum & Woodhull, Prentice Hall, 1998
- 'An Introduction to Operating Systems', H. Deitel, (Second edition), Addison-Wesley, 1990
- 'Principles of Concurrent Programming', M.Ben-Ari, Prentice Hall, 1982
- 'Operating Systems Concepts' (3rd Ed.), Peterson and Silberschatz, Addison Wesley
- 'An Operating System Vade Mecum'(2e), R. Finkel, Prentice Hall, 1988
- 'Operating System Design, The XINU Approach', D. Comer, Prentice Hall, 1984
- 'Operating Systems, Advanced Concepts', Maekawa, Oldehoeft and Oldehoeft, Benj. Cummings, 1987
- 'Operating System Elements', P. Calingaert, Prentice Hall, 1982
- 'Fundamentals of Operating Systems', A. Lister, Springer Verlag, 1979
- 'VAX/VMS Internals and Data Structures', Kenah and Bate, Digital Press, 1984

Last Modified: 25 August 2014