CS 502 Operating Systems
Craig E. Wills
Monday, September 8, 2003

WPI, Fall 2003
Syllabus

Instructor

Craig E. Wills, FL-236, cew@cs.wpi.edu. Office hours: 9:00–10:00 Tuesday, 2:00–3:30 Thursday, any time for short questions. Electronic mail is the most effective method to contact me.

Course Web Page

Copies of all handouts, assignments, notes and old exams will be posted as appropriate on the course Web page. The address for it is `http://www.cs.wpi.edu/~cs502/f03/`.

Purpose

This is a graduate-level course in the design of operating systems. It covers principles, design decisions, design techniques, policies, and mechanisms. The course focuses on the design of general-purpose multiprogramming systems and covers processes, resource allocation, concurrency, memory management, time management, device control, synchronization and mutual exclusion. The goals are 1) to acquaint students with the components of a structured operating system, 2) to show students abstractions underlying design choices, tradeoffs, and their consequences, 3) to give students "hands on" experience with operating systems and related applications, and 4) to introduce students to research literature.

Prerequisites

Working knowledge of common data structures such as stacks, queues, and linked lists. Strong programming background in a high-level programming language such as C/C++. An interest in learning about operating systems.

Text Books

Required:

Modern Operating Systems. Andrew S. Tanenbaum, Prentice Hall, 2nd edition, 2001.

Reference:

Operating Systems Concepts. A. Silberschatz, P.B. Galvin and G. Gagne, John Wiley & Sons, 6th edition, 2002. A good, general supplement to the text.

Grading Policy

Final grades will be computed as follows:
Midterm Exam: 30%;

Final Exam: 30%;
Homework, projects, quizzes, and class participation: 40%.

Grading policy for each project and homework will be provided at the time of the assignment. Final grades will reflect the extent to which you have demonstrated understanding of the material, and completed the assigned projects. No incomplete grades will be assigned unless there exist exceptional, extenuating circumstances.

- **Programming Assignments**

  There will be 3-5 programming assignments. Assignments will involve programming in C/C++ on Unix systems. They will be done individually by each student. Students are assumed to be competent in a high-level programming language such as C or C++. System calls and other aspects of Unix will be introduced as the course progresses and programming projects are assigned.

- **Exams and Quizzes**

  There will be two in-class exams (including a final exam during the last week), plus the possibility of pop quizzes for which no advance notice will be provided. Exams will be closed book, closed notes.

- **Written Homeworks**

  There is a possibility of written homework assignments. Written assignments consist of problems from the book, made up problems, or readings from literature.

Late Policy

Each homework and programming assignment will be given a point value when it is handed out. The point value indicates the weight of the assignment relative to the other assignments. Late programs and homeworks will be be penalized 5% of total assignment value per day (with the weekend counting as one day) or partial day, and no assignments will be accepted after seven days beyond the due date. All programs and written homeworks are due at the *start* of class on the due date. Homeworks and programs turned in after the start of class will be counted late. Projects will be submitted as directed in class. Exceptions to these rules can be made only *a priori*. Finally, no assignments will be accepted after Wednesday, December 17 to allow sufficient time for grading.

Academic Honesty

Unless explicitly noted, all work is to be done on an individual basis. You are encouraged to talk with others about ideas and material in the course, particularly in preparing for exams. However all work, in the form or code or answers to problems, you submit for grading must be your work. Misrepresentation of the work of another as one's own submitted work is a violation of academic honesty. Aiding someone else to commit an act of academic dishonesty is also a violation. Submitting individually-assigned work that was jointly done with another person is a violation of academic honesty.

Any violation of the WPI's guidelines for academic honesty will result in no credit for the course and referral to the Student Affairs Office. More information on definitions, responsibilities and procedures regarding the WPI academic honesty policy can be found at `http://www.wpi.edu/Pubs/Policies/Judicial/sect5.html`.

Schedule

The following is a tentative outline of the material that will be covered in this course. All references to chapters are from Tanenbaum's book. Not all sections will be covered from each chapter and the lectures will be supplemented with material from other sources.

week 1: 9-8. Course introduction and overview. Introduction, Chapter 1. Process Management 2.1.

week 2: 9-15. Interprocess Communication and Coordination 2.3. 2.4

week 3: 9-22. Process Scheduling 2.5. Unix IPC.

week 4: 9-29. Threads 2.2. Deadlock, Chapter 3.

week 5: 10-6. Memory Management 4.1. Virtual Memory 4.3

week 6: 10-13. Virtual Memory (cont.). Paging 4.4, 4.5, 4.6, 4.7. Segmentation 4.8.

week 7: 10-20. Midterm Exam.

week 8: 10-27. No class.

week 9: 11-3. I/O Devices, Chapter 5.

week 10: 11-10. I/O Devices (cont.), File Systems, Chapter 6.

week 11: 11-17. File Systems, (cont.).

week 12: 11-24. Protection/Security, Chapter 9.

week 13: 12-1. Operating Systems Design, Chapter 12.

week 14: 12-8. Miscellaneous Topics.

week 15: 12-15. Final exam.