

COMP 530 — INTRODUCTION TO OPERATING SYSTEMS

Fall 2007

MW 2:00-3:15, Sitterson 115

Kevin Jeffay

jeffay@cs.unc.edu

<http://www.cs.unc.edu/~jeffay/courses/comp530>

August 22, 2007

Course Overview

An operating system is an essential part of almost all computer systems. In fact, most consumer appliances that contain a processor have an operating system inside. (Your dishwasher really does have an operating system somewhere inside it!)

In this course we primarily study general purpose, *time-shared* operating systems. In this context the operating system is the software system that provides the interface between users, their applications, and the underlying hardware. The purpose of this course is to introduce some of the fundamental concepts in the design of a time-shared operating system. These include:

- Processes and inter-process communication and synchronization
- CPU scheduling
- Memory management and virtual memory
- Secondary storage management
- File systems
- Deadlock detection and prevention
- Distributed operating systems and services
- Security and authentication*

Prerequisites

Prerequisites for this course are COMP 410/121 (Data Structures — including the “UNIX immersion” labs) and COMP 411/120 (Machine Organization), or permission of the instructor. *Both of these courses are hard prerequisites (i.e., you will not survive in this course if you have not had either course).* In addition a working knowledge of the Java programming language is assumed.

* Time permitting.

Course Materials

The textbook for this course is essentially any recent (post 2000) edition of any of the texts written by Silberschatz and Galvin, and Gagne and published by John Wiley & Sons. These include:

- *Operating Systems Concepts*, 7th edition, 2005,
- *Operating Systems Concepts with Java*, 6th edition, 2004.
- *Applied Operating Systems Concepts — Windows XP Update*, first edition, 2003 (the non-“Windows XP Update” edition is also fine).

We will attempt to cover the majority of chapters 1-12 (of the 7th edition, which also happens to be more or less chapters 1-12 in the other versions). Time permitting we’ll cover some of the advanced material and case studies at the back of the book.

This course has a “laboratory” component. You will be asked write several programs that illustrate the main concepts of concurrent programming and process management. These programs will be written in Java and possibly in C⁺⁺. The Java programs will be written on a UNIX machine. *By listing COMP 410/121 as a prerequisite I assume you are familiar with at least C⁺⁺ and that you know how to edit, compile, and execute a program on UNIX.*

Knowledge of the Java programming language is assumed and a Java reference manual is strongly recommended. If you do not already own a Java reference text, two good Java references are:

- *The Java Programming Language*, third edition, by Ken Arnold, James Gosling, and David Holmes, Addison Wesley, 2000,
- *Java in a Nutshell*, third edition, by David Flanagan, O’Reilly & Associates, 1999.

A supplementary (*i.e.*, optional) text that covers UNIX fundamentals is:

- *Learning the Unix Operating System*, 5th edition, by Jerry Peek, Grace Todino-Gonguet, and John Strang, O’Reilly & Associates, 2001.

In addition to these texts, I will hand out a set of lecture notes at each class. These notes are purposely incomplete and are meant to simply be a note-taking aid. If you miss class, copies of lecture notes, as well as copies of all class handouts, assignments, *etc.*, can be found on the COMP 530 home page on the Web at <http://www.cs.unc.edu/~jeffay/courses/comp530> (also linked from the Departmental home page at <http://www.cs.unc.edu/Courses/>).

Grading

Your grade in this class will based on:¹

- homework and programming assignments (approximately 40%),
- 2 midterm examinations (approximately 30%),
- final examination (approximately 30%), and
- class participation.

A student’s level of participation in class may be used to either adjust the final course grade up by a “+” (*e.g.*, from a B to a B+) or down by a “-” (*e.g.*, from a C to a C-).

¹ The percentages listed are only approximate and are subject to change (by no more than 10%).

Homework will be assigned approximately *every* two to three weeks. You can expect to always have an outstanding homework assignment throughout the semester. Assignments will be either laboratory (programming) exercises or written word problems. All assignments will be collected at the *start of class* on the day on which they are due. Written homework will be collected by the instructor; programming assignments will be submitted via electronic mail directly to the TA.

All homework submitted after its deadline is considered late. Assignments that are received within 24 hours after the original deadline are considered to be “one day late,” within 48 hours, “two days late,” *etc.* A late homework assignment will be accepted without penalty if the following conditions are met:

- The total “lateness” of all homework assignments received to date (including the current assignment) does not exceed 4 days.
- The student does not miss class on the day the assignment is due. Exceptions to this requirement must be approved by the instructor in advance.

If either condition is not met then the student will receive a failing grade on the assignment.

Late programs should be mailed to the TA. Late written assignments must be *hand delivered* to either the instructor, the TA, or Ms. Tammy Pike at the Front Desk of Sitterson Hall Mann. In the case of late written assignments, the lateness of the assignment will be assessed based on when the assignment is *received* and *not* on when it is submitted.

Students who submit all homework assignments on time will receive a 10% homework grade bonus at the end of the semester.

Office Hours

Office hours for the instructor will be by appointment, however, a generally good time to meet would be on MW from 3:30-4:30. The office hours for the TA will be announced during the second week of class.

Course Conduct and the Honor Code

Students are *encouraged* to work together on written homework assignments. *Acceptable collaboration* on written homework assignments includes:

- discussing the assigned problems to understand their meaning,
- discussing possible approaches to assigned problems,

In all cases you must *explicitly* acknowledge any and all substantive help received from other individuals during the course of the preparation of your homework solution. That is, if you collaborate with other individuals then you *must* include an explicit acknowledgment in your homework solution of the persons from whom you received aid. You should include the acknowledgement with your Honor Code pledge. Acknowledging others, if done properly, will not adversely effect your grade.

Unacceptable collaboration on written homework includes:

- Copying (verbatim or substantially similar use) of physical papers or computer files,²
- Working with individuals who are not presently members of the course, and
- Submission of solutions that are jointly authored, or authored either wholly or in part by other individuals.

² This includes computer files that are copied and then edited and/or reformatted.

The general rule to be followed is that the strategy and approach of solutions may be developed jointly but *all* actual solutions (*i.e.*, the final solution) must be *constructed* and *written up* individually. Work done jointly should not be done in sufficient detail as to make it a final solution. For example, solutions may *sketched* out jointly, however each student must construct the final form of their solution individually and write-up their own solution.

For programming assignments *absolutely no code may be shared between students*. If you copy or transcribe one or more lines of code from any other student (including students/persons not in this course) you are in violation of the Honor Code.

Unacceptable collaboration will be considered a violation of the UNC Honor Code and will be reported to the Student Attorney General.

Should questions arise the course of working on a problem please feel free to immediately contact the instructor either by telephone, electronic mail, or by an office visit. In principle, if you work with others in good faith and are honest and generous with your attributions of credit you will have no problems.

Course Schedule

Date	Topic	Homework
Aug. 22	Course introduction & mechanics	HW 1 assigned
Aug. 27, 29	History of OS	HW 2 assigned
Sep. 3	Labor Day Holiday	
Sep. 5	Processes	
Sept. 10	Process Implementations & Scheduling	HW 3 assigned
Sept. 12	Process Implementations & Scheduling	
Sept. 17	CPU Scheduling Policies	
Sept. 19	Process Coordination & Synchronization	
Sept. 24, 26	Semaphores	HW 4 assigned
Oct. 1, 3	Monitors	HW 5 assigned
Oct. 8, 10	Message Passing	
Oct. 15	Memory Management Basics	
Oct. 17, 22	Virtual Memory	
Oct. 24	Midterm Exam (Tentative)	
Oct. 29	Midterm Exam Discussion	HW6 assigned
Oct. 31, Nov 5	Page Replacement Algorithms	
Nov. 7	Load Control and Virtual Memory Design Issues	
Nov. 12	Segmentation and Shared Memory	HW 7 assigned
Nov. 14, 19	Deadlock	
Nov. 21	Thanksgiving break	
Nov. 26	Secondary Storage Management	HW 8 assigned
Nov. 28	Endterm Exam	
Dec. 3	Secondary Storage Performance	
Dec. 5	Endterm exam review	
Fri. Dec. 14 12 PM	Final Exam	