CS 450 Operating Systems,
Course Overview
Fall 2011, 251 ESB

(Sect: 01 CRN: 84055, 3 credits, 3 hr lectures per week)

Class: 251 ESB

Instructor:	**Camille Hayhurst**
Email:	**cahayhurst@mail.wvu.edu** or **zerie@aol.com**
Web page:	**www.csee.wvu.edu/~hayhurst**
Office:	759 Engineering Sciences Building
Phone:	**(304) 293-0405 extension 39693**
Office Hours:	**M W F – 9:00 to 11:50 and other times by appointment**

Graduate Assistant: **Forrest Desjardins**

Laboratory: **This course does not include closed lab sessions**. **Friday of each week, will be a lab session, not lecture.** For this project you will need a PC running either the Windows or Linux operating system, and an ANSI C compiler. You must have Administrator access to the PC. A USB port is also required. You are encouraged to use your own computers where possible. Some PC's running DOS, and equipped with the MPX software and Turbo C are available in room 851, and may be used if necessary.

Prerequisites: CS 350 (or equivalent). This course requires that students have at least an introductory knowledge of the C programming language (**NOT just C++ or Java)**.. **Prerequisites will be enforced. Students who are not comfortable writing C programs using pointers and multiple source files will have difficulty with course projects.**

Required Text: The text and project notes listed below will be made available via the web. These are copyrighted materials, and access will be password protected.

1. CS 450 Text: selections from "A Practical Approach to Operating Systems", by J.D. Mooney and M.B. Lane.
2. CS 450 Project Manual, by J.D. Mooney, and M.G. Lane

These can be found on Dr. Mooney's web site: www.csee.wvu.edu/~jdm/classes/cs450

Expected Learning Outcomes:
1. Demonstrate a working knowledge of basic OS concepts, especially process management, I/O management, and the user interface
2. Briefly discuss several advanced OS concepts including file systems and virtual memory.
3. Explain Highlights of OS history and its relation to present-day OSs
4. Carry out the development of a significant OS implementation project using the C language
5. Work effectively as a member of a software development project group

Outline
Tentative schedule

Week Of	TOPIC	READING
August 22	Class introduction	Chapter 1
	OS overview and a little history	
	PROJECT SESSION: INTRODUCTION TO MPX & Turbo C, R1 introduced 8/26	**Module 1 assigned**
August 29	History of operating systems	Chapter 2
	Project session, Continue R1, Interface Design Chapter 3 9/2	
September 5	Labor Day -- Process Management	Chapter 4
	Project Session: Module 2 introduced 9/9	**Module 2 assigned**
September 12	Process Management -- continued	Chapter 4
	Project session: R1 Due 9/16	**Module R1 due,**
September 19	Process Scheduling	Chapter 5
	Project session – Work Day for Project, R2 demos 9/23 – Demo of project progress	
September 26	Interrupts	Chapter 6
	Project session: Modules 3 & 4 introduced 9/30	**Module R3 & R4 assigned,**
October 3	*Midterm Exam—Oct 5th*	**Study hard**
	Project session –R2 Due 10/7	**Module R2 due**
October 10	Basic memory Management	Chapter 7
	Project session, Module R5 introduced 10/14	**Module R5 assigned**
October 17	Time Management	Chapter 8
	Project session -- Work Day, continue R5, Demo R3/R4 – Question and Answers 10/21, Demo project progress	
October 24	I/O Devices	Chapter 9
	Project session, Module R3 & R4 due 10/28 Last Day to drop classes Oct 28th	**Modules R3 & R4 due**
October 31	I/O device management	Chapter 10
	Project session – Module R6 introduced 11/4	**Module R6 assigned**
November 7	OS structure	Chapter 11
	Project session: Module R5 due 11/11	**Module R5 due**
November 14	File systems	Chapter 12
	Project session: Work Day for Project 11/18 Preliminary R6 code due	**Preliminary R6 code due**
November 21	**Thanksgiving Recess**	REST UP
November 28	**Virtual Memory**	Chapter 13
	Project help session, R6 demos, 12/2	
December 5	**Dead week, Review, project presentations and orals NO CLASS Wednesday and Friday, Final Review Monday**	**Dead Week - Module R6 due, PROJECT DEMOS AND ORAL EXAMS!!!!!**

December 12	**FINAL EXAM -- Monday 12/12**	**3 – 5 pm !!!!!**

There will be no class the Wednesday and Friday of dead week; the time will be used for project work and oral exams. Projects MUST be finished, and oral exams scheduled <u>by the end of dead week.</u> PROJECTS MUST BE FINISHED BEFORE FINALS WEEK!!!!!!!!!!!

<u>Rules of Operation:</u>

Attendance: Class attendance is strongly recommended. Students are responsible for all material covered in the course, keeping track of assignments and examination dates. Attendance on lab days is required. Three unapproved absences will result in a 5% reduction of your individual project average.

MPX Group Project: The MPX project is a very important component of this course. During the course students will work in groups of 3-4 students to develop a **primitive** multitasking operating system through a series of deliverables. The components of the project grade are assigned on a group basis and all members of the group receive the same grade. Your instructor reserves the right to make INDIVIDUAL assignments, which will add additional detail to the project and will contribute to an individual's project grade. **You instructor also reserves the right to adjust an individuals project grade either up or down based on their participation within their group. PICK YOUR GROUP CAREFULLY.**

The final deliverable will be due during dead week when each group will participate in an **<u>oral exam and project demonstration</u>**. Individual grades for the final Module, will be based on each persons **<u>PERFORMANCE</u>** during the oral examination, and your **<u>software's performance</u>** during the project demonstration.

The project is intended both to strengthen your understanding of OS concepts, and to provide an experience in developing system software as a part of a project group. Since an operating system must interact directly with hardware resources, you will also increase your familiarity with basic hardware structures such as interrupts and simple I/O devices. The course will also develop your proficiency in the C Language, commonly used in systems programming. Groups will also be expected to produce both a programmers and users guide for their evolving system.

Modules will be assigned approximately every three weeks, and will overlap.

IF you substantially fail to fulfill your responsibilities in your project group, or do not participate in the oral examination, you will receive a zero for your project score and you will fail the course.

<u>Exams</u>: There will be a midterm and final exam.

Exam Absences: There are no scheduled make-up exams. Absence from examinations must be arranged with the instructor prior to the examination. However, consistent with WVU guidelines, students absent from regularly scheduled examinations because of authorized University activities will have the opportunity to take them at an alternate time.

Grading: Grades for the course will be based on the point assignments given below. At the end of the semester letter grades will be assigned based on the final average. The grading scale is also shown. This scale may be adjusted slightly (lowered) at the instructor's discretion, but not by much.

Midterm	25%
Final Exam	30%
Lecture Homework	5%
MPX Group Project:	40%

Grading Policy:

90 - 100%	**A**
80 - 91%	**B**
70 - 81%	**C**
60 - 71%	**D**
< 60%	**F**

A grade of I (incomplete) may be given only in cases of bonafide personal emergency. You are required to sign a contract explaining the nature of the emergency and agreeing to complete all necessary work within 60 days.

Academic Honesty: All work, on exams to be done on an individual basis. Project work must be the original work of the project group, not borrowed from other groups in this class or previous classes. Evidence to the contrary will be regarded as academic dishonesty and will be dealt with accordance with the Lane Department of CSEE academic dishonesty policy. You are expected to abide by the WVU principles of academic honesty.

Other Policies:

Academic Integrity
The integrity of the classes offered by any academic institution solidifies the foundation of its mission and cannot be sacrificed to expediency, ignorance, or blatant fraud. Therefore, I will enforce rigorous standards of academic integrity in all aspects and assignments of this course. For the detailed policy of West Virginia University regarding the definitions of acts considered to fall under academic dishonesty and possible ensuing sanctions, please see the Student Conduct Code at http://www.arc.wvu.edu/admissions/ integrity.html. Should you have any questions about possibly improper research citations or references, or any other activity that may be interpreted as an attempt at academic dishonesty, please see me *before* the assignment is due to discuss the matter.

Social Justice: West Virginia University is committed to social justice. I concur with that commitment and expect to foster a nurturing learning environment based upon open communication, mutual respect, and non-discrimination. Our University does not discriminate on the basis of race, sex, age, disability, veteran status, religion, sexual orientation, color, or national origin. Any suggestions as to how to further such a positive and open environment in this class will be appreciated and given serious consideration.

NOTE: If you are a learning, sensory or physically disabled student and feel that you need special assistance in regard to lectures, reading assignments or testing, please contact your instructor after class or during office hours. The first week of the semester is the best time to inform your instructor of any disability, and make appropriate arrangements with Disability Services (293-6700)