UCF
School of Electrical Engineering & Computer Science
COP 4600: Operating Systems
Fall 2005

Syllabus

Instructor: Euripides Montagne  Tele.: 823-2684  email: eurip@cs.ucf.edu

Lecture meetings:
TR 12:00 (noon) – 1:15 p.m. (ENG2 105)

Office hours:
MW 3:00 p.m. – 5:00 p.m. and TR 3:15 – 4:15 (CSB 239)

TA: Hao Cheng  Tele.: 823-3483  email: haocheng@cs.ucf.edu

Project office hours:
Wednesday 3:00 p.m. – 5:00 p.m. (CC1 202)

Course Outline: The goal of the course is to teach fundamentals concepts and design principles of operating systems.


Prerequisites:
• COP 3530C – Computer Science III.
• COP 3402C – System Concepts/Programming.
• Proficiency in C and Familiarity with UNIX.

If you have not satisfied all of the above prerequisites, you do not belong in this class and have little chance of passing.

Reference Guide:
The textbook for the course is: H. M. Deitel, P. Deitel, and D. Choffnes, “Operating Systems” 3rd Edition, Prentice Hall, 2004. We will cover Chapters 1-13. You are responsible for the material contained in all of those chapters, even if it is not discussed in class. Time permitting we may cover parts of Chapter 19.

Style of Class Meetings:
Class meetings will not consist of traditional lectures, with the instructor doing most of the talking and the student doing most of the listening. Rather, meetings will consist of discussions on each topic and the instructor will help guide the discussion by asking questions.

Grading Policy:
• (25%) Exam #1 – closed book, closed notes exam given in class.
• (25%) Exam #2 – closed book, closed notes exam given in class.
• (25%) Final Exam – closed book, closed notes comprehensive exam given during final exam week.
   Note: You must score at least 60% on this exam to pass the course.
• (20%) Programming project – a large, multi-part simulation of a multiprogramming operating system. Written in C on a UNIX system, this project is not easy, but can be done in the time allotted. The grade for this project will be divided between your C code, one or more demonstrations of your project, your documentation and quizzes given on selected topics from the project.
• (5%) Concurrent programming assignments.

Note: Any academic dishonesty (including, but not limited to, Cheating, copying and/or plagiarism) with respect to any exam or assignment in this class will result in a grade of \textit{F}, following by the usual procedures for dealing with such behavior, as describe in the \textit{UCF Golden Rule: a handbook for students}.

\textbf{The Semester Plan:} Tentative.
Aug. 25 - Computer System Structure.
Aug. 30 - Interrupt Handling.
Sept. 01 - Interrupt Handling
Sept. 06 - Operating System Structure.
Sept. 08 - Processes and Threads.
Sept. 13 - Process Synchronization.
Sept. 15 - Process Synchronization.
Sept. 20 - Process Scheduling.
Sept. 22 - System Performance Evaluation
Sept. 27 - Review
Sept. 29 - \textbf{First Midterm Exam}.
Oct. 04 - System Performance Evaluation
Oct. 06 - Memory Management
Oct. 11 - Memory Management
Oct. 13 - Memory Management
Oct. 18 - Virtual Memory
Oct. 20 - Virtual Memory
Oct. 25 - Review
Oct. 27 - \textbf{Second Midterm Exam}.
Nov. 01 - I/O structure
Nov. 03 - Disk Scheduling
Nov. 08 - Performance Evaluation I/O system
Nov. 10 - Resource Allocation and Deadlock
Nov. 15 - Resource Allocation and Deadlock
Nov. 17 - File System
Nov. 22 - File System
Nov. 29 - Review
Dec. 01 - \textbf{Final Exam}

\textbf{COP 4600 Programming Project (Fall 2005)}

This project is divided into 3 parts to make it more manageable. Details will be given out well before the due dates for each part (the parts of the project are called \textit{objectives}). This project must be written in \textit{C} (not \textit{C++}) on a UNIX system. The standard for this class will be the Sun Sparc system in the main computer lab called Olympus. You are welcome to write and test code on some other system, if you wish, but it will be graded on Olympus and if it does not work there, it does not work. You will be given an Olympus account and, once the project has begun, should check your e-mail regularly for updates.

To pass this course, you \textbf{must} successfully complete objectives 1, 2 and 3. \textbf{No exceptions.}

Each objective will have a due date and points will be subtracted for submission after that date. Also, after each due date some evaluations of you progress will be made. This may include a walk through of your code with the instructor or grader, a quiz on the objectives that was just completed (including questions about code, data structures and/or algorithms) or a short, written description of the purpose and implementation of the objective. Details will be handed out with each objective and I reserve the right to change the method of evaluation at any time.
In general, this project will give you a better understanding of the data structures and control flow of a multiprogramming operating system and also provide you with experience in developing and debugging a complex software project.

**Let's make this clear:** when working on the project, you are allowed to talk to other students about programming concepts, C syntax and general solutions to problems (algorithms or questions about the project instructions), but you are not allowed to share, exchange or copy code. Both the source and the recipient of any exchange of code are equally at fault.

**Important Dates:**
- Classes Begin **August 22nd**.
- Withdrawal Deadline is **October 14th**.
- Classes End **December 3rd**.
- Summer Holidays are:
  - Labor Day **Sept. 5**.
  - Veteran’s **November 11**.
  - Thanksgiving **Nov. 24 – 26**.