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

Lecture meetings: TR 4:30 p.m. – 5:45 p.m.(CL1 121)

Office hours: MW from 10:00 a.m. to 12:00 a.m. (HEC 217)
              TR from 2:00 p.m. to 4:00 p.m. (HEC 217)

TA: Merrill McKeen    Tele.: (407) 823-4733   email: merrill@cs.ucf.edu
Office hours: TR From 1:30 p.m. to 2:30 p.m. (HEC 254)

TA: Lisa Batsch-Smith    Tele.: (321) 377-8359  email: lbatsch@gmail.com
Office hours: MW from 11 a.m. to 12 a.m. (HEC 242)

TA: Jianyong Dai     Tele.: (407) 823-2524i   email: daijy@cs.ucf.edu
Office hours: T from 2:00 p.m. to 4:00 p.m. (HEC 313)

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


Prerequisites:
- COP 3503 – Computer Science II.
- COP 3402C – System Concepts/Programming.
- COT 3960 - Foundation Exam.
- 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.


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:
- (20%) Exam #1 – closed book, closed notes exam given in class.
- (20%) 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.
- (30%) 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 F, following by the usual  
procedures for dealing with such behavior, as describe in the UCF Golden Rule: a handbook for students.

The Semester Plan: Tentative.
Aug. 21st  Operating system fundamentals.
Aug. 23rd  Computer system structure.
Aug. 28th  Interrupt handling.
Aug. 30th  Interrupt handling
Sept. 4th  Operating system structure.
Sept. 6th  Processes and threads.
Sept. 11th  Process synchronization.
Sept. 13th  Concurrent programming
Sept. 18th  Process scheduling
Sept. 20th  I/O subsystem
Sept. 25th  Review
Sept. 27th  First Midterm Exam.
Oct. 2nd   Device handlers and I/O request handling
Oct. 4th   Disk scheduling
Oct. 9th   System Performance Evaluation
Oct. 11th  System Performance Evaluation
Oct. 16th  Memory organization
Oct. 18th  Virtual memory
Oct. 23rd  Review
Oct. 25th  Second Midterm Exam.
Oct. 30th  Virtual memory
Nov. 1st   Resource allocation and deadlock
Nov. 6th   Resource Allocation and Deadlock
Nov. 8th   Protection mechanisms
Nov. 15th  Protection mechanisms
Nov. 20th  File system
Nov. 29th  Review
Dec. 6th   Final Exam – 4pm – 6:50pm
This project is divided into 4 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 objectives). This project must be written in C (not 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 must successfully complete objectives 1, 2 and 3. No exceptions.

Each objective will have a due date and points will be subtracted for submission after that date (up to 5 days late, 20% off; more than five days late, you get “0” for that objective). Also, after each due date some evaluations of your 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.

Lets 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: Monday August 20.
- First Midterm exam: Thursday Sept 27.
- Withdrawal deadline: Oct. 2.
- Second Midterm exam: Thursday Nov 1.
- Classes end: Monday Dec 3.
- Final exam: Dec. 6th, From 4:00 p.m. to 6:50 p.m.
- Fall Holidays are:
  - Labor Day: Monday, Sept. 3rd.
  - Veteran’s Day: Monday, Nov. 12th.