COURSE SYLLABUS

*****************************************************************************
YEAR COURSE OFFERED:  2013
SEMESTER COURSE OFFERED:  Fall
DEPARTMENT:  Computer Science
COURSE NUMBER:  CS4375
NAME OF COURSE:  Theory of Operating Systems
NAME OF INSTRUCTOR:  Dr. Eric Freudenthal
*****************************************************************************

The information contained in this class syllabus is subject to change without notice. Students are expected to be aware of any additional course policies presented by the instructor during the course.

*****************************************************************************

Learning Objectives

On successful completion of this course, students will

1. be able to apply the following in new situations:
   a. operating system objectives and functions
   b. process definition/description and control/management
   c. threads, symmetric multiprocessing, microkernels
   d. mutual exclusion and synchronization (software and hardware approaches)—semaphores, monitors, message passing, readers/writers problem
   e. concurrency: deadlock and starvation—principles of deadlock, deadlock prevention, avoidance, and detection
   f. dining philosophers problem
   g. memory management—paging, segmentation
   h. virtual memory—hardware and control structures
   i. scheduling algorithms

2. be able to apply:
   a. file management (file organization, directories, and sharing), record blocking, secondary storage management
   b. multiprocessor and real-time scheduling
   c. I/O management and disk scheduling

3. have been introduced to:
   a. Windows NT operating system
   b. UNIX operating system
   c. distributed processing, client/server, an clusters

Page 1 of 5
COURSE SYLLABUS

d. distributed process management

Major Assignments/Exams

- Exams and quizzes
  - Types
    - Final exam: date and time are specified by the university.
    - Quests (Frequent Quizzes in lieu of Tests)
      - Generally unannounced, at least one every two weeks, generally focus on recently studied topics, but may contain topics studied earlier
      - Short: Generally 10-20 minutes
      - May contained timed portions on topics students are instructed to memorize in lecture (for example, editor commands, binary encoding of hexadecimal digits, and powers of 2 up to $2^{10}$).
      - May not be "made up"
  - Multiple skills may be measured by the same test question. For example, if a question requires students to construct a finite state machine and implement it in C, assembly, and machine language, each of the following may be independently assessed
    - Appropriateness and completeness of state machine design and representation
    - Quality and completeness of C code (algorithm, variable and function naming, understandability)
    - Translation of various types of C statements into assembly language (all scored separately)
    - Quality and completeness of assembly language code (including appropriate use of addressing modes, array indexing, operand order)
    - Accuracy of machine language representation (opcode, addressing modes, relative branch offsets...)
    - Ability to communicate clearly.
  - To the extent that it is practical, useful competency (generally a binary value) for distinct skills will be assessed independently
    - Thus, if an exam with four questions measures ten skills, ten measurements will be computed.
    - A single problem may provide opportunity to demonstrate all or most of the skills being measured.
    - A skill not demonstrated by any answer provided by a student due to not providing complete and correct answers to all problems will be assessed as not meeting the threshold of useful competency.
    - It is possible that all skills measured by a particular test will be demonstrated in multiple problems.
    - Tests are designed to require substantially less time than allotted. Therefore not completing an exam within the allotted time may be an
COURSE SYLLABUS

indication of weak understanding worthy of discussion with the instructor.

- Notes on test-taking strategy
  - If short of time - it is not generally advantageous to partially answer multiple questions in a manner that repeatedly demonstrates the same skills.

- The grade for an exam or quiz will correspond to the fraction of skills in which useful competency is demonstrated. Generally
  - 100% corresponds to an A+ (4.3 on a 4-pt scale)
  - 50% corresponds to an F (zero on a 4-pt scale)
  - Conversion back to 4-pt scale: \( \text{Grade} = (f - 0.5) \times 8.6 \) where \( f \) is the fraction described above

- As in life, all exams are cumulative.
  - If test (midterm/quiz/final) \( T_1 \) occurs before test \( T_2 \), a skill measured in \( T_1 \) may also be assessed in \( T_2 \).

- Overall test grade: computed by tabulated skills
  - Each test (midterm/quiz/final) measures some set of skills
  - Many skills may be measured at multiple times during the semester
  - Aggregate test grade is computed based upon skills demonstrated during the semester
    - Later measurements dominate earlier
    - Typically, if 3 or more measurements are taken for the same skill, the assessed level will be computed from the last three measurements.

- Assignments and labs
  - Intention:
    - Assignments and labs provide an opportunity for students to practice and explore concepts presented in class.
    - Students are expected to act professionally
      - By helping each other select and design problem-solving approaches
      - By reading whatever resources they find relevant
      - By attributing credit to any person or reference materials that substantively contributed to their solutions
      - By only submitting solutions they fully understand.
      - Professionalism includes honesty, clarity, and accuracy.
  - Types:
    - Some class sessions will conclude with an assignment due at the beginning of the next class session unless otherwise indicated.
    - Labs will be assigned during the lab course. Due in 1 week unless otherwise indicated.
  - Rules
    - Students must only submit solutions that fairly reflect their own understandings.
    - Solutions must clearly and fairly attribute credit to people and resources that contributed to their design or preparation.
COURSE SYLLABUS

- Descriptive text included with solutions must be composed by the student submitting it.
  - Implications
    - It is academic dishonesty for a student to submit a solution they cannot replicate individually or to not fairly credit their sources.

Grade Computation

- Lab and exam grades are computed separately.
  - Lab grades are averaged
  - Exam (and quiz) grades are aggregated by skill, and then averaged
  - Overall course grades are the minimum of lab and exam grades.

Required Reading


Recommended Reading

- Recommended by students (and by no way required): Android app "Programmer Mental Math" by Joel Jurix.

Accommodations for people with disabilities

If you have a disability and need classroom accommodations, please contact the Center for Accommodations and Support Services (CASS) at 747-5148, or by email to cass@utep.edu, or visit their office located in UTEP Union East, Room 106. For additional information, please visit the CASS website at www.sa.utep.edu/cass.

List of discussion/lecture topics

Note: detailed lecture notes are available on the course web site.

- Introduction
  - What is an operating system?
  - History
  - Operating systems for various devices
  - Review of computer architecture
  - Summary of OS concepts
  - System Calls
  - Operating System Structure
COURSE SYLLABUS

• Processes and Threads
  • Processes
  • Threads
  • Inter-process communication
  • Classical problems
  • Scheduling

• Synchronization (mostly deadlock prevention)
  • Resources
  • Deadlock (another reason why ostriches do not dominate the planet)
  • Detection of deadlock
  • Stopping systems from entering deadlock (avoidance)
  • Designing systems that cannot deadlock (prevention)

• Memory Management
  • Single address space (both mono- and multiprogramming)
  • better nomenclature than book
    • virtual memory
    • demand v. voluntary systems
    • paged v. segmented systems
  • Managing paged systems
    • exposes a eviction (replacement) problem
    • program behavior (locality)
    • algorithms
    • implementation
    • simulation
  • Managing segmented systems
    • exposes both a placement & eviction problem

• Input/Output
  • characteristics of i/o devices
  • how to structure drivers
  • disks
  • terminals, graphical devices
  • networked terminals
  • power mgmt

• File Systems
  • files
  • directories
  • implementation
  • special files (/dev, links, /proc)
  • example filesystems