CS 3080: OPERATING SYSTEMS

Semester Hours: 3.0  
Contact Hours: 3

Coordinator: Robert Dyer

Text: Operating System Concepts

Author(s): SILBERSHATZ GALVIN, GAGNE

Year: 2008

SPECIFIC COURSE INFORMATION

Catalog Description:

Features of modern multiprocessing operating systems. Threads and processes; resource management; scheduling, concurrency, and communication; virtual memory management; secondary storage management. Students cannot get credit for both CS 3080 and CS 3270. Prerequisite: Grade of C or better in CS 2020 and CS 2170 or CS 2190.

Course type: REQUIRED

SPECIFIC COURSE GOALS

- I can describe process scheduling algorithms, and compare their performance.
- I can use language primitives to manage threads and processes.
- I can describe concurrency issues and compare approaches to solving them.
- I can implement pseudo-code & actual code to solve certain synchronization problems.
- I can describe real and virtual memory management algorithms.
- I can derive the mapping between virtual and real addresses.
- I can describe certain scheduling algorithms for device management.

STUDENT OUTCOMES ADDRESSED BY THIS COURSE

- B.1 Analyze a given problem, and identify and define the computing requirements appropriate to its solution
- B.2 Use current techniques, skills, and tools in computing practice
• B.3 Apply mathematical foundations, algorithmic principles, and computer science theory as appropriate in modeling and solving real-world problems

LIST OF TOPICS COVERED

• Overview (~ 10%)
  o OS history and features
  o Process, user and kernel threads
  o Security

• Scheduling (~ 20%)
  o Process and thread management
  o Scheduling algorithms
  o Performance tradeoffs
  o Examples

• Concurrency (~ 20%)
  o Race condition
  o Mutual exclusion algorithms for processes and threads
  o Deadlock
  o Examples

• Communication (~ 15%)
  o Shared memory
  o Pipes and other paradigms
  o Examples

• Memory Management (~ 15%)
  o Real and virtual memory
  o Address Translation
  o Paging algorithms
  o Performance and examples

• Device Management (~ 10%)
  o Device interaction
- Buffer management
- Disk schedulers
- Platform Specifics (~ 10%)
  - Windows
  - Unix