Course number and name:	**CS 06395 Operating Systems**
Credits and contact hours:	3 credits / 3 contact hours
Course Coordinator:	Patrick McKee
Instructional Materials:	Modern Operating Systems, 4th, Andrew Tanenbaum and Herbert Bos, 2015.

Specific course information

Catalog description: The course concentrates on the design and functions of the operating systems of multi-user computers. Its topics include time sharing methods of memory allocation and protection, files, CPU scheduling, input-output management, interrupt handling, deadlocking and recovery and design principles. The course discusses one or more operating systems for small computers, such as UNIX.

Prerequisites: CS 04222 Data Structures and Algorithms and
CS 06205 Computer Organization

Type of Course: ☒ Required ☐ Elective ☐ Selected Elective

Educational objectives for the course

1. **context switching.** The student has been able to accurately explain the role of context switching in an operating system and how/when the operating system decides which process to switch to.
    - ABET (1) Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.
    - ABET (3) Communicate effectively in a variety of professional contexts.

2. **deadlocks**. The student has demonstrated an understanding on how to detect, prevent, and solve (using multiple methods) deadlocks that occur in an operating system.
    - ABET (1) Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.
    - ABET (3) Communicate effectively in a variety of professional contexts.

3. **memory management**. The student has explained multiple ways that an operating system can allocate an address space to a process and how virtual memory is managed via page eviction algorithms.
    - ABET (1) Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.
    - ABET (3) Communicate effectively in a variety of professional contexts.

4. **OS theory**. The students should have an understanding of operating system's theory and implementation.
    - ABET (2) Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline.

5. **hardware**. The students have described the hardware components of modern computing environments and their individual functions.
    - ABET (2) Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline.

6. **OS functionality**. The students have described the role and basic functions of an operating system, and how operating systems interact with hardware and software applications.
    - ABET (2) Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline.

7. **OS security**. The students have identified and described basic security issues of operating systems.
    - ABET (4) Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.

Required list of topics to be covered

1. Application processes and threads; Process/Thread Management
    a. Race conditions
    b. Semaphores
    c. Scheduling algorithms
    d. Privileged and non-privileged states
    e. Domain separation, process isolation, resource encapsulation, least privilege
    f. Inter-process communication
    g. Message passing systems
2. Memory (real, virtual, and management)
    a. Memory hierarchy
    b. Bitmaps
    c. Memblocks
    d. Heap allocation and freeing
    e. Stack allocation and freeing
    f. Virtual memory
    g. Page eviction algorithms
3. Files systems
    a. File construction
    b. Inodes
    c. FAT file systems
    d. Journaling systems
4. Input/output
    a. Hardware components
    b. Interrupt driven I/O
    c. Port-mapped I/O
    d. DMA systems

    e. Device independent OS layers
    f. Hardware clocks
5. Deadlocks
    a. Deadlock definition
    b. Modeling
    c. Detection and recovery
    d. Distributed system concepts
6. OS Programming
    a. Virtualization / hypervisors
    b. Creation and operation of a virtualized development environment
    c. Programming direct control of memory access
    d. Port-map programming
7. Security
    a. Fundamental security design principles as applied to an OS
    b. Access controls (models and mechanisms)
    c. Real-time operating systems
    d. Distributed OS architectures
    e. Buffer overflows
    f. Monitoring and logging systems software