Course number and name:	**CS 04390 Operating Systems**
Credits and contact hours:	3 credits. / 3 contact hours
Faculty Coordinator:	Patrick McKee
Text book, title, author, and year:	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

Specific goals 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 to which process to switch

    - ABET (c) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs

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 (b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution

    - ABET (c) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs

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 (c) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs

4. **OS theory**. The students should have an understanding of operating systems theory and implementation. They will understand OS internals to the level that they can design and implement significant architectural changes to an existing OS.

    - ABET (c) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs

5. **hardware**. The students have described the hardware components of modern computing environments and their individual functions.

    - ABET (c) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs

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 (c) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs

7. **OS security**. The students have identified and described basic security issues of operating systems.

    - ABET (c) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs

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. Privilege states
    g. Inter-process communications

2. Memory (real, virtual, and management)
    a. Bitmaps
    b. Memblocks
    c. Virtual memory
    d. Page eviction algorithms
3. Files systems
    a. inode
    b. FAT
    c. Journaling systems
    d. File construction
4. Input/output
    a. Device independent OS layer
    b. DMA systems
    c. Programmed I/O
    d. Synchronous vs. asynchronous
5. Deadlocks
    a. Detection
    b. Recovery
    c. Dynamic avoidance
    d. Concurrency and Synchronization, Deadlocks
    e. Complexity measures: number of processors
6. OS Programming
    a. Virtualization / hypervisors
    b. Creation and operation of virtualization technology
    c. Clear interface semantics
    d. Programming to operating systems internal interfaces
    e. Resource optimization
    f. Resource management
    g. Run time overhead minimization
    h. Programming direct control of memory access and flow control
7. Security
    a. Fundamental security design principles as applied to an OS
    b. Access controls (models and mechanisms)
    c. Real-time operating systems/security issues
    d. Distributed OS architectures & security issues
    e. Buffer overflows
    f. Security concerns in systems software
    g. Monitoring and logging systems software