CS 520 Introduction to Operating Systems Course Outcomes

Each course outcome is followed in parentheses by the Program Outcome to which it relates.

- Explain how memory is allocated among multiple resources. ()

- Explain how the data structures of the UNIX file system serve to map file names to blocks of a storage device. ()

- Explain the value and dangers of in-memory caching of file system disk blocks; describe the algorithms for checking block and file consistency. ()

- Explain how virtual memory is implemented with paging and segmentation. (*MS-CS C Depth,*)

- Explain in detail how a virtual address is translated into a physical address on at least one common hardware architecture. (*MS-CS C depth,*)

- Explain the value and drawbacks of "non-traditional" hardware architectures for supporting virtual memory; e.g., inverted page table and TLB-only. ()

- List causes of interrupts and traps and explain how an operating system, in concert with hardware support, services interrupts and system call traps. (*MS-CS C depth,*)

- Implement a C program that uses UNIX's file system interface. (*MS-CS A development,*)

- Analyze proofs of some classic concurrency protocols and devise similar protocols with some conditions changed. ()

- Apply semaphores to solving a wide range of multi-processing synchronization problems. ()

- Apply the deadlock detection algorithm to a given state of a system of processes and resources, and determine whether that state is safe. ()

- Develop discrete-event-based simulation programs using pseudo-random number generators and apply these programs to simulating process scheduling in operating systems. (*MS-CS A development,*)

- Explain the principles of OS security, basic types of attacks, and common countermeasures. ()