Computer Science 4302 – Operating Systems

Student Learning Outcomes

1. **Overview**
   The student will learn what operating systems are, what they do, and how they are designed and constructed. The student will be introduced to what the common features of an operating system are, what an operating system does for the user, and what it does for the computer-system operator.
2. **Process Management**
   The student will be introduced to the process concept and concurrency as the heart of modern operating systems. The student will cover methods for process scheduling, interprocess communication, process synchronization, and deadlock handling. Threads will also be discussed as well as an examination of issues related to multicore systems.
3. **Memory Management**
   The student will be introduced to the management of main memory during the execution of a process. To improve both the utilization of the CPU and the speed of its response to its users, the computer must keep several processes in memory. The student will be introduced to several of the many different memory-management schemes, reflecting various approaches to memory management. The student will also explore the effectiveness of particular algorithms depending on varying situations.
4. **Storage Management**
   The student will learn how the file, mass storage, and I/O are handled in a modern computer system. The file system provides the mechanism for on-line storage of and access to both data and programs. The student will explore the classic internal algorithms and structures of storage management and will learn a firm practical understanding of the algorithms used, their properties, advantages, and disadvantages. The student will be introduced to secondary and tertiary storage, and the student will explore system I/O in depth, including I/O system design, interfaces, and internal system structures and functions. Performance issues associated with I/O devices will also be covered.
5. **Protection and Security**
   The student will learn the mechanisms necessary for the protection and security of computer systems. The processes in an operating system must be protected from one another's activities, and to provide such protection, system users must ensure that only processes that have gained proper authorization from the operating system can operate on the files, memory, CPU, and other resources of the system. Protection is a mechanism for controlling the access of programs, processes, or users to the resources defined by a computer system. This mechanism must provide a means of specifying the controls to be imposed, as well as a means of enforcement. Security protects the integrity of the information stored in the system, as well as the physical resources of the system, from unauthorized access, malicious destruction or alteration, and accidental introduction of inconsistency.
6. **Distributed Systems**
   Students will learn about collections of processors that do not share memory or a clock. These are referred to as Distributed Systems. By providing the user with access to the various resources that it maintains, a distributed system can improve computation speed and data availability and reliability. Students will learn that distributed systems also provide the user with a distributed file system, which is a file-service system whose users, servers, and storage devices are dispersed among the sites of a distributed system. A distributed system must provide various mechanisms for process synchronization and communication, as well as for dealing with deadlock problems and a variety of failures that are not encountered in a centralized system.
7. **Special-Purpose Systems**
   Students will learn about systems used for specific purposes, including real-time systems and multimedia systems. These systems have specific requirements that differ from those of general-

purpose systems. Students will learn that real-time systems may require not only that computed results be "correct" but also that the results be produced within a specified deadline period. Multimedia systems require quality-of-service guarantees ensuring that the multimedia data are delivered to clients within a specific time frame.

8. **Case Studies**
   In the final segment of the course, students will be presented with case studies integrating the concepts described in earlier segments of the text by describing real operating systems. These systems include Linux, Windows XP, FreeBSD, Mach, and Windows 2000.

Course Content

Textbook: *Operating Systems Concepts,* 8th Edition, by Abraham Silberschatz, Peter Baer Galvin, and Greg Gagne.
The following chapters including the particular sections listed are covered. (See textbook "Contents")

1. **Introduction**
   What Operating Systems Do
   Computer-System Organization
   Computer-System Architecture
   Operating- System Structure
   Operating-System Operations
   Process Management
   Memory Management
   Storage Management
   Protection and Security
   Distributed Systems
   Special-Purpose Systems
   Computing Environment**s**
   Open-Source Operating Systems
2. **Operating System Structures**
   Operating-System Services
   User Operating-System Interface
   System Calls
   Types of System Calls
   System Programs
   Operating-System Design and Implementation
   Operating-System Structure
   Virtual Machines
   Operating-System Debugging
   Operating-System Generation
   System Boot
3. **Processes**
   Process Concept
   Process Scheduling
   Operations on Processes
   Interprocess Communication
   Examples of IPC Systems
   Communication in Client-Server systems

4. **Threads**
   - Overview
   - Multithreading Models
   - Thread Libraries
   - Threading Issues
   - Operating-System Examples
5. **CPU Scheduling**
   - Basic Concepts
   - Scheduling Criteria
   - Scheduling Algorithms
   - Thread Scheduling
   - Multiple-Processor Scheduling
   - Operating-System Examples
   - Algorithm Evaluation
6. **Process Synchronization**
   - Background
   - The Critical-Section Problem
   - Peterson's Solution
   - Synchronization Hardware
   - Semaphores
   - Classic Problems of Synchronization
   - Monitors
   - Synchronization Examples
   - Atomic Transactions
7. **Deadlocks**
   - System Model
   - Deadlock Characterization
   - Methods for Handling Deadlocks
   - Deadlock Prevention
   - Deadlock Avoidance
   - Deadlock Detection
   - Recovery from Deadlock
8. **Main Memory**
   - Background
   - Swapping
   - Contiguous Memory Allocation
   - Paging
   - Structure of the Page Table
   - Segmentation
   - The Intel Pentium
9. **Virtual Memory**
   - Background
   - Demand Paging
   - Copy-on-Write
   - Page Replacement
   - Allocation of Frames
   - Thrashing
   - Memory-Mapped Files
   - Allocating Kernel Memory

Other Considerations
Operating-System Examples

10. **File-System Interface**
    File Concept
    Access Methods
    Directory and Disk Structure
    File-System Mounting
    File Sharing
    Protection
11. **File-System Implementation**
    File-System Structure
    File-System Implementation
    Directory Implementation
    Allocation Methods
    Free-Space Management
    Efficiency and Performance
    Recovery
    NFS
    The WAFL File System
12. **Mass-Storage Structure**
    Overview
    Disk Structure
    Disk Attachment
    Disk Scheduling
    Disk Management
    Swap-Space Management
    RAID Structure
    Stable-Storage Implementation
    Tertiary-Storage Structure
13. **I/O Systems**
    Overview
    I/O Hardware
    Application I/O Interface
    Kernel I/O Subsystem
    Transforming I/O Requests to Hardware Operations
    STREAMS
    Performance
14. **Protection**
    Goals of Protection
    Principles of Protection
    Domain of Protection
    Access Matrix
    Implementation of Access Matrix
    Access Control
    Revocation of Access Rights
    Capability-Based Systems
    Language-Based Protection
15. **Security**
    The Security Problem

21. **The Linux System**
    - Linux History
    - Design Principles
    - Kernel Modules
    - Process Management
    - Scheduling
    - Memory Management
    - File Systems
    - Input and Output
    - Interprocess Communication
    - Network Structure
    - Security
22. **Windows XP**
    - History
    - Design Principles
    - System Components
    - Environmental Subsystems
    - File System
    - Networking Programmer Interface
23. **Influential Operating Systems**
    - Feature Migration
    - Early Systems
    - Atlas
    - XDS-940
    - THE
    - RC 4000
    - CTSS
    - MULTICS
    - IBM OS/360
    - TOPS-20
    - CP/M and MS/DOS
    - Macintosh Operating System and Windows
    - Mach
    - Other Systems