UNIVERSITY OF NEBRASKA AT OMAHA
COURSE SYLLABUS/DESCRIPTION

Department and Course Number	CSCI 4500
Course Title	Operating Systems
Course Coordinator	Stanley Wileman
Total Credits	3
Date of Last Revision	October 23, 2008

1.0 Course Description

1.1 Overview of content and purpose of the course
This course introduces the concepts, structure and mechanisms of operating systems. Included are basic concepts of operating systems, concurrency, process synchronization, input/output, memory management, file systems, and system security. A contemporary operating system's API is studied in detail, and applications are written using this API, as well as limited implementation of major operating system algorithms.

1.2 For whom course is intended
This course is intended for upper division students majoring in computer science as well as others with a strong interest in operating systems.

1.3 Prerequisites of the course (Courses)
The prerequisites for this course include an understanding of basic digital hardware and assembly language programming (CSCI 3710 or equivalent), data structures (CSCI 3320), and introductory calculus (MATH 1950). Students will benefit from also having a strong background in computer architecture (CSCI 4350).

1.4 Prerequisites of the course (Topics)
1.4.1 Algorithm design
1.4.2 High-level programming language (like C or C++)
1.4.3 Computer organization at the machine-language level
1.4.4 Fundamental data structures (in particular, arrays, lists, and trees)

1.5 Unusual circumstances of the course
None

2.0 Objectives

2.1 Upon completion of this course, a student should be able to:

2.1.1 understand the purposes, major components and key mechanisms of operating systems;

2.1.2 understand the type of decisions involved in operating system design;

2.1.3 understand the context within which the operating system functions;

2.1.4 understand the relation between computer system architecture and operating system features;

2.1.5 understand the historical development of architecture and operating systems;

2.1.6 understand the major algorithms used in various operating system components and the factors used to evaluate different designs,

2.1.7 utilize the application program interface (API) for at least one contemporary operating system to construct programs that illustrate that API; and

2.1.8 be familiar with methods for providing concurrency, communication and synchronization among concurrent tasks.

3.0 Content and Organization

	Contact hours
3.1 Introduction to Operating Systems	6.0

3.1.1 Views of an operating system
- 3.1.1.1 Resource management
- 3.1.1.2 Extended machine

3.1.2 What is not in an operating system
- 3.1.2.1 Command interpreter
- 3.1.2.2 Development tools
- 3.1.2.3 Applications

3.1.3 Operating system history
- 3.1.3.1 Stand-alone systems
- 3.1.3.2 Batch systems
- 3.1.3.3 Multiprogramming and time-sharing systems
- 3.1.3.4 Personal computers, networks, distributed systems

3.1.4 Basic operating system concepts
- 3.1.4.1 Processes, tasks, threads
- 3.1.4.2 Files
- 3.1.4.3 The command interpreter
- 3.1.4.4 Other user interfaces

3.1.5 Application program interfaces (system calls)
- 3.1.5.1 Process management
- 3.1.5.2 Signaling
- 3.1.5.3 File management
- 3.1.5.4 Directory management
- 3.1.5.5 Protection
- 3.1.5.6 Time management

3.1.6 Operating system structuring

3.1.6.1 Monolithic systems
3.1.6.2 Layered systems
3.1.6.3 Virtual machines
3.1.6.4 Client-server computing

3.2 Concurrency 10.0

3.2.1 Methods for achieving concurrency
3.2.1.1 Processes
3.2.1.2 Threads

3.2.2 Task communication and synchronization
3.2.2.1 Race conditions
3.2.2.2 Critical sections
3.2.2.3 Busy waiting mutual exclusion
3.2.2.4 Sleep and wakeup
3.2.2.5 Semaphores and mutexes
3.2.2.6 Monitors
3.2.2.7 Message passing

3.2.3 Classic task communication problems
3.2.3.1 Dining philosophers
3.2.3.2 Producer/consumer
3.2.3.3 Readers and writers

3.2.4 Process and thread scheduling
3.2.4.1 Process states
3.2.4.2 Long term scheduling
3.2.4.2.1 First come first served
3.2.4.2.2 Priority
3.2.4.2.3 Shortest job first
3.2.4.3 Short term scheduling
3.2.4.3.1 Round robin
3.2.4.3.2 Priority
3.2.4.3.3 Multiple queues
3.2.4.4 Real time scheduling

3.3 Input/Output 7.0

3.3.1 I/O hardware principles
3.3.1.1 Devices
3.3.1.2 Controllers
3.3.1.3 Direct memory access

3.3.2 I/O software principles
3.3.2.1 Interrupt handlers
3.3.2.2 Device drivers
3.3.2.3 Kernel functions
3.3.2.4 Application models (traditional, event driven)

3.3.3 Deadlocks

- 3.3.3.1 Specific examples
- 3.3.3.2 Deadlock theory
- 3.3.3.3 Dealing with deadlock
  - 3.3.3.3.1 Ignore
  - 3.3.3.3.2 Detect and recover
  - 3.3.3.3.3 Prevention
  - 3.3.3.3.4 Avoidance

3.3.4 Specific device classes
- 3.3.4.1 Disk drives
  - 3.3.4.1.1 Hardware
  - 3.3.4.1.2 Software
- 3.3.4.2 Clocks and timers
  - 3.3.4.2.1 Hardware
  - 3.3.4.2.2 Software
  - 3.3.4.2.3 Implementing multiple timers
- 3.3.4.3 Terminals
  - 3.3.4.3.1 Hardware
  - 3.3.4.3.2 Software

3.4 Memory management 10.0

3.4.1 Early systems: monoprogramming

3.4.2 Multiprogramming with fixed partitions

3.4.3 Multiprogramming with variable partitions

3.4.4 Managing available memory
- 3.4.4.1 Bit maps
- 3.4.4.2 Linked lists

3.4.5 Swapping

3.4.6 Virtual memory
- 3.4.6.1 Paging
- 3.4.6.2 Page tables
- 3.4.6.3 Associative memory
- 3.4.6.4 Inverted page tables

3.4.7 Page replacement algorithms
- 3.4.7.1 OPT
- 3.4.7.2 FIFO
- 3.4.7.3 LRU
- 3.4.7.4 LRU approximations
- 3.4.7.5 Clock

3.4.8 Optimization
- 3.4.8.1 The working set model
- 3.4.8.2 Optimizing applications for paged virtual memory
- 3.4.8.3 Local and global allocation policies
- 3.4.8.4 Page size issues

3.4.9 File mapping to virtual memory

3.4.9.1 Principles
3.4.9.2 Unification of I/O and virtual memory
3.4.9.3 Segmentation
3.4.9.4 Paging and segmentation

3.5 File systems 8.0
3.5.1 File characteristics
3.5.1.1 Naming
3.5.1.2 Structure
3.5.1.3 Attributes
3.5.1.4 Operations
3.5.2 Directory structures
3.5.2.1 Flat file systems
3.5.2.2 Hierarchical file systems
3.5.2.3 Path names
3.5.2.4 Directory operations
3.5.3 File implementation
3.5.4 Directory implementation
3.5.5 Disk space management

3.6 System security 4.0
3.6.1 Security concepts
3.6.2 Physical security
3.6.3 Types of security attacks
3.6.4 User authentication
3.6.5 Protection mechanisms
3.6.5.1 Protection domains
3.6.5.2 Access control lists
3.6.5.3 Capabilities