Outline of Topics to be Covered:

1. Introduction
   1.1. What is an Operating System?
      1.1.1. The Operating System as an Extended Machine
      1.1.2. The Operating System as a Resource Manager
   1.2. History of Operating Systems
      1.2.1. The First Generation (1945–55) Vacuum Tubes
      1.2.2. The Second Generation (1955–65) Transistors and Batch Systems
      1.2.3. The Third Generation (1965–1980) ICs and Multiprogramming
      1.2.4. The Fourth Generation (1980–Present) Personal Computers
   1.3. Computer Hardware Review
      1.3.1. Processors
      1.3.2. Memory
      1.3.3. Disks
      1.3.4. Tapes
      1.3.5. I/O Devices
      1.3.6. Buses
      1.3.7. Booting the Computer
   1.4. The Operating System Zoo
      1.4.1. Mainframe Operating Systems
      1.4.2. Server Operating Systems
      1.4.3. Multiprocessor Operating Systems
      1.4.4. Personal Computer Operating Systems
      1.4.5. Handheld Computer Operating Systems
      1.4.6. Embedded Operating Systems
      1.4.7. Sensor Node Operating Systems
      1.4.8. Real-Time Operating Systems
      1.4.9. Smart Card Operating Systems
   1.5. Operating System Concepts
      1.5.1. Processes
      1.5.2. Address Spaces
      1.5.3. Files
      1.5.4. Input/Output
      1.5.5. Protection
      1.5.6. The Shell
      1.5.7. Ontogeny Recapitulates Phylogeny
   1.6. System Calls
      1.6.1. System Calls for Process Management
1.6.2. System Calls for File Management
1.6.3. System Calls for Directory Management
1.6.4. Miscellaneous System Calls
1.6.5. The Windows Win32 API

1.7. Operating System Structure
1.7.1. Monolithic Systems
1.7.2. Layered Systems
1.7.3. Microkernels
1.7.4. Client-Server Model
1.7.5. Virtual Machines
1.7.6. Exokernels

1.8. The World According to C
1.8.1. The C Language
1.8.2. Header Files
1.8.3. Large Programming Projects
1.8.4. The Model of Run Time

1.9. Research on Operating Systems
1.10. Outline of the Rest of this Book
1.11. Metric Units
1.12. Summary

2. Processes and Threads
2.1. Processes
2.1.1. The Process Model
2.1.2. Process Creation
2.1.3. Process Termination
2.1.4. Process Hierarchies
2.1.5. Process States
2.1.6. Implementation of Processes
2.1.7. Modeling Multiprogramming

2.2. Threads
2.2.1. Thread Usage
2.2.2. The Classical Thread Model
2.2.3. POSIX Threads
2.2.4. Implementing Threads in User Space
2.2.5. Implementing Threads in the Kernel
2.2.6. Hybrid Implementations
2.2.7. Scheduler Activations
2.2.8. Pop-Up Threads
2.2.9. Making Single-Threaded Code Multithreaded

2.3. Interprocess Communication
2.3.1. Race Conditions
2.3.2. Critical Regions
2.3.3. Mutual Exclusion with Busy Waiting
2.3.4. Sleep and Wakeup
2.3.5. Semaphores
2.3.6. Mutexes
2.3.7. Monitors
2.3.8. Message Passing
2.3.9. Barriers

2.4. Scheduling
   2.4.1. Introduction to Scheduling
   2.4.2. Scheduling in Batch Systems
   2.4.3. Scheduling in Interactive Systems
   2.4.4. Scheduling in Real-Time Systems
   2.4.5. Policy versus Mechanism
   2.4.6. Thread Scheduling

2.5. Classical IPC Problems
   2.5.1. The Dining Philosophers Problem
   2.5.2. The Readers and Writers Problem

2.6. Research on Processes and Threads

2.7. Summary

3. Memory Management
   3.1. No Memory Abstraction
   3.2. A Memory Abstraction: Address Spaces
      3.2.1. The Notion of an Address Space
      3.2.2. Swapping
      3.2.3. Managing Free Memory
   3.3. Virtual Memory
      3.3.1. Paging
      3.3.2. Page Tables
      3.3.3. Speeding Up Paging
      3.3.4. Page Tables for Large Memories
   3.4. Page Replacement Algorithms
      3.4.1. The Optimal Page Replacement Algorithm
      3.4.2. The Not Recently Used Page Replacement Algorithm
      3.4.3. The First-In, First-Out (FIFO) Page Replacement Algorithm
      3.4.4. The Second-Chance Page Replacement Algorithm
      3.4.5. The Clock Page Replacement Algorithm
      3.4.6. The Least Recently Used (LRU) Page Replacement Algorithm
      3.4.7. Simulating LRU in Software
      3.4.8. The Working Set Page Replacement Algorithm
      3.4.9. The WSClock Page Replacement Algorithm
      3.4.10. Summary of Page Replacement Algorithms
   3.5. Design Issues for Paging Systems
      3.5.1. Local versus Global Allocation Policies
      3.5.2. Load Control
      3.5.3. Page Size
      3.5.4. Separate Instruction and Data Spaces
      3.5.5. Shared Pages
      3.5.6. Shared Libraries
      3.5.7. Mapped Files
      3.5.8. Cleaning Policy
3.5.9. Virtual Memory Interface

3.6. Implementation Issues
   3.6.1. Operating System Involvement with Paging
   3.6.2. Page Fault Handling
   3.6.3. Instruction Backup
   3.6.4. Locking Pages in Memory
   3.6.5. Backing Store
   3.6.6. Separation of Policy and Mechanism

3.7. Segmentation
   3.7.1. Implementation of Pure Segmentation
   3.7.2. Segmentation with Paging: MULTICS
   3.7.3. Segmentation with Paging: The Intel Pentium

3.8. Research on Memory Management

3.9. Summary

6. Deadlocks
   6.1. Resources
      6.1.1. Preemptable and Nonpreemptable Resources
      6.1.2. Resource Acquisition
   6.2. Introduction to Deadlocks
      6.2.1. Conditions for Resource Deadlocks
      6.2.2. Deadlock Modeling
   6.3. The Ostrich Algorithm
   6.4. Deadlock Detection and Recovery
      6.4.1. Deadlock Detection with One Resource of Each Type
      6.4.2. Deadlock Detection with Multiple Resources of Each Type
      6.4.3. Recovery from Deadlock
   6.5. Deadlock Avoidance
      6.5.1. Resource Trajectories
      6.5.2. Safe and Unsafe States
      6.5.3. The Banker's Algorithm for a Single Resource
      6.5.4. The Banker's Algorithm for Multiple Resources
   6.6. Deadlock Prevention
      6.6.1. Attacking the Mutual Exclusion Condition
      6.6.2. Attacking the Hold and Wait Condition
      6.6.3. Attacking the No Preemption Condition
      6.6.4. Attacking the Circular Wait Condition
   6.7. Other Issues
      6.7.1. Two-Phase Locking
      6.7.2. Communication Deadlocks
      6.7.3. Livelock
      6.7.4. Starvation
   6.8. Research on Deadlocks
   6.9. Summary

** If time is available

9. Security
   9.1. The Security Environment
9.1.1. Threats
9.1.2. Intruders
9.1.3. Accidental Data Loss
9.2. Basics of Cryptography
   9.2.1. Secret-Key Cryptography
   9.2.2. Public-Key Cryptography
   9.2.3. One-Way Functions
   9.2.4. Digital Signatures
   9.2.5. Trusted Platform Module
9.3. Protection Mechanisms
   9.3.1. Protection Domains
   9.3.2. Access Control Lists
   9.3.3. Capabilities
   9.3.4. Trusted Systems
   9.3.5. Trusted Computing Base
   9.3.6. Formal Models of Secure Systems
   9.3.7. Multilevel Security
   9.3.8. Covert Channels
9.4. Authentication
   9.4.1. Authentication Using Passwords
   9.4.2. Authentication Using a Physical Object
   9.4.3. Authentication Using Biometrics
9.5. Insider Attacks
   9.5.1. Logic Bombs
   9.5.2. Trap Doors
   9.5.3. Login Spoofing
9.6. Exploiting Code Bugs
   9.6.1. Buffer Overflow Attacks
   9.6.2. Format String Attacks
   9.6.3. Return to libc Attacks
   9.6.4. Integer Overflow Attacks
   9.6.5. Code Injection Attacks
   9.6.6. Privilege Escalation Attacks
9.7. Malware
   9.7.1. Trojan Horses
   9.7.2. Viruses
   9.7.3. Worms
   9.7.4. Spyware
   9.7.5. Rootkits
9.8. Defenses
   9.8.1. Firewalls
   9.8.2. Antivirus and Anti-Antivirus Techniques
   9.8.3. Code Signing
   9.8.4. Jailing
   9.8.5. Model-Based Intrusion Detection
9.8.7. Java Security
9.9. Research on Security
9.10. Summary