

Review

CS 256/456

Dept. of Computer Science, University of Rochester

Final Exam

- Closed book for CS grads who take Comps Exam; open book for others
- Cover everything from the beginning, but more emphasis on the parts after midterm

What is an Operating System?

- It is an extended machine
 - Hides the messy details about hardware resources
 - Presents users with a resource abstraction that is easy to use
- It is a resource manager
 - Allows multiple users/programs to share resources fairly, efficiently, ...

Overall Picture

- OS components
 - Process Management and Scheduling
 - Synchronization
 - Memory Management
 - I/O System Management
 - File and Secondary-Storage Management
- OS structures
 - Monolithic kernel
 - Micro-kernel
 - Extensible systems
 - Virtual machines

Processes & Threads

- Process
 - Process concept
 - OS data structure for a process
 - Operations on processes
- Thread
 - Thread concept
 - Compared with process
 - less context switch overhead
 - more efficient synchronization between threads
 - User/kernel threads

CPU Scheduling

- Selects from among the processes/threads that are ready to execute, and allocates the CPU to it.
- CPU scheduling may take place at:
 1. Hardware interrupt/software exception.
 2. System calls.
- Scheduling schemes:
 - FCFS
 - Shortest job first
 - Priority scheduling
 - Round-robin
- CPU scheduling in practice

Synchronization

- Concurrent access to shared data may result in race condition
- The Critical-Section problem
 - Pure software solution
 - With help from the hardware
- Synchronization without busy waiting
 - Semaphore
 - Mutex lock

High-level Synchronization

- Classic synchronization problems
 - Bounded buffer (producer/consumer)
 - Dining philosopher
- High-level synchronization primitives
 - Monitor
 - Condition variables

Deadlocks

- Deadlocks
 - Four characterizations: Mutual exclusion, Hold and wait, No preemption, Circular wait
- Handling deadlocks:
 - Ignore the problem and pretend that deadlocks would never occur.
 - Ensure that the system will *never* enter a deadlock state.
 - deadlock prevention
 - deadlock avoidance
 - Allow the system to enter a deadlock state and then detect/recover.

Memory Management

- Address binding
 - compile-time, load-time, execution-time
 - Logical vs. physical address
- Memory management
 - space allocation & address translation (memory mapping unit)
- Paging (non-contiguous allocation)
 - address translation: page tables and TLB
 - hierarchical page tables, inverted/hashed page tables
- Segmentation
 - compile time: segmented logical addresses
 - execution time: translated into physical addresses

Virtual Memory

- **Virtual memory** - separation of user logical memory from physical memory.
 - Only part of the program address space needs to be in physical memory for execution.
 - Copy-on-write: allows for more efficient process creation.
 - Memory-mapped I/O.
- Page replacement algorithm: the algorithm that picks the victim page.
 - Metrics: page-fault rate; implementation cost/feasibility.
 - FIFO, Optimal, LRU, LRU approximation.

I/O & Storage Systems

- I/O:
 - interrupt-driven
- Disk Structure
- Disk Scheduling
 - FCFS, SSTF, SCAN, C-SCAN, deadline scheduling, anticipatory scheduling [Iyer & Druschel, SOSP 2001]

File System

- File system interface
 - files/directories
 - access models and operations
- Space allocation for disk files
 - contiguous allocation, linked allocation, indexed allocation
 - space efficiency and access efficiency (random/sequential)
- Free space management
 - bit map, linked list,
- I/O buffer management
 - caching and prefetching
- Journaling file system & log-structured file system

Security

- User authentication
 - UNIX user authentication and attacks
 - Login spoofing
- Buffer overflow attack
- Viruses and anti-virus techniques

Protection

- Operating system consists of a collection of objects, hardware or software (e.g., files, printers)
- Protection problem - ensure that each object is accessed correctly and only by those processes that are allowed to do so
- Access control list & capabilities
- Formal models of secure systems
 - covert channels

Concurrent Online Servers

- Servers that
 - accept concurrent requests (potentially high concurrency)
 - serve online users (interactive responses)
- CPU-intensive (with some but very little I/O)
 - CPU efficiency is the main concern
 - process/threads, user threads
 - user thread blocking problems and solution
- Disk I/O-intensive (too much data there)
 - Disk I/O efficiency is the main concern
 - non-working conserving I/O scheduling
 - more aggressive prefetching

Microkernel

- Microkernel structure:
 - Moves functionalities from the kernel into "user" space.
- Benefits:
 - Modular design?
 - More reliable/secure (less code is running in kernel mode)
- Disadvantage on performance:
 - Tend to have more frequent domain crossings.
- Two types of micro-kernels:
 - Running user-level OS in a trusted server - Mach
 - Running user-level OS within untrusted user processes - Exokernel
 - more flexibility, but problems on security and isolation

OS Extension/Customization

- Approaches
 - Exokernel
 - Downloading code into the kernel (safe languages and sandboxing)
 - The Graybox approach
 - Virtual machine
- Comparison on
 - flexibility
 - simplicity - how much changes in the original OS?
 - overhead of each customization/extension