IS 335: Information Technology in Business
Lecture Outline
Operating Systems

Objectives

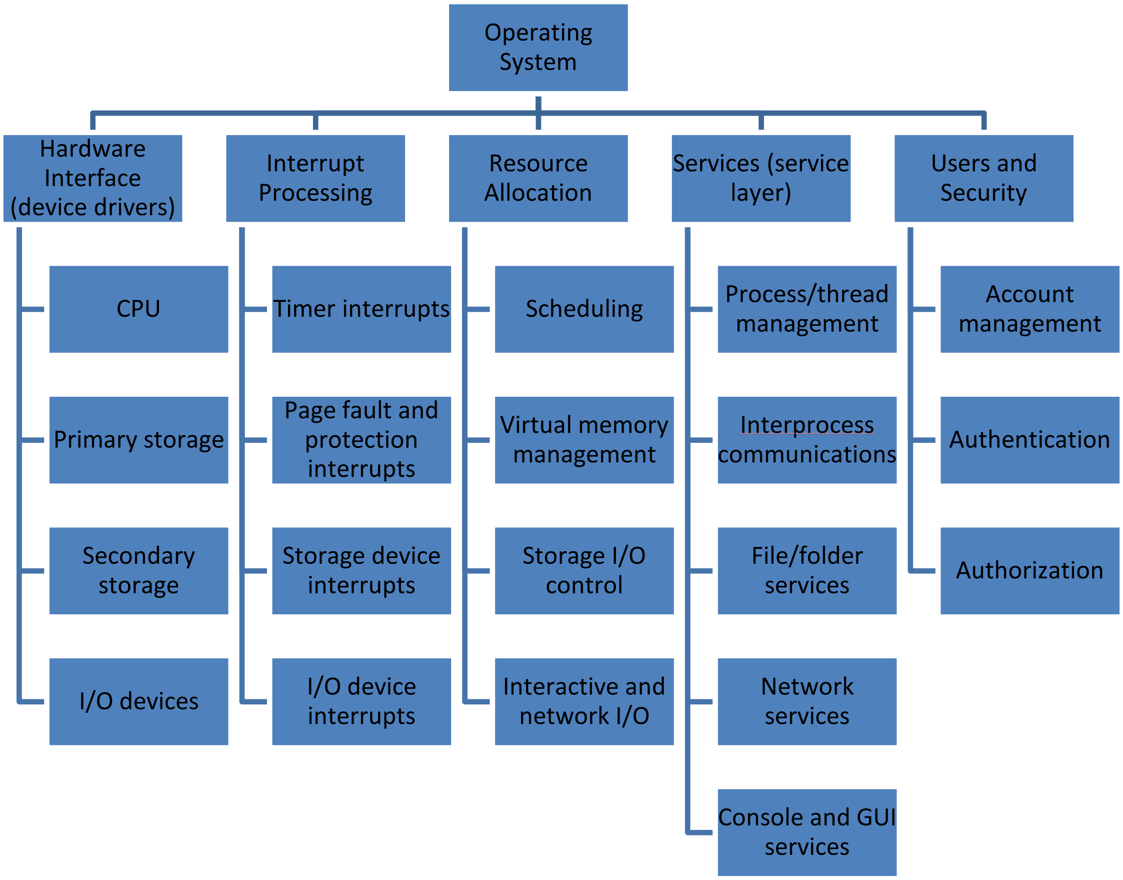
- Describe the functions and layers of an operating system
- List the resources allocated by the operating system and describe the allocation process
- Explain how an operating system manages processes and threads
- Compare CPU scheduling methods
- Explain how an operating system manages memory

Operating System Overview

- Important part of all information systems
- Manages all hardware resources and allocates them to users and applications as needed
- Accesses files and directories, creates and moves windows, and accesses resources over a network

Operating System Functions

- Functions are divided into five main groups
- Resource allocation:
  - Ensure that overall system objectives are achieved efficiently and effectively
  - Bridge between users, their processes, and the hardware resources used by these processes.



Operating System Layers

- Using layers makes the OS easier to maintain
- Command layer (shell)
- Command language or job control language (JCL)
- Service layer
- Service call
- Kernel

Resource Allocation

- Single-tasking and multitasking
- Resource allocation tasks
- Real and virtual resources

Single-Tasking Resource Allocation

- Involves only two running programs
  - Application
  - OS (grants the application all unused hardware resources)
- Single-tasking operating systems are small and efficient
- There's only one active program

Multitasking Resource Allocation

- Norm for modern general-purpose computers
- Allows flexibility of application and system software
- Resource allocation goals
  - Meet resource needs of each program
  - Prevent programs from interfering with one another
  - Efficiently use hardware and other resources

Resource Allocation Tasks

- Keep detailed records of available resources; know which resources can satisfy which requests
- Schedule resources based on specific allocation policies
- Update records to reflect resource commitment and release by programs and users

Real and Virtual Resources

- Real resources
  - Physical devices and associated system software
- Virtual resources
  - Resources that are apparent to a process or user
  - Meet or exceed real resources by:
- •Rapidly shifting resources unused by one program to other programs that need them
- Substituting one type of resource for another

Process Management

- Process
  - Unit of executing software managed independently by OS
  - Can request and receive hardware resources and OS services
  - Can be stand-alone or part of a group that cooperates to achieve a common purpose
  - Can communicate with other processes executing on the same computer or on other computers

Process Control Data Structures

- Process control block (PCB)
  - Created, updated, and deleted by OS
  - Used by OS to perform many functions (e.g., resource allocation, secure resource access, protecting active processes from interference with other active processes)
  - Normally organized into a larger data structure (called a linked list, process queue, or process list)

Process Control Data Structures (continued)

- Processes can spawn other processes and communicate with them
  - Parent process
  - Child process
  - Sibling process
  - Process family

Threads

- Portion of a process that can be scheduled and executed independently
- Can execute concurrently on a single processor or simultaneously on multiple processors
- Share all resources allocated to parent process
- Advantage: reduce OS overhead for resource allocation and process management
- Thread control block (TCB) and run queues

CPU Allocation

- OS makes rapid decisions about which threads receive CPU control and for how long that control is retained
- Threads usually share CPUs (concurrent or interleaved execution)

Thread States

- Ready
  - Waiting for access to the CPU
- Running
  - Retains control of CPU until the thread or its parent process terminates normally or an interrupt occurs
- Blocked
  - Waiting for some event to occur (completion of service request or correction of an error condition)

Interrupt Processing

- Thread can be blocked waiting for resources
- Thread is put into a wait state and its state stored on the stack
- Some interrupt handler will process the blocked thread's request
- When the resource has been allocated, thread is moved from block state to ready or running state
- The thread remains in the blocked state until the request is satisfied or a time out occurs

Scheduling

- Decision-making process used by OS to determine which ready thread moves to the running state
- Typical methods
  - Preemptive scheduling
  - Priority-based scheduling
  - Real-time scheduling

Preemptive Scheduling

- A thread can be removed involuntarily from the running state
- Functions of the supervisor (portion of OS that receives control)
  - Calls appropriate interrupt handler
  - Transfers control to the scheduler
- Functions of the scheduler
  - Updates status of any process or thread affected by last interrupt
  - Decides which thread to dispatch to the CPU
  - Updates thread control information and the stack to reflect the scheduling decision
  - Dispatches selected thread
- Processing steps on left occur after Thread 1 makes an I/O service call
- Processing steps on right occur after I/O device completes I/O operation

Timer Interrupts

- Generated at regular intervals by CPU to give scheduler an opportunity to suspend currently executing thread
- Not a “real” interrupt
- No interrupt handler to call
- Supervisor passes control to the scheduler
- Important CPU hardware feature for multitasking OSs

Priority-Based Scheduling

- Determines which ready thread should be dispatched to the CPU according to:
  - First come first served (FCFS)
  - Explicit priority
  - Shortest time remaining (STR)

Real-Time Scheduling

- Guarantees minimum amount of CPU time to a thread if the thread makes an explicit realtime scheduling request when it is created
- Guarantees a thread enough resources to complete its function within a specified time
- Often used in transaction processing, data acquisition, and automated process control

Memory Allocation

- The assignment of specific memory addresses to system software, application programs, and data
- OS allocates memory
- –When threads are created; responds to requests for additional memory during a thread’s lifetime
  - To itself and for other needs (buffers and caches)

Physical Memory Organization

- Main memory can be regarded as a sequence of contiguous, or adjacent, memory cells
- Most significant byte
- Least significant byte
- Addressable memory
- Physical memory

Single-Tasking Memory Allocation

- Bulk of OS normally occupies lower memory addresses
  - Application program is loaded immediately above it
- Contiguous memory allocation
- Address resolution
  - Process of determining physical memory address that corresponds to memory reference

Multitasking Memory Allocation

- The operating system:
  - Finds free memory regions in which to load new processes and threads
  - Reclaims memory when processes or threads terminate

Multitasking Memory Allocation (continued)

- Goals of multitasking memory allocation
  - Allow as many active processes as possible
  - Respond quickly to changing memory demands of processes
  - Prevent unauthorized changes to a process's memory region(s)
  - Perform memory allocation and addressing as efficiently as possible

Memory Fragmentation

- Occurs when memory partitions allocated to a single process or purpose are scattered throughout physical memory
- To address the problem
  - Compaction (large overhead)
  - Noncontiguous memory allocation

Noncontiguous Memory Allocation

- Portions of a process can be allocated to free partitions anywhere in memory
- Uses small fixed-sized partitions
- More flexible than contiguous memory allocation, but requires more complex partition tables and address calculations

Virtual Memory Management

- Allocates portions of processes (pages) to small memory partitions (page frames)
- Swaps pages between memory and secondary storage as needed
- Page hits, page faults, page tables
- Page files and victims

Memory Protection

- Refers to protecting memory allocated to one program from unauthorized access by another program
- Prevents errors in one program from generating errors in another
- Adds overhead to each write operation

Memory Management Hardware

- Complex memory management procedures incur substantial overhead
- Modern CPUs incorporate advanced memory allocation and address resolution functions in hardware (e.g., Intel Pentium)