





Four Components of a Computer System





Computer System Organization

- Computer-system operation
 - One or more CPUs, device controllers connect through common bus providing access to shared memory
 - Concurrent execution of CPUs and devices competing for memory cycles



Operating System Concepts – 7th Edition, Jan 12, 2005

Storage-Device Hierarchy





Operating System Concepts – 7th Edition, Jan 12, 2005



Operating System Structure

- Multiprogramming needed for efficiency
 - Single user cannot keep CPU and I/O devices busy at all times
 - Multiprogramming organizes jobs (code and data) so CPU always has one to execute
 - A subset of total jobs in system is kept in memory
 - One job selected and run via job scheduling
 - When it has to wait (for I/O for example), OS switches to another job
- Timesharing (multitasking) is logical extension in which CPU switches jobs so frequently that users can interact with each job while it is running, creating interactive computing
 - **Response time** should be < 1 second
 - Each user has at least one program executing in memory ⇒**process**
 - If several jobs ready to run at the same time ⇒ CPU scheduling
 - If processes don't fit in memory, swapping moves them in and out to run
 - Virtual memory allows execution of processes not completely in memory





Memory Layout for Multiprogrammed System





Operating System Concepts – 7th Edition, Jan 12, 2005



Basic idea of Multiprogramming







Operating-System Operations

- Interrupt driven by hardware
- Software error or request creates exception or trap
 - Division by zero, request for operating system service
- Other process problems include infinite loop, processes modifying each other or the operating system
- Dual-mode operation allows OS to protect itself and other system components
 - User mode and kernel mode
 - Mode bit provided by hardware
 - Provides ability to distinguish when system is running user code or kernel code
 - Some instructions designated as privileged, only executable in kernel mode
 - System call changes mode to kernel, return from call resets it to user





Transition from User to Kernel Mode

Timer to prevent infinite loop / process hogging resources

- Set interrupt after specific period
- Operating system decrements counter
- When counter zero generate an interrupt
- Set up before scheduling process to regain control or terminate program that exceeds allotted time





Silberschatz, Galvin and Gagne ©2005



Transition from User to Kernel Mode





Operating System Concepts – 7th Edition, Jan 12, 2005

Silberschatz, Galvin and Gagne ©2005



Process Management

- A process is a program in execution. It is a unit of work within the system. Program is a passive entity, process is an active entity.
- Process needs resources to accomplish its task
 - CPU, memory, I/O, files
 - Initialization data
- Process termination requires reclaim of any reusable resources
- Single-threaded process has one program counter specifying location of next instruction to execute
 - Process executes instructions sequentially, one at a time, until completion
- Multi-threaded process has one program counter per thread
- Typically system has many processes, some user, some operating system running concurrently on one or more CPUs
 - Concurrency by multiplexing the CPUs among the processes / threads

