**Chapter 1 Introduction**

* What Operating System Do
* Computer System Organization
* Computer System Architecture
* Operating System Structure
* Operating System Operations
* Process Management
* Memory Management
* Storage Management
* Protection and Security
* Distributed Systems
* Special Purpose Systems
* Computing Environment

Objective

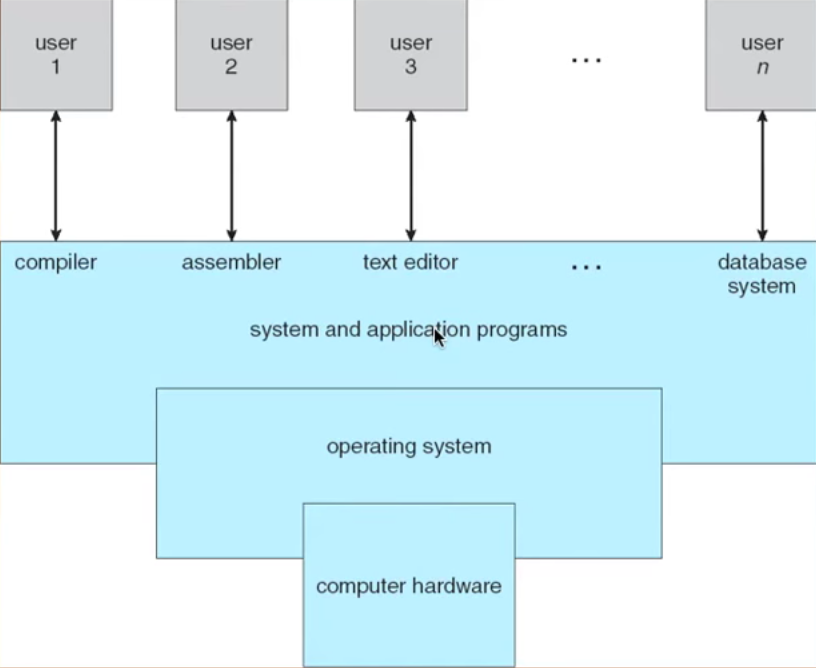
* To provide grand tour of major operating systems components
* To provide coverage of basic computer system organization

What is an Operating System?

* A program that acts as an intermediary between a user of a computer and computer hardware
* Computer system goals:
* Execute user programs and make solving user problems easier
* Make the computer system convenient to use
* Use the computer hardware in an efficient manner

Computer System Structure

* Computer system can be divide into four components
* Hardware --- provide basic computing resources
* CPU, memory, I/O devices
* Operating system
* Controls and coordinates use of hardware among various applications and users
* Application programs --- define the ways in which the system resources are used to solve the computing problems of the users
* Word processors, compilers, web browsers, database system, video games
* Users
* People, machines, other computers



Four components of a computer System

We have many users login on the same hardware and resource

The operating system do is that control who will access and when will access

Operating System Definition

* OS is a resource allocator
* Manage all resources
* Decides between conflicting requests for efficient and fair resource use
* OS is a control program
* Controls executions of programs to prevent errors and improper use of the computer
* “Take one program running at all times on the computer” is the **kernel**. Everything else is either a system program (ships with the operating system) or an application program

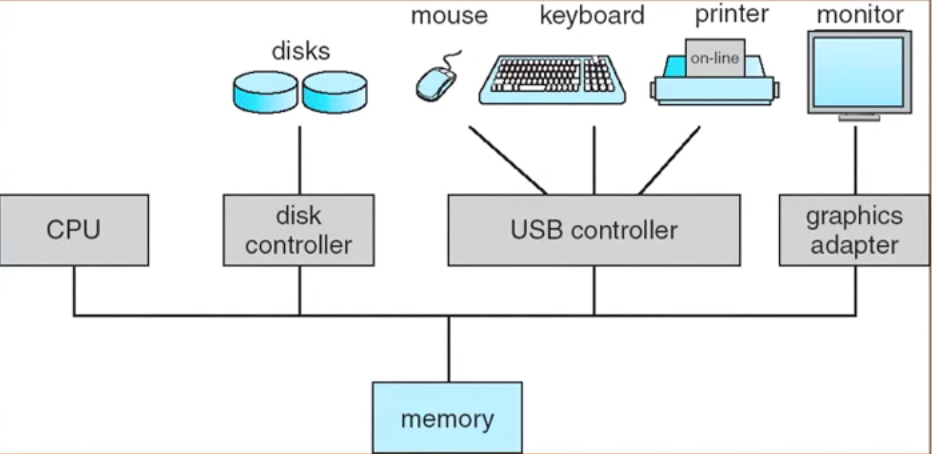
Computer Startup

* **bootstrap** **program** is loaded at power up or reboot
* Typically stored in ROM or EPROM, generally known as **firmware**
* Initializes all aspects of system
* Loads operating system kernel and starts execution

Computer System Organization

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

Computer System Operation

* I/O devices and the CPU can execute concurrently
* Each device controller is in change of a particular device type
* Each device controller has a local buffer
* CPU moves data from/ to main memory to/from local buffers
* I/O is from the device to local buffer of controller
* Device controller informs CPU that it has finished its operation by causing an interrupt

Common Functions of Interrupts

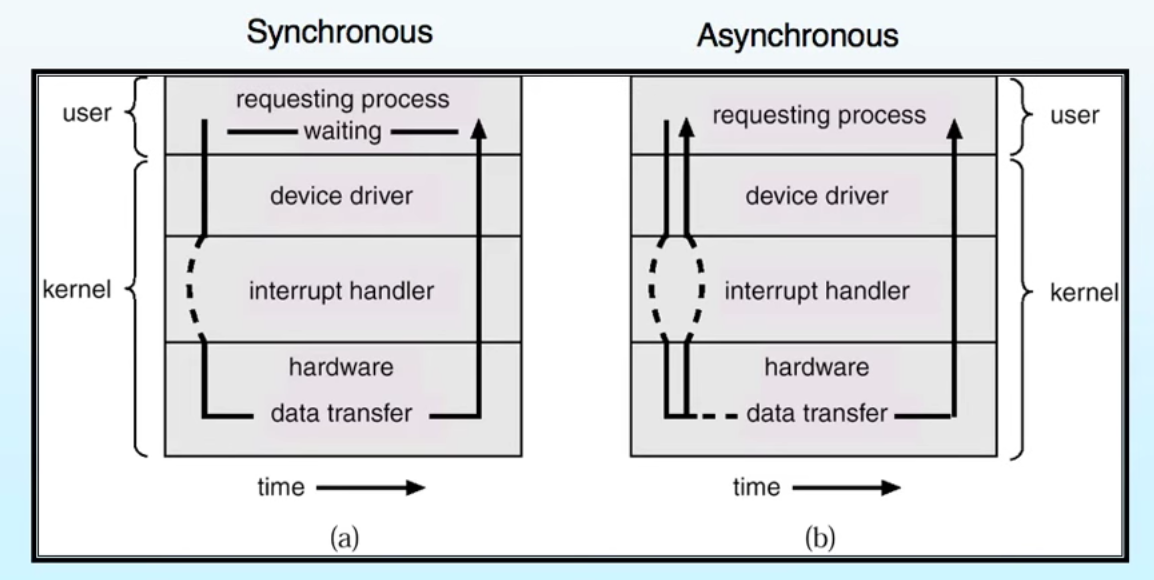
* Interrupt transfers control to the interrupt service routine generally, through the interrupt vector, which contains the address of all the service routines
* Interrupt architecture must save the address of the interrupted instructions
* Incoming interrupts are disabled while another interrupt is being processed to prevent a lost interrupt
* A trap is a software generated interrupt caused either by an error or a user request
* An operating system is interrupt driven

Interrupt Handling

* The operating system preserves the state of the CPU by storing registers and the program counter
* Determines which type of interrupt has occurred:
* Polling
* Vectored interrupt system
* Separate segments of code determine what action should be taken for each type of interrupt

I/O Structure

* After I/O starts, control returns to user program only upon I/O completion
* Wait instruction idles the CPU until the next interrupt
* Wait loop (contention for memory access)
* At most one I/O request is outstanding at a time, no simultaneous I/O processing
* After I/O starts, control returns to user program without waiting for I/O completion
* System call --- request to the operating system to aloe user to wait for I/O completion
* Device status table contains entry for each I/O device indicating its type, address and state
* Operating system indexes into I/O device table to determine devices status and to modify table entry to include interrupt



Two I/O Methods

Direct Memory Access Structure

* Used for high speed I/O devices able to transmit information at close to memory speeds
* Device controller transfers blocks of data from buffer storage directly to main memory without CPU intervention
* Only one interrupt is generated per block, rather than the one interrupt per byte

Storage Structure

* Main memory--- only large storage media that the CPU can access directly
* Secondary storage--- extension of main memory that provides large nonvolatile storage capacity
* Magnetic disks--- rigid metal or glass platters covered with magnetic recording material
* Disk surface is logically divide into tracks, which are subdivided into sectors
* The disk controller determines the logical interaction between the device and the computer

Storage Hierarchy

* Storage systems organized in hierarchy
* Speed
* Cost
* Volatility
* Caching--- copying information into faster storage system; main memory can be viewed as a last cache for secondary memory

Faster Storage (cached) checked first to determine if information is there:

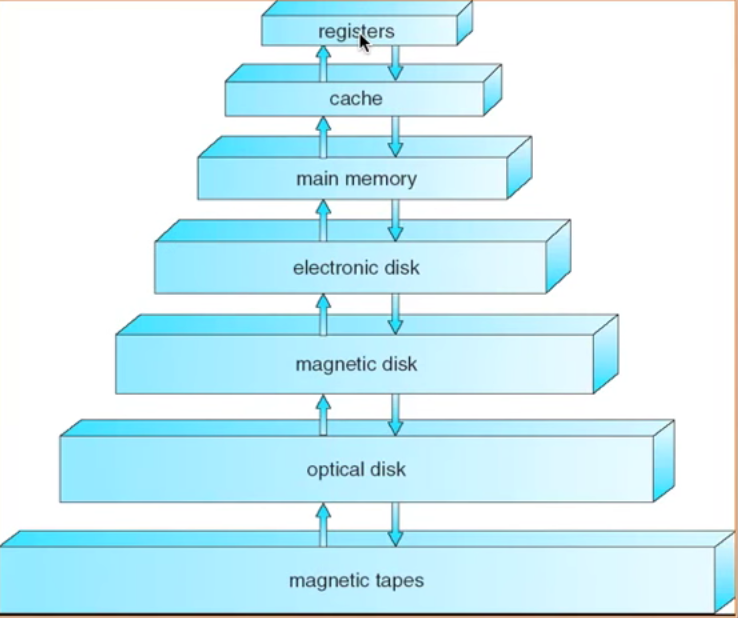
If it is, information used directly from the cache

If not, data copied to cache and used there

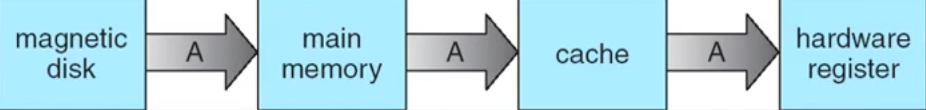
Cache smaller than storage being cached

Cache management important design problem

Cache size and replacement policy



Migration of Integer A from Disk to Register



* Multitasking environments must be careful to use most recent value, no matter it is stored in the storage hierarchy
* Multiprocessor environment must provide cache coherency in hardware such that all CPUs have the most recent value in their cache
* Distributed environment situation even more complex
* Several copies of a datum can exist
* Various solution covered in Chapter 17

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 less than 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 process don’t fit in memory, **swapping** moves them in and out to run
* **Virtual memory** allows execution of processes not completely in memory

Operating System Operations

* Interrupt driven by hardware
* Software error or request creates **exceptions** 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 provide by hardware
* Provides ability to distinguish when system is running user code or kernel code
* Some instructions designed as **privileged**, only executable in kernel mode
* System call changes mode to kernel, return from call resets it to user

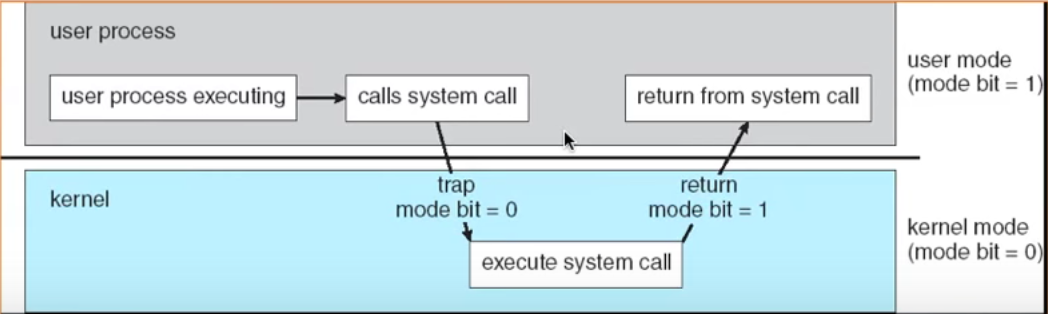
Transition from User to Kernel mode

Timer to prevent infinite loop/ process hogging resources

Set interrupt after specific period

OS decrements counter

When counter 0 GENERATE AN INTERRUPT



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-thread process has one **Program counter** specifying location of next instruction to execute
* Multi-thread process has one program counter per thread
* Typically, system has many process, some user, some operating system running concurrently on one or more CPUs
* Concurrency by multiplexing the CPUs among the processes/threads

Memory Management

* All data in memory before and after processing
* All instructions in memory in order to execute
* Memory management determines what is in memory when
* Optimizing CPU utilization and computer response to users
* Memory management activities
* Keep track of which parts of memory are currently being used and by whom
* Deciding which process (or parts thereof) and data to move into and out of memory
* Allocating and deal locating memory space as needed

Storage Management

* OS provides uniform, logical view of information storage
* Abstracts physical properties to logical storage unit --- file
* Each medium is controlled by device (i.e., disk drive, trap drive)
* Varying properties include access speed, capacity, data transfer rate, access method (sequential or random)
* File System management
* Files usually organized into directories
* Access control on most systems to determine who can access what
* OS activities include
* Creating and deleting files and directories
* Primitives to manipulate files and dirs.
* Mapping files onto secondary storage
* Backup files onto stable (non-volatile) storage media