CSC139 Operating System Principles

Spring 2019, Part 1-2

Instructor: Dr. Yuan Cheng

Session Plan

- Introduction to Operating Systems

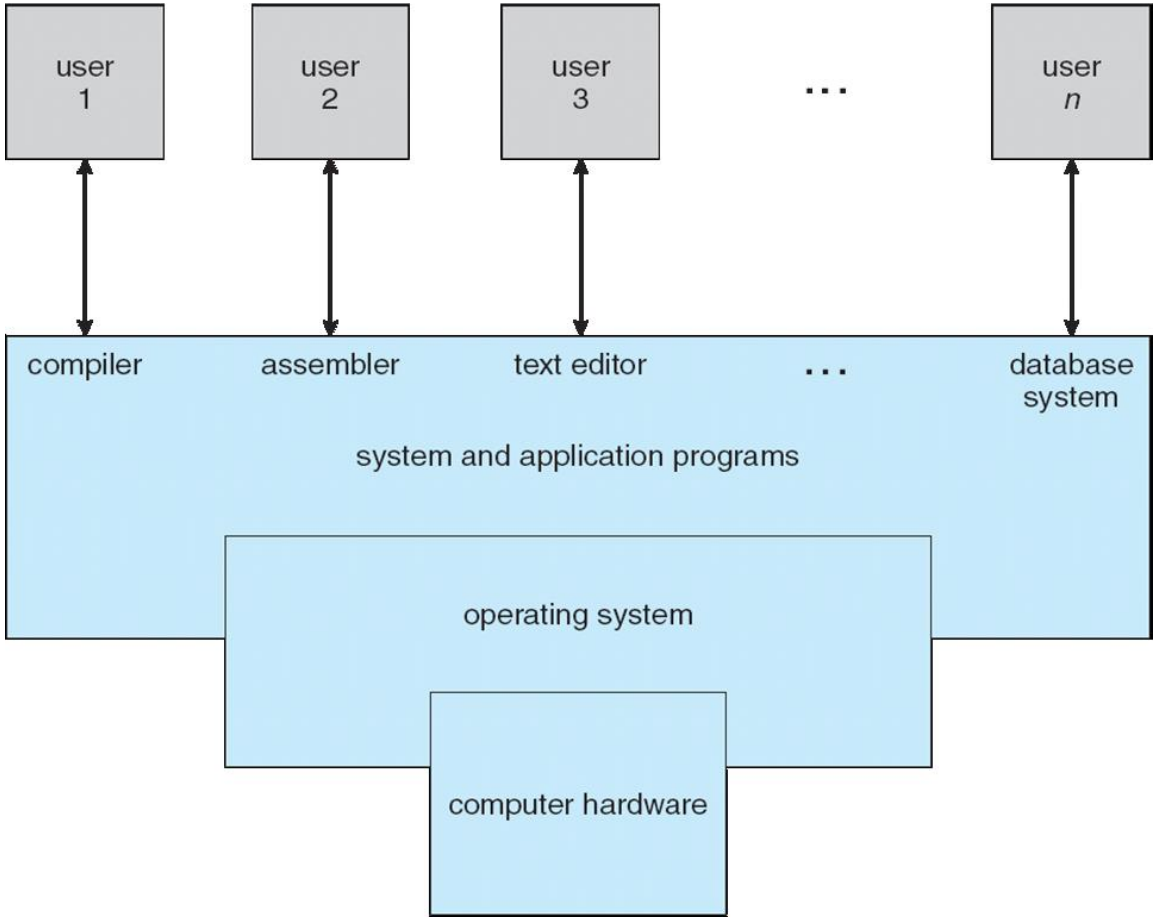
Chapter Objectives

- Describe the general organization of a computer system and the role of interrupts
- Describe the components in a modern, multiprocessor computer system
- Illustrate the transition from user mode to kernel mode
- Discuss how operating systems are used in various computing environments
- Provide examples of free and open-source operating systems

What is an Operating System?

- A program that acts as an *intermediary* between the user and the hardware
  - Provides a virtual execution environment on top of hardware that is more convenient than the raw hardware interface
- Operating 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
- The *government* metaphor

Four Components of a Computer System



Computer System Components

1. *Hardware* – provides basic computing resources
   - CPU, memory, I/O devices
2. *Operating system* – controls and coordinates use of hardware among various applications and users
3. *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 systems, video games
4. *Users*
   - People, machines, other computers

What do a Computer System Do

- Depends on the point of view
- Users want convenience, ease of use and good performance
  - Don't care about resource utilization
- But shared computer such as mainframe or minicomputer must keep all users happy
- Users of dedicate systems such as workstations have dedicated resources but frequently use shared resources from servers
- Handheld computers are resource poor, optimized for usability and battery life
- Some computers have little or no user interface, such as embedded computers in devices and automobiles

Operating System Definition

- OS is a resource allocator
  - Manages all resources
  - Decides between conflicting requests for efficient and fair resource use
- OS is a control program
  - Controls execution of programs to prevent errors and improper use of the computer

Operating System Definition (cont.)

- No universally accepted definition
- “Everything a vendor ships when you order an operating system” is a good approximation
  - But varies wildly
- “The 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.

Quiz 1: Operating System Components

- Which of the following are likely components of an operating system? Check all that apply.
  - A. File editor
  - B. File system
  - C. Device driver
  - D. Cache memory
  - E. Web browser
  - F. Scheduler

Evolution of Operating Systems

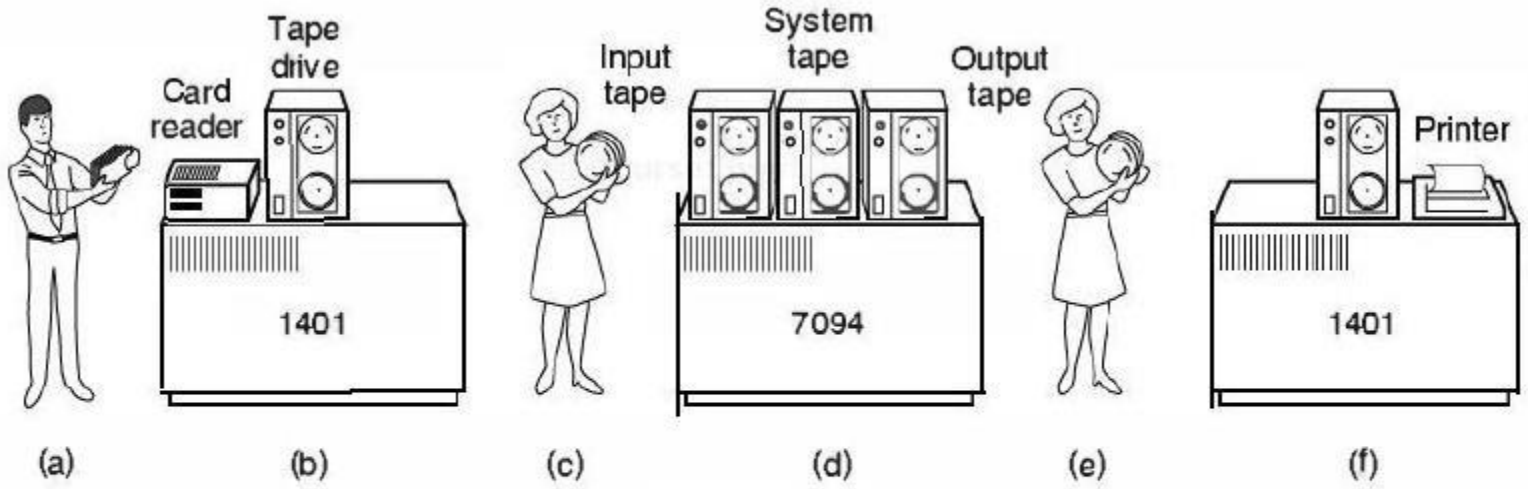
- Mainframe Systems
  - Batch Systems (mid 1950s – mid 1960s)
  - Multi-programmed Systems (1960s)
  - Time-sharing Systems (1970s)
- Personal Computer Operating Systems (1980s)
- Modern Variants
  - Parallel Systems
  - Distributed Systems
  - Real-time and Embedded Systems
  - Ubiquitous Systems

Mainframe Systems

- First computers used to tackle many commercial and scientific applications
- Evolved from simple batch systems, multi-programmed systems, to time-sharing system

Batch Systems

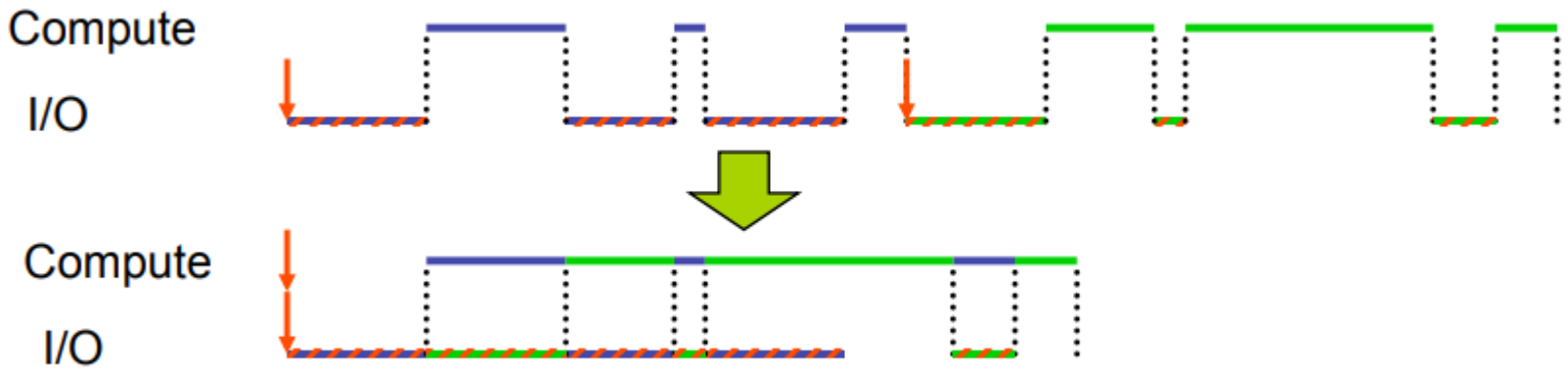
- To speed up processing, operators batched together jobs with similar needs and ran them through the computer as a group.

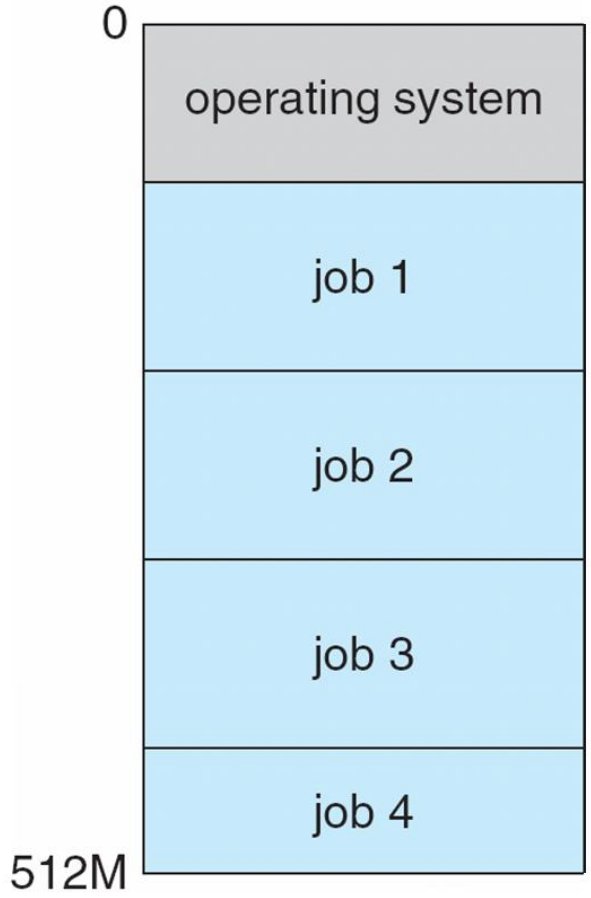


An early batch system. (a) Programmers bring cards to 1401. (b) 1401 reads batch of jobs onto tape. (c) Operator carries input tape to 7094. (d) 7094 does computing. (e) Operator carries output tape to 1401. (f) 1401 prints output.

Multi-programming Systems

- The OS keeps several jobs in memory simultaneously.
- The CPU is switched to another job when I/O takes place





Time-sharing Systems

- Extension of multiprogramming systems to allow on-line interaction with users
- User feels as if she has the entire machine
- Based on time-slicing: divides CPU time equally among active users
- Optimizes for *response time* at the cost of *throughput*

PC Operating Systems

- Earliest ones in the 1980s
- computer system originally dedicated to a single user
- Objective: User convenience and responsiveness
  - Individuals have sole use of computers
  - A single user may not need advanced features of mainframe OS (maximize utilization, protection)

Parallel Systems

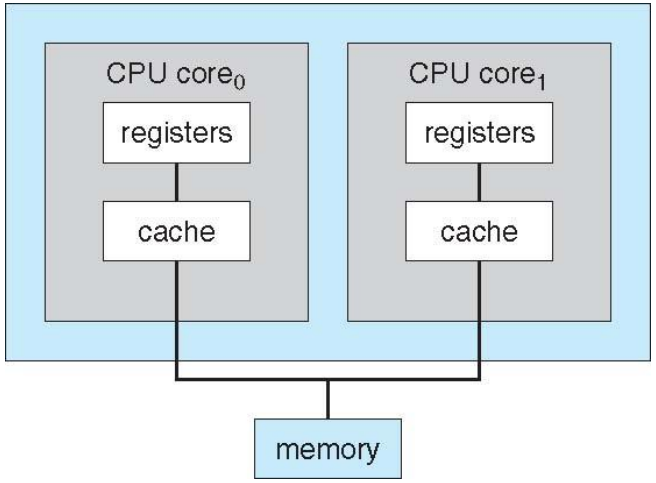
- Parallel systems growing in use and importance
  - Also known as multiprocessor systems, tightly-coupled systems
  - Processors share memory and a clock; communication usually takes place through the shared memory
  - Advantages include:
    1. Increased throughput
    2. Economy of scale
    3. Increased reliability – graceful degradation or fault tolerance
  - Two types:
    1. Asymmetric Multiprocessing – each processor is assigned a specific task.
    2. Symmetric Multiprocessing – each processor performs all tasks

Parallel Systems (cont.)

- Symmetric multiprocessing (SMP)
  - All processors are peers
  - Kernel routines can execute on different CPUs, in parallel
- Asymmetric multiprocessing (AMP)
  - Master/slave structure
  - The kernel runs on a particular processor
  - Other CPUs can execute user programs and OS utilities

Parallel Systems (cont.)

- Multi-core architectures
  - Include multiple computing cores on a single chip
  - Need to exploit parallelism at run-time



Distributed Systems

- Distribute the computation among several physical processors
- Loosely coupled clustered system – each processor has its own local memory; processors communicate with one another through various communications lines
- Advantages of distributed systems
  - Resource and Load Sharing
  - Scalability

Real-time and Embedded Systems

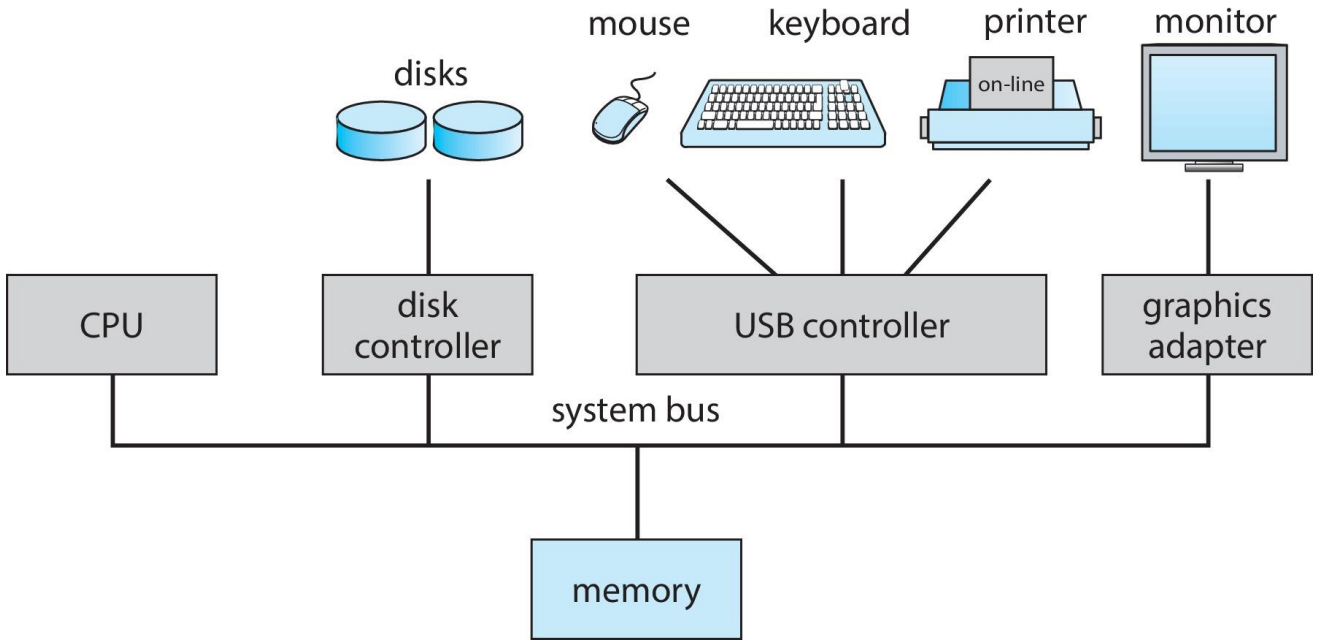
- A real-time system is used when rigid time requirements have been placed on the operation of a processor or the flow of data.
- An embedded system is a component of a more complex system – Control of a nuclear plant
  - Missile guidance
  - Control of home and car appliances
- Real-time systems
  - Real-time means “predictable” not “fast” – have well-defined time constraints
  - May be either hard or soft real-time
    - Hard real-time: OS guarantees that applications will meet their deadlines
    - Soft real-time: OS provides prioritization, on a best-effort basis

Ubiquitous Systems

- PDAs, personal computers, cellular phones, sensors
- Challenges:
  - Small memory size
  - Slow processor
  - Different display and I/O
  - Battery concerns
  - Scale
  - Security
  - Naming

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



Computer System Operation

- I/O devices and the CPU can execute concurrently
- Each device controller is in charge 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

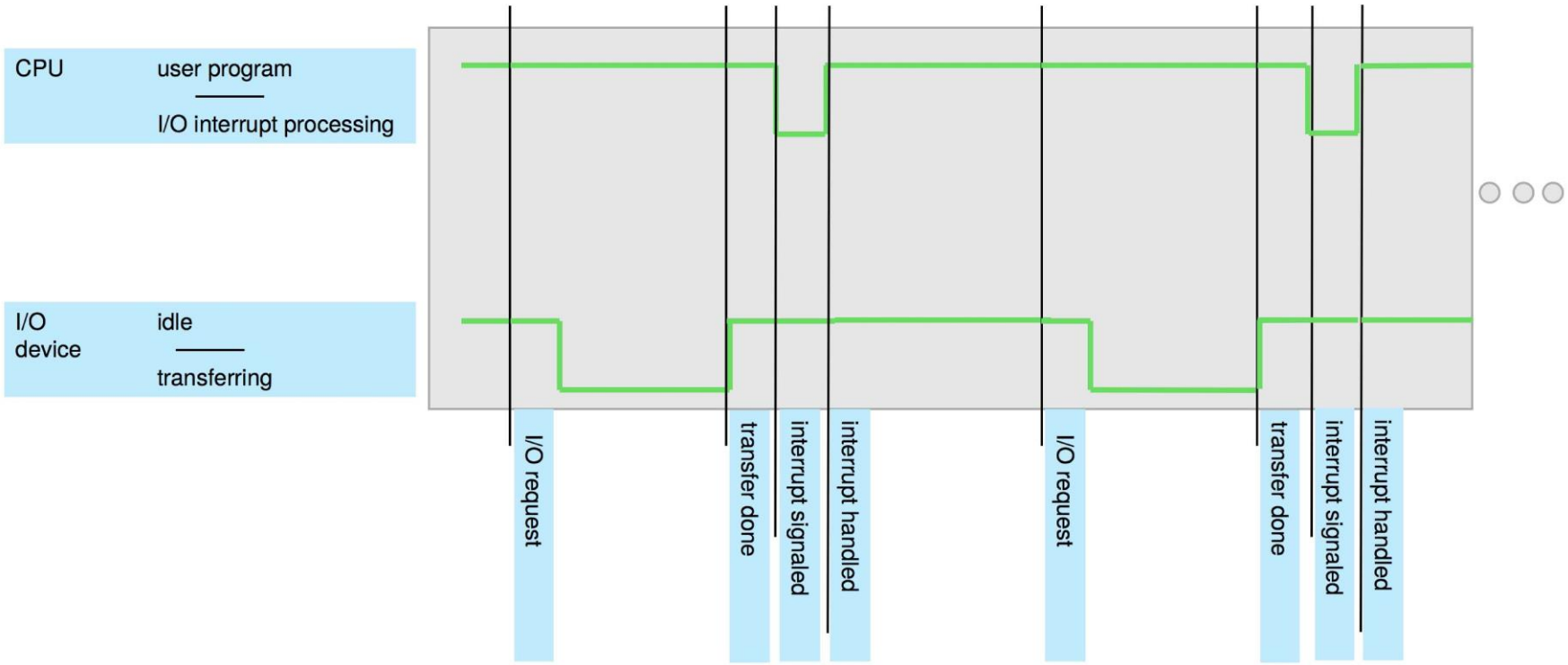
Common Functions of Interrupts

- Interrupt transfers control to the interrupt service routine generally, through the interrupt vector, which contains the addresses of all the service routines
- Interrupt architecture must save the address of the interrupted instruction
- A trap or exception is a software-generated interrupt caused either by an error or a user request
- An operating system is *interrupt driven*

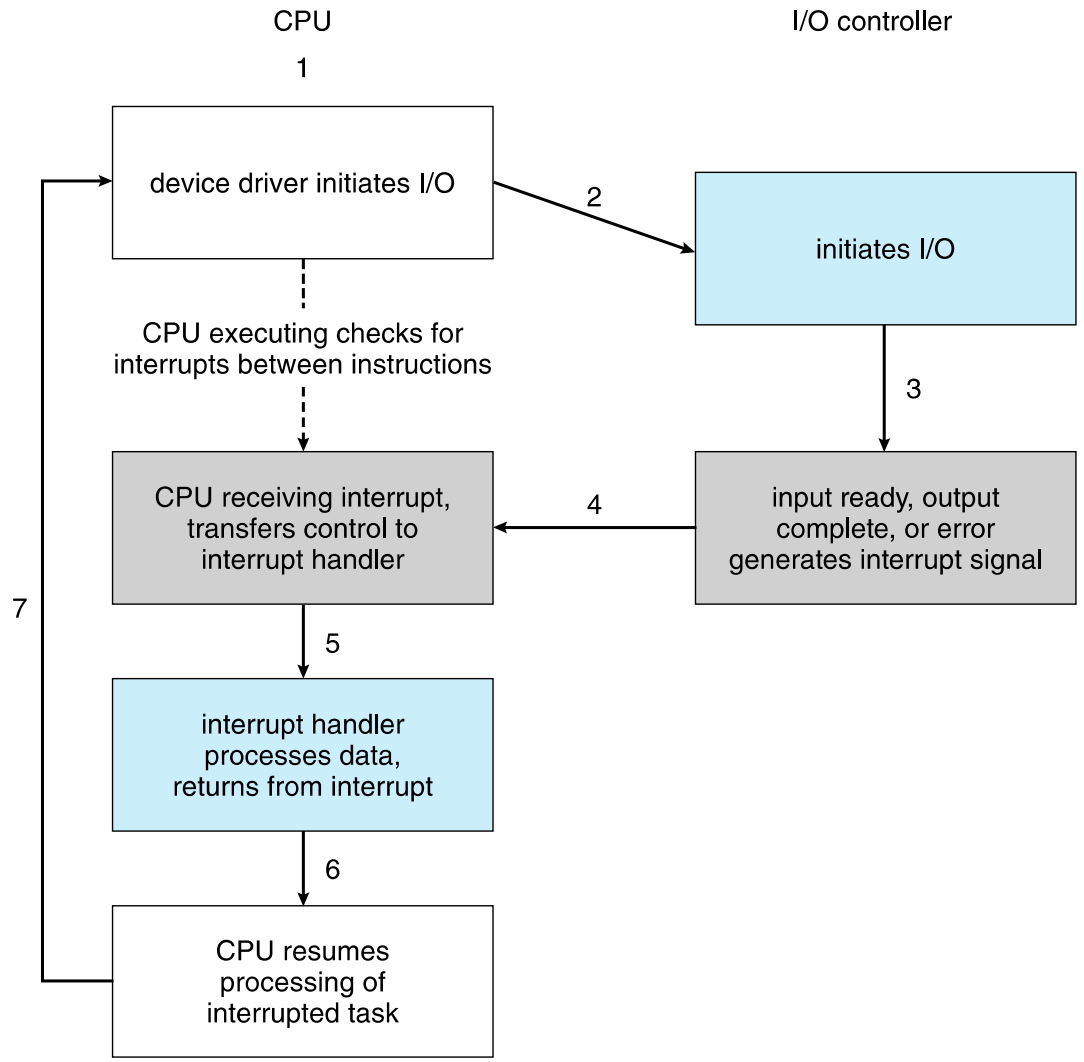
Interrupt Handling

- 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

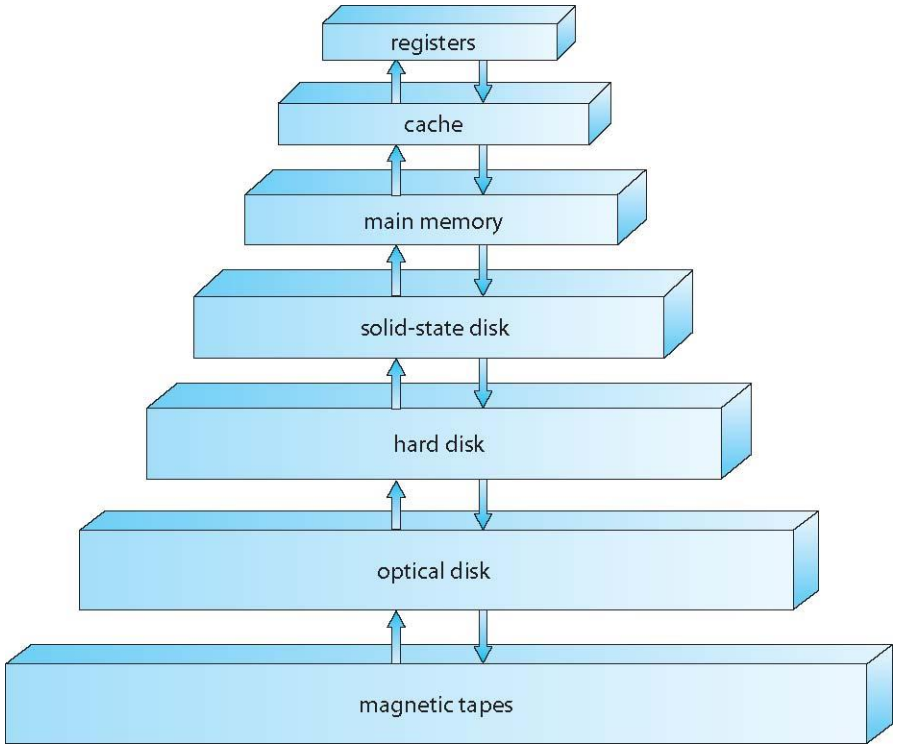
Interrupt Timeline



Interrupt-driven I/O Cycle



Storage-Device Hierarchy



Storage Structure

- Main memory – only large storage media that the CPU can access directly
  - Random access
  - Typically volatile
- Secondary storage – extension of main memory that provides large nonvolatile storage capacity
  - Hard disks – rigid metal or glass platters covered with magnetic recording material
    - Disk surface is logically divided into tracks, which are subdivided into sectors
    - The disk controller determines the logical interaction between the device and the computer
  - Solid-state disks – faster than hard disks, nonvolatile
    - Various technologies
    - Becoming more popular

Storage Hierarchy

- Storage systems organized in hierarchy
  - Speed
  - Cost
  - Volatility
- Caching – copying information into faster storage system; main memory can be viewed as a cache for secondary storage
- Device driver for each device controller to manage I/O
  - Provides uniform interface between controller and kernel

Performance of Various Levels of Storage

Level	1	2	3	4	5
Name	registers	cache	main memory	solid state disk	magnetic disk
Typical size	< 1 KB	< 16MB	< 64GB	< 1 TB	< 10 TB
Implementation technology	custom memory with multiple ports CMOS	on-chip or off-chip CMOS SRAM	CMOS SRAM	flash memory	magnetic disk
Access time (ns)	0.25 - 0.5	0.5 - 25	80 - 250	25,000 - 50,000	5,000,000
Bandwidth (MB/sec)	20,000 - 100,000	5,000 - 10,000	1,000 - 5,000	500	20 - 150
Managed by	compiler	hardware	operating system	operating system	operating system
Backed by	cache	main memory	disk	disk	disk or tape

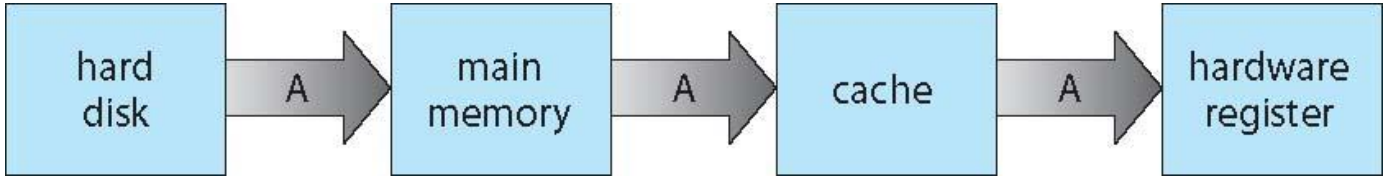
Movement between levels of storage hierarchy can be explicit or implicit

Caching

- Skew rule: 80% requests hit on 20% hottest data
- Important principle, performed at many levels in a computer (in hardware, operating system, software)
- Information in use copied from slower to faster storage temporarily
- Faster storage (cache) checked first to determine if information is there
  - If it is, information used directly from the cache (fast)
  - 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 data “A” from Disk to Register

- Multitasking environments must be careful to use most recent value, no matter where 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 solutions covered in Chapter 17

Operating-System Operations

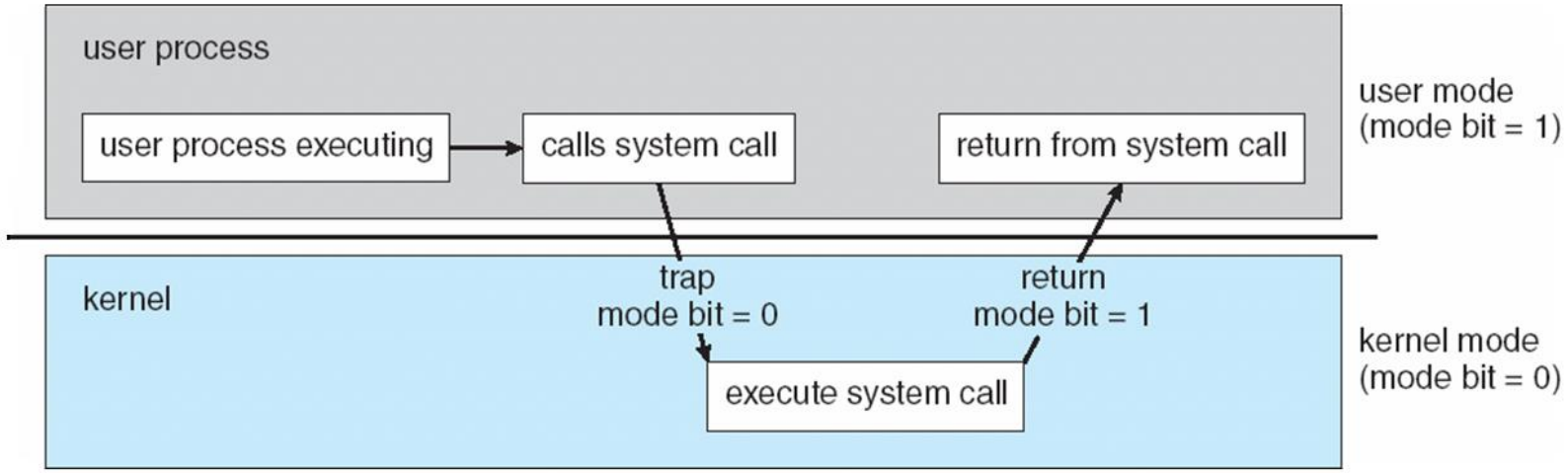
- Interrupt driven (hardware and software)
  - Hardware interrupt by one of the devices
  - Software interrupt (exception or trap):
    - Software error (e.g., division by zero)
    - Request for operating system service
    - Other process problems include infinite loop, processes modifying each other or the operating system

Operating-System Operations (cont.)

- 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
- Increasingly CPUs support multi-mode operations
  - i.e. virtual machine manager (VMM) mode for guest VMs

Transition from User to Kernel Mode

- Timer to prevent infinite loop / process hogging resources
  - Timer is set to interrupt the computer after some time period
  - Keep a counter that is decremented by the physical clock.
  - Operating system set the counter (privileged instruction)
  - When counter zero 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-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

Process Management Activities

- The operating system is responsible for the following activities in connection with process management:
  - Creating and deleting both user and system processes
  - Suspending and resuming processes
  - Providing mechanisms for process synchronization
  - Providing mechanisms for process communication
  - Providing mechanisms for deadlock handling

Memory Management

- To execute a program all (or part) of the instructions must be in memory
- All (or part) of the data that is needed by the program must be in memory.
- Memory management determines what is in memory and when
  - Optimizing CPU utilization and computer response to users
- Memory management activities
  - Keeping track of which parts of memory are currently being used and by whom
  - Deciding which processes (or parts thereof) and data to move into and out of memory
  - Allocating and deallocating 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, tape 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 directories
    - Mapping files onto secondary storage
    - Backup files onto stable (non-volatile) storage media

Mass-Storage Management

- Usually disks used to store data that does not fit in main memory or data that must be kept for a “long” period of time
- Proper management is of central importance
- Entire speed of computer operation hinges on disk subsystem and its algorithms
- OS activities
  - Free-space management
  - Storage allocation
  - Disk scheduling
- Some storage need not be fast
  - Tertiary storage includes optical storage, magnetic tape
  - Still must be managed – by OS or applications
  - Varies between WORM (write-once, read-many-times) and RW (read-write)

I/O Subsystem

- One purpose of OS is to hide peculiarities of hardware devices from the user
- I/O subsystem responsible for
  - Memory management of I/O including buffering (storing data temporarily while it is being transferred), caching (storing parts of data in faster storage for performance), spooling (the overlapping of output of one job with input of other jobs)
  - General device-driver interface
  - Drivers for specific hardware devices
- Interrupt handlers and device drivers are crucial in the design of efficient I/O subsystems

Protection and Security

- Protection – any mechanism for controlling access of processes or users to resources defined by the OS
- Security – defense of the system against internal and external attacks
  - Huge range, including denial-of-service, worms, viruses, identity theft, theft of service
- Systems generally first distinguish among users, to determine who can do what
  - User identities (user IDs, security IDs) include name and associated number, one per user
  - User ID then associated with all files, processes of that user to determine access control
  - Group identifier (group ID) allows set of users to be defined and controls managed, then also associated with each process, file
  - Privilege escalation allows user to change to effective ID with more rights

Exit Slips

- Take 1-2 minutes to reflect on this lecture
- On a sheet of paper write:
  - One thing you learned in this lecture
  - One thing you didn’t understand

Next session

- We will discuss:
  - Operating System Structures
- Reading assignment: (skim through before class and continue reading over the weekend)
  - SGG: Ch. 2

Acknowledgment

- The slides are partially based on the ones from
  - The book site of *Operating System Concepts (Tenth Edition)*: http://os-book.com/