Device driver

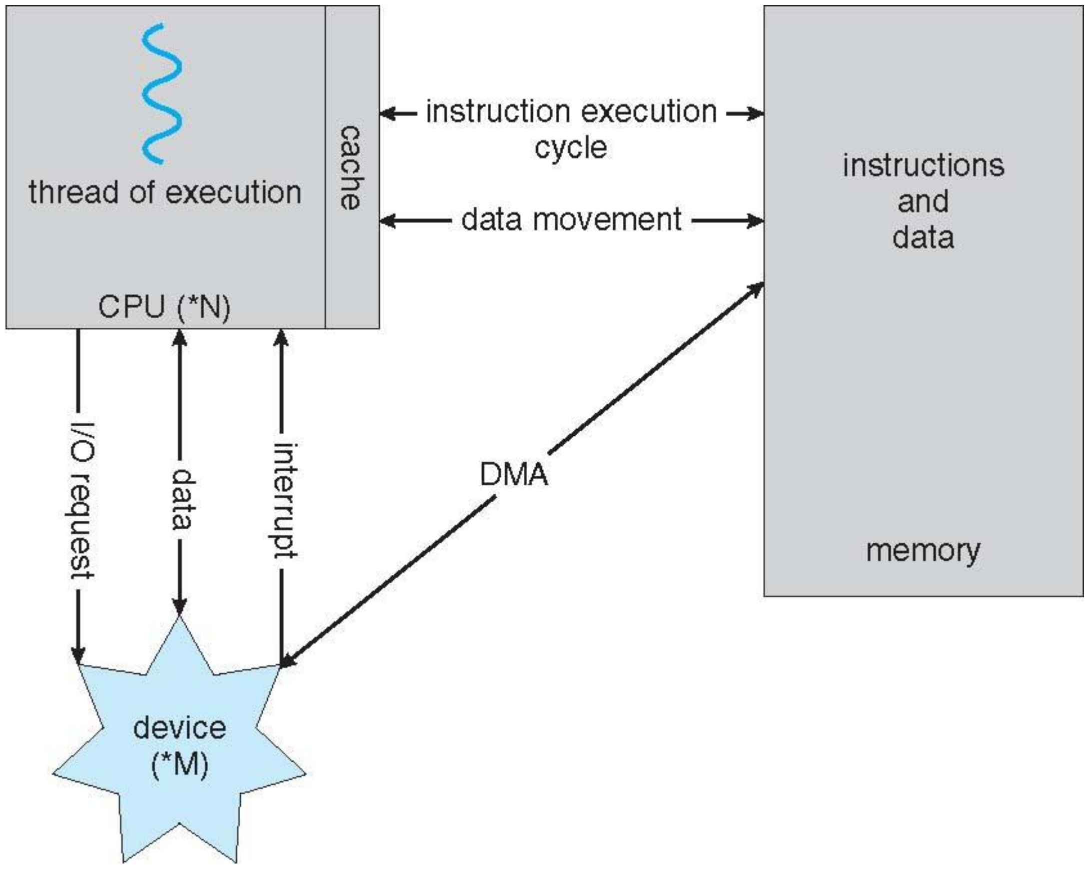
- Operating systems have device driver for each device controller.
- Device driver understands the device controller and presents a uniform interface to the device to the rest of the operating system.
- Controller transfer data from device to its local buffer, and informs the device driver about its completion via an interrupt.
- Device driver returns control to the operating system.

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.

How a Modern Computer Works



A von Neumann architecture

Interrupt

- Hardware- by sending signal to the CPU
- Software-executing special operation called a ***system call(monitor call).***
- Interrupt transfers control to the interrupt service routine generally, through the ***interrupt vector, which contains the addresses of all the service routines.***

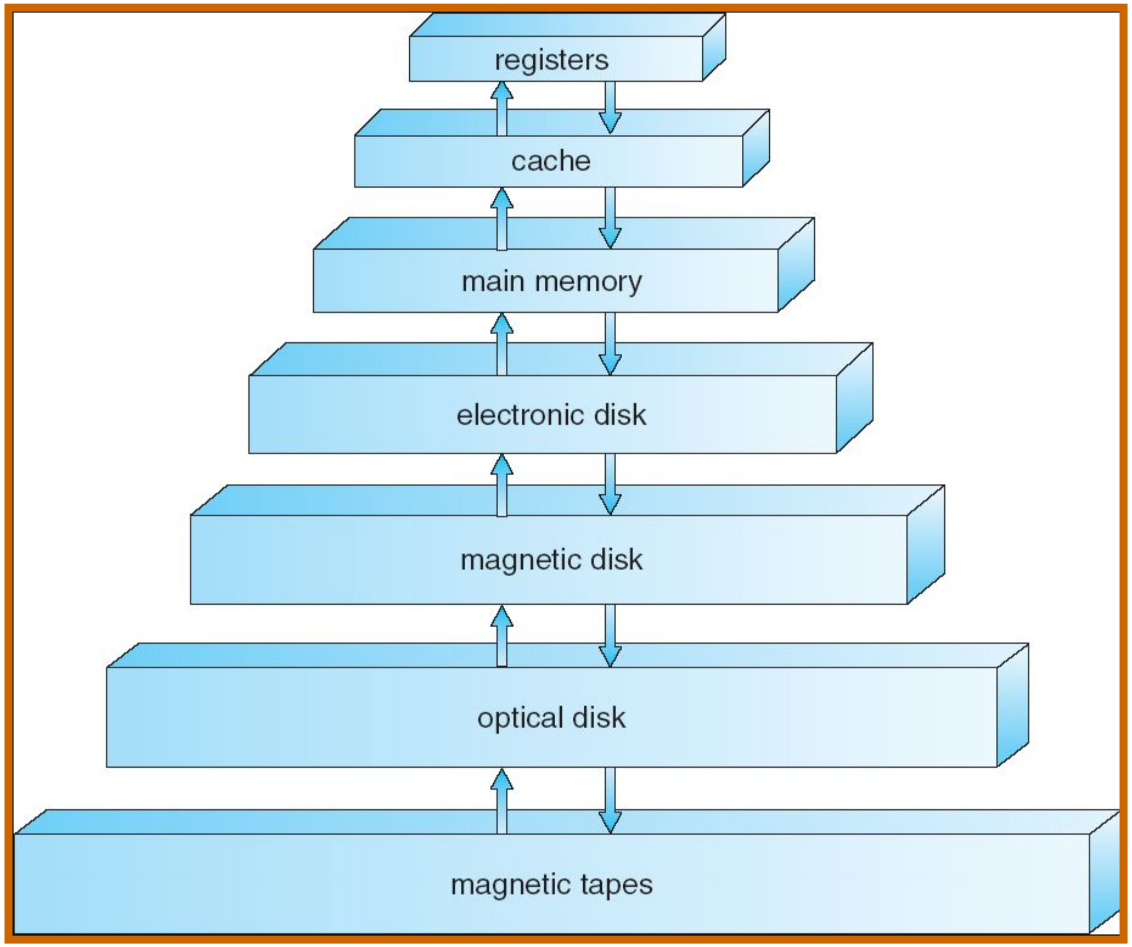
Examples of Interrupts

- Key pressed
- Disk or other I/O task finished
- System clock

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 divided into *tracks*, which are subdivided into *sectors*.
  - The *disk controller* determines the logical interaction between the device and the computer.

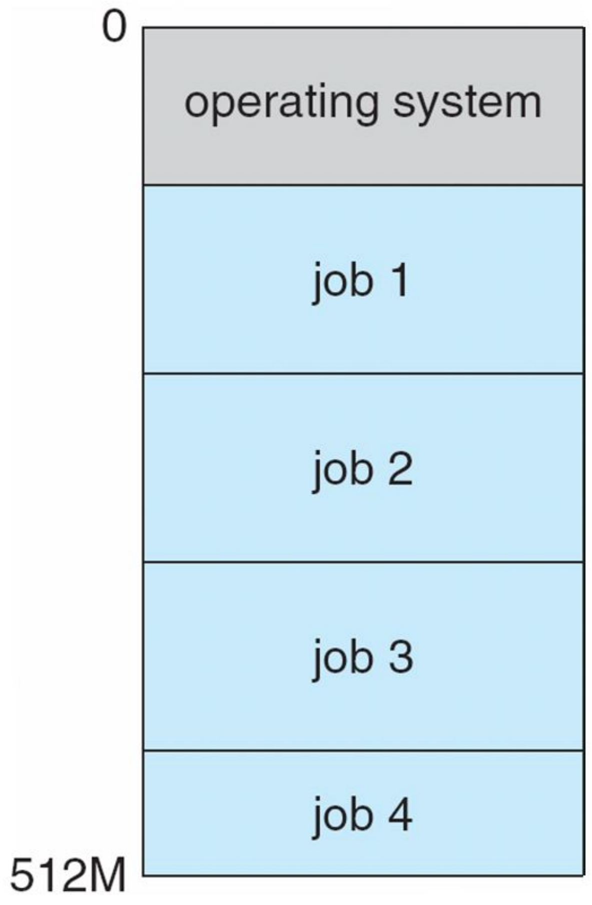
Storage-Device Hierarchy



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

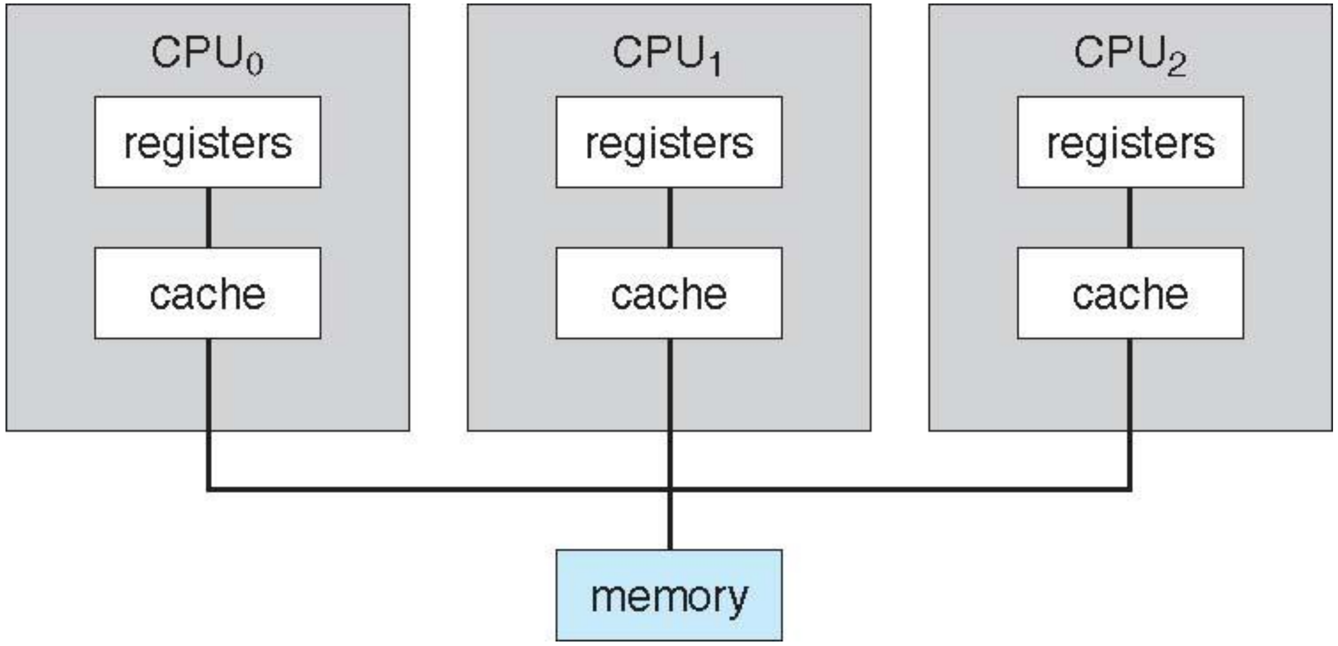
Memory Layout for Multiprogrammed System



Computer-System Architecture

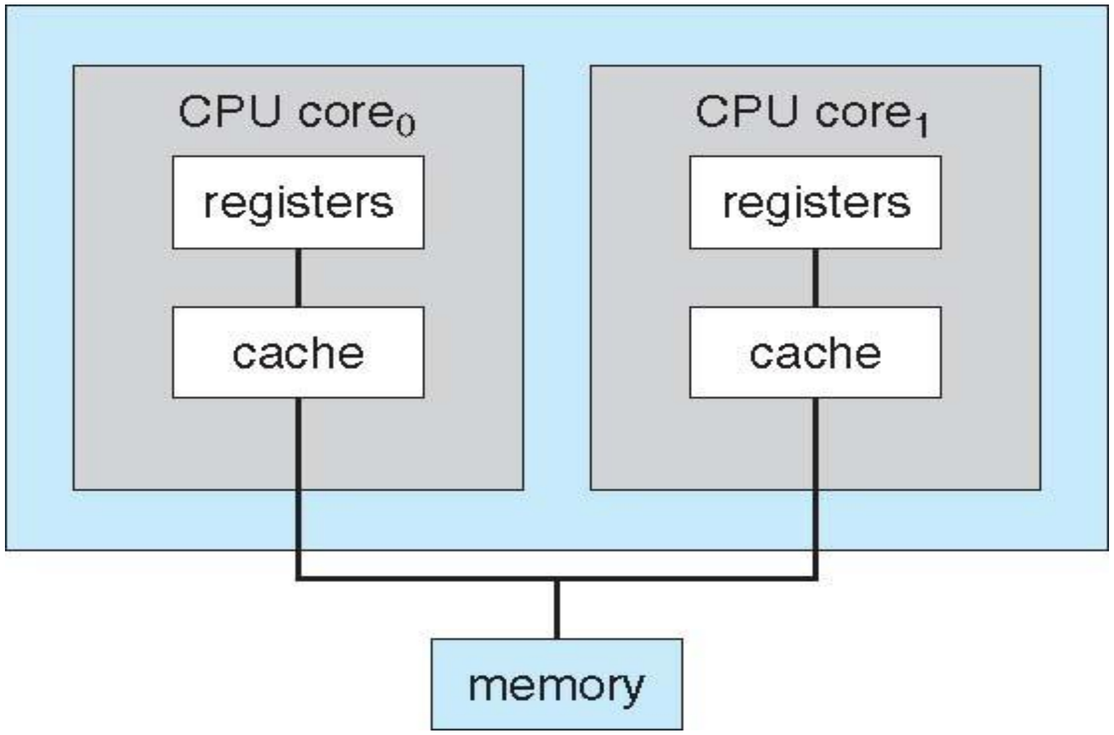
- Most systems use a single general-purpose processor
  - Most systems have special-purpose processors as well
- **Multiprocessors** systems growing –two or more processors
  - Also known as **parallel systems**, **tightly-coupled systems**
  - Advantages include:
    1. **Increased throughput**
    2. **Economy of scale-many processors work on same set of data.(can be stored on one disk)**
    3. **Increased reliability** – graceful degradation or fault tolerance
  - Two types:
    1. **Asymmetric Multiprocessing** – each processor is assigned a specie task. Master processor schedules and allocates work to the slave processors.
    2. **Symmetric Multiprocessing** – each processor performs all tasks. No master slave relatinship.

Symmetric Multiprocessing Architecture



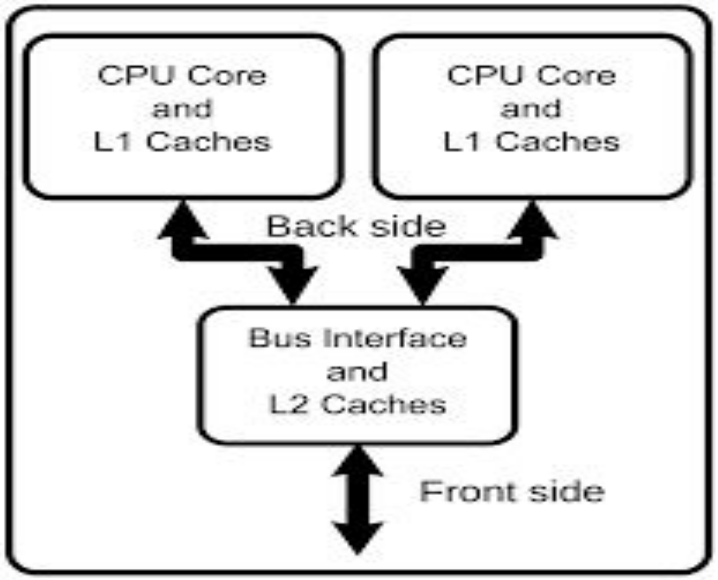
A Dual-Core Design

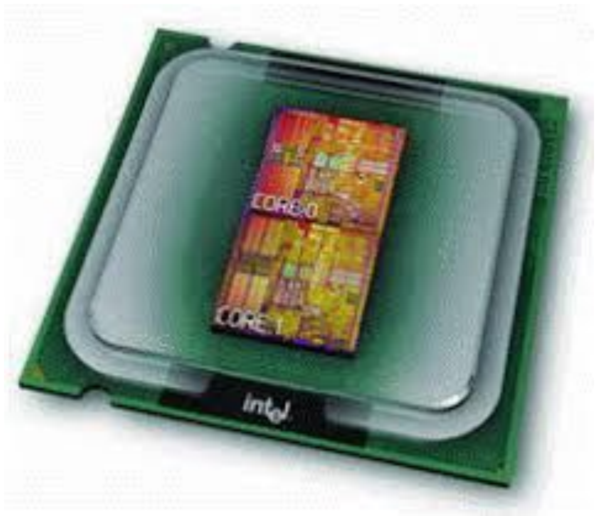
- Multi-chip and **multicore**



Dual core

INTEL®©'06 E5200
PENTIUM® DUAL-CORE
SLAY7 COSTA RICA
2.50GHZ/2M/800/06
3833A595

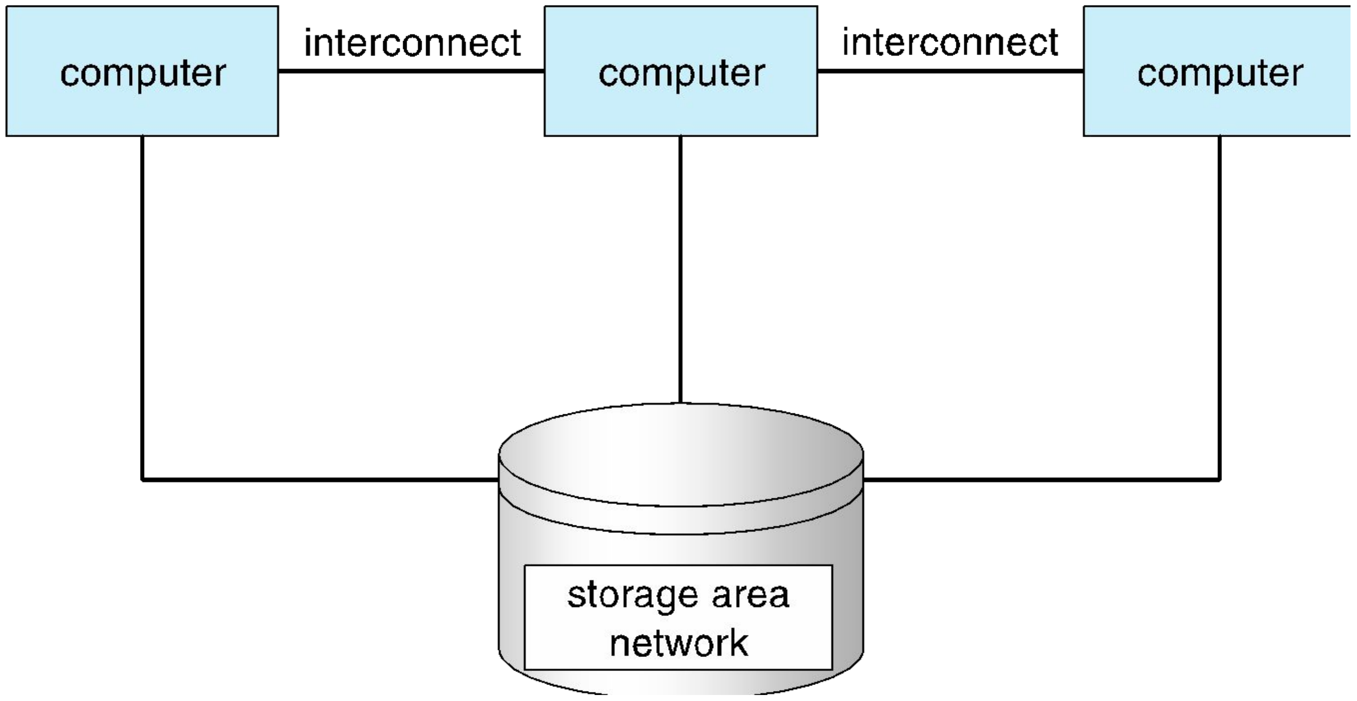

CPU Core and L1 Caches
CPU Core and L1 Caches
Back side
Bus Interface and L2 Caches
Front side


CORE 0
CORE 1
intel

Clustered Systems

- Like multiprocessor systems, but multiple systems working together
  - Usually sharing storage via a **storage-area network (SAN)**
  - Provides a **high-availability** service which survives failures
    - **Asymmetric clustering** has one machine in hot-standby mode
    - **Symmetric clustering** has multiple nodes running applications, monitoring each other
  - Some clusters are for **high-performance computing (HPC)**
    - Applications must be written to use **parallelization**
  - Some have **distributed lock manager** (**DLM**) to avoid conflicting operations

Clustered Systems



What is HPC

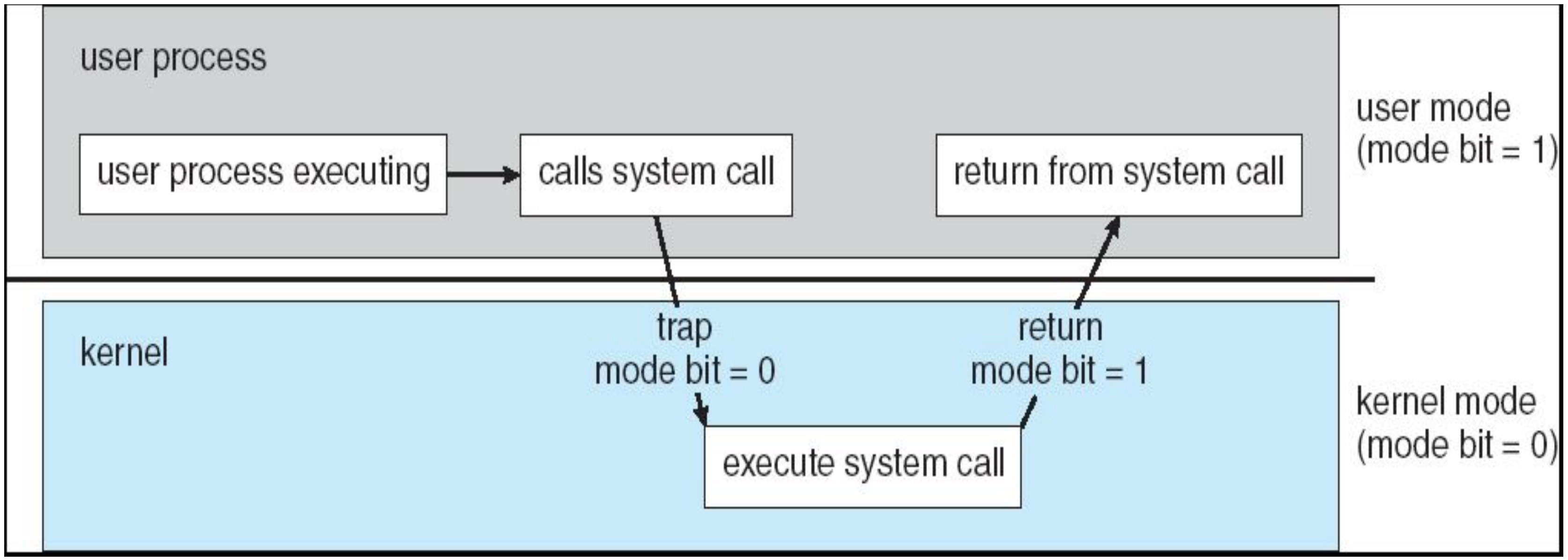
- High-performance computing (HPC) is the use of parallel processing for running advanced application programs efficiently, reliably and quickly. The term applies especially to systems that function above a teraflop or 10^{12} floating-point operations per second. The term HPC is occasionally used as a synonym for supercomputing.
- Some supercomputers work at more than a petaflop or 10^{15} floating-point operations per second.

Operating-System Operations

- Modern operating systems are interrupt driven
- 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(eg:an instruction to switch to user mode,timer management.
    - System call changes mode to kernel, return from call resets it to user

Transition from User to Kernel Mode

- Timer to prevent infinite loop
  - Set interrupt after specific period
  - Operating system decrements counter
  - When counter zero generate an interrupt



What is a Timer?

- Prevent a user program from getting stuck in an infinite loop
- Timer can be set to interrupt the computer after a specified period
- Every time the clock ticks the counter is decremented.
- When the counter reaches 0, an interrupt occurs.

Operating system functions

- Process management
- Memory management
- Storage management

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
- 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

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 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 directoriess
    - 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

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)
  - General device-driver interface
  - Drivers for specific hardware devices

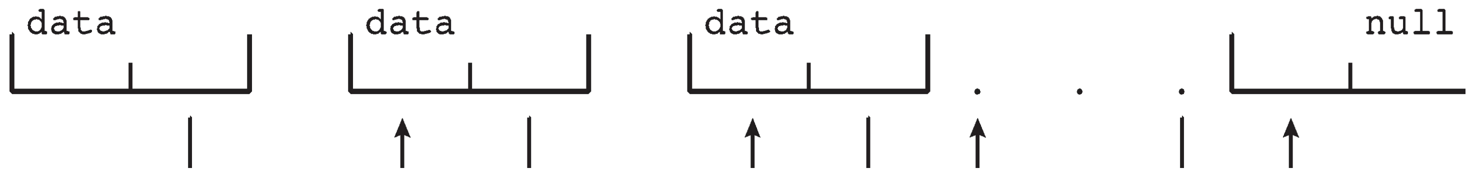
Protection and Security

- **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

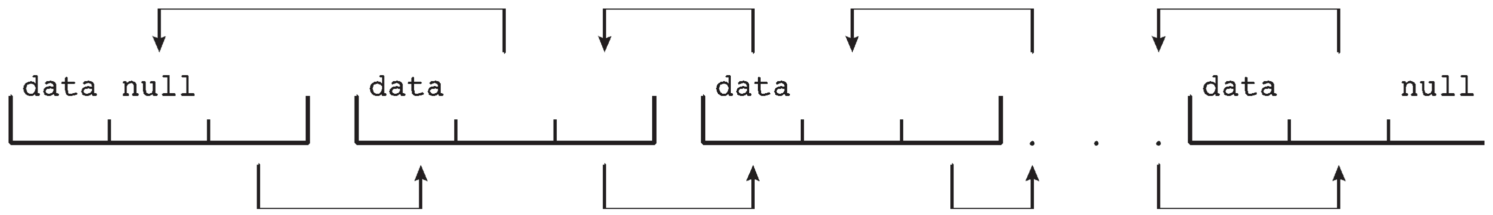
- Systems generally first distinguish among users, to determine who can do what
  - User identities (**user IDs**, security IDs) include name and associated number, one per user
  - User ID then associated with all files, processes of that user to determine access control
  - Group identifier (**group ID**) allows set of users to be defined and controls managed, then also associated with each process, file

Kernel Data Structures

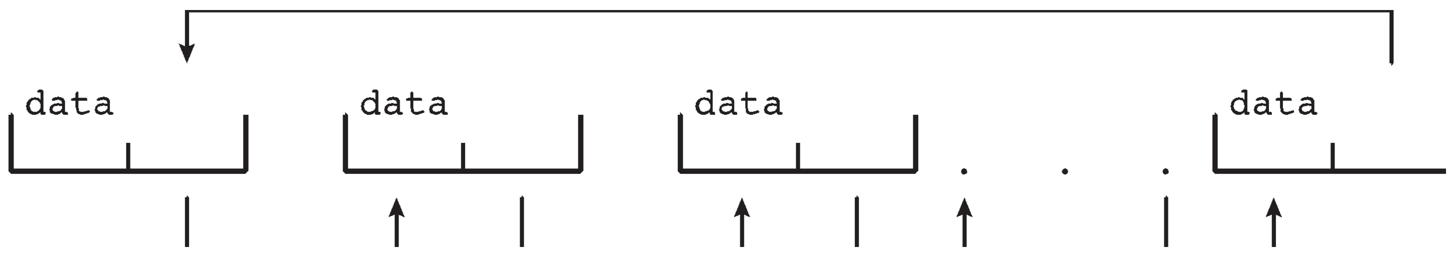
- Many similar to standard programming data structures
- ***Singly linked list***



- ***Doubly linked list***

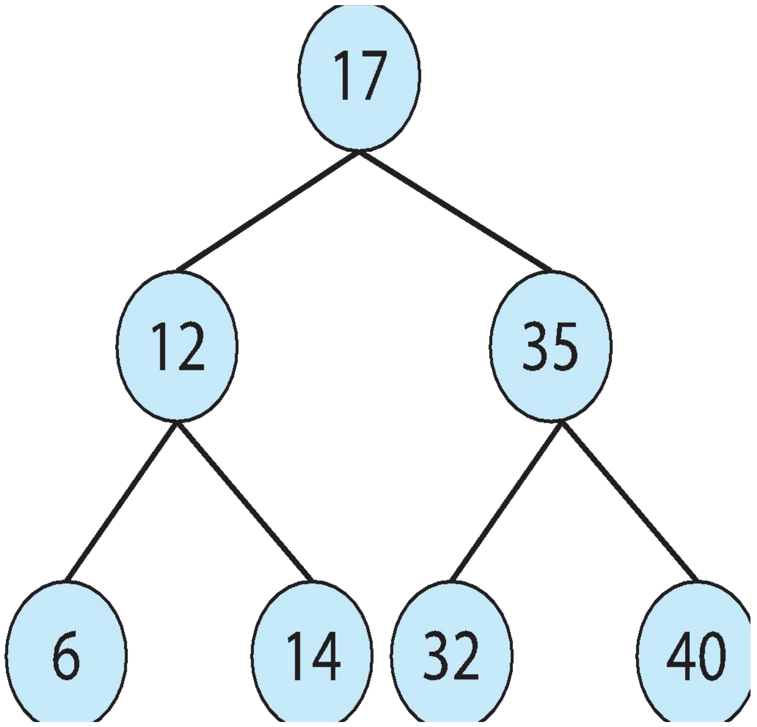


- ***Circular linked list***



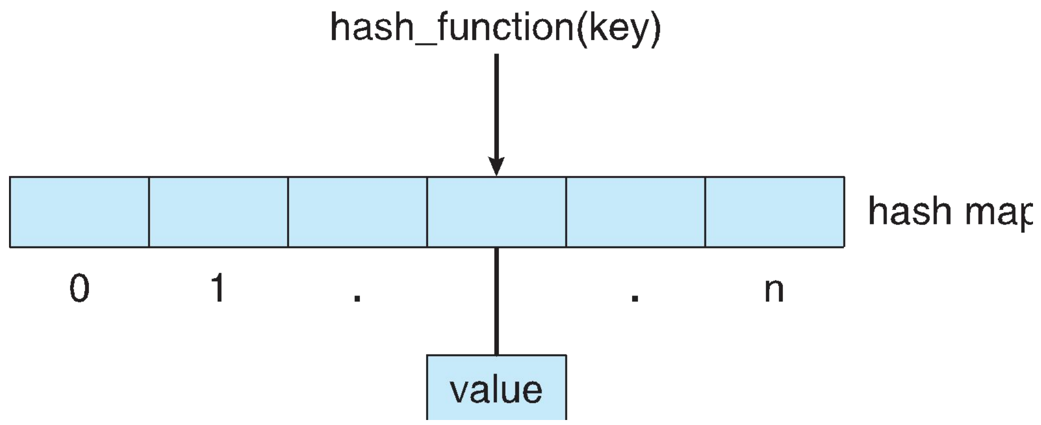
Kernel Data Structures

- **Binary search tree**



Kernel Data Structures

- **Hash function** can create a **hash map**



- Linux data structures defined in

  include files <linux/list.h>, <linux/kfifo.h>, <linux/rbtree.h>

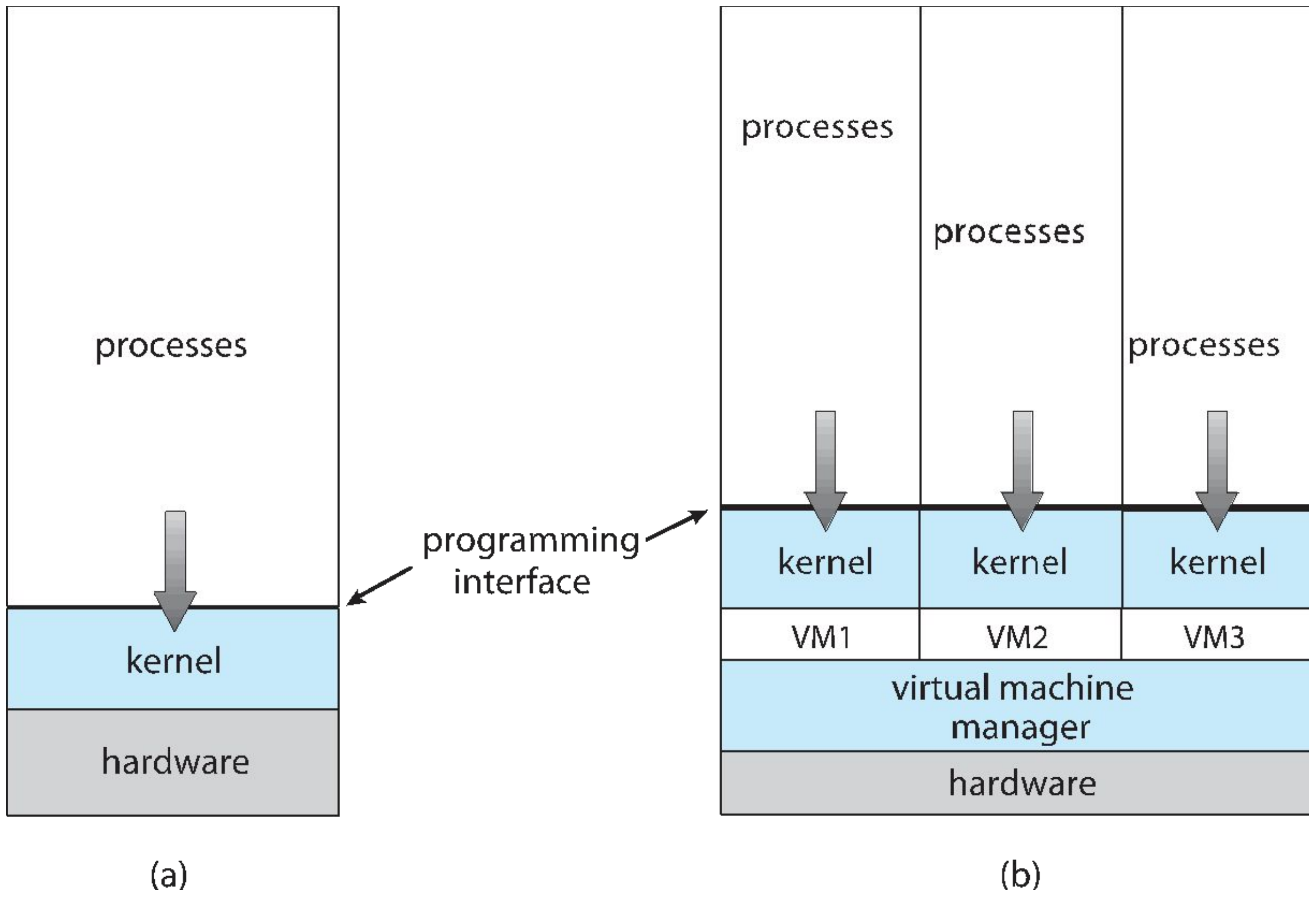
Computing Environments - Virtualization

- Allows operating systems to run applications within other OSes
  - Vast and growing industry
- **Emulation** used when source CPU type different from target type (i.e. PowerPC to Intel x86)
  - Generally slowest method
  - When computer language not compiled to native code – **Interpretation**
- **Virtualization** – OS natively compiled for CPU, running **guest** OSes also natively compiled
  - Consider VMware running WinXP guests, each running applications, all on native WinXP **host** OS
  - **VMM** (virtual machine Manager) provides virtualization services

Computing Environments - Virtualization

- Use cases involve laptops and desktops running multiple OSes for exploration or compatibility
  - Apple laptop running Mac OS X host, Windows as a guest
  - Developing apps for multiple OSes without having multiple systems
  - QA testing applications without having multiple systems
  - Executing and managing compute environments within data centers
- VMM can run natively, in which case they are also the host
  - There is no general purpose host then (VMware ESX and Citrix XenServer)

Computing Environments - Virtualization

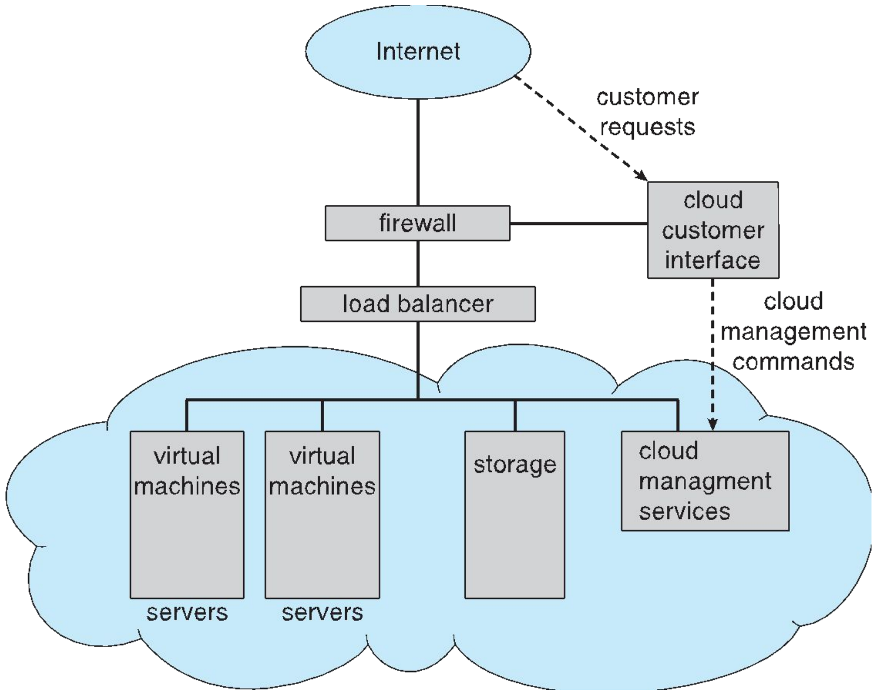


Computing Environments – Cloud Computing

- Delivers computing, storage, even apps as a service across a network
- Logical extension of virtualization because it uses virtualization as the base for it functionality.
  - Amazon **EC2** has thousands of servers, millions of virtual machines, petabytes of storage available across the Internet, pay based on usage
- Many types
  - **Public cloud** – available via Internet to anyone willing to pay
  - **Private cloud** – run by a company for the company’s own use
  - **Hybrid cloud** – includes both public and private cloud components
  - Software as a Service (**SaaS**) – one or more applications available via the Internet (i.e., word processor)
  - Platform as a Service (**PaaS**) – software stack ready for application use via the Internet (i.e., a database server)
  - Infrastructure as a Service (**IaaS**) – servers or storage available over Internet (i.e., storage available for backup use)

Computing Environments – Cloud Computing

- Cloud computing environments composed of traditional OSes, plus VMMs, plus cloud management tools
  - Internet connectivity requires security like firewalls
  - Load balancers spread traffic across multiple applications



Computing Environments – Real-Time Embedded Systems

- Real-time embedded systems most prevalent form of computers
  - Vary considerable, special purpose, limited purpose OS, **real-time OS**
  - Use expanding
- Many other special computing environments as well
  - Some have OSes, some perform tasks without an OS
- Real-time OS has well-defined fixed time constraints
  - Processing ***must*** be done within constraint
  - Correct operation only if constraints met

Open-Source Operating Systems

- Operating systems made available in source-code format rather than just binary **closed-source**
- Counter to the **copy protection** and **Digital Rights Management (DRM)** movement
- Started by **Free Software Foundation (FSF)**, which has "copyleft" **GNU Public License (GPL)**
- Examples include **GNU/Linux** and **BSD UNIX** (including core of **Mac OS X**), and many more
- Can use VMM like VMware Player (Free on Windows), Virtualbox (open source and free on many platforms - http://www.virtualbox.com)
  - Use to run guest operating systems for exploration