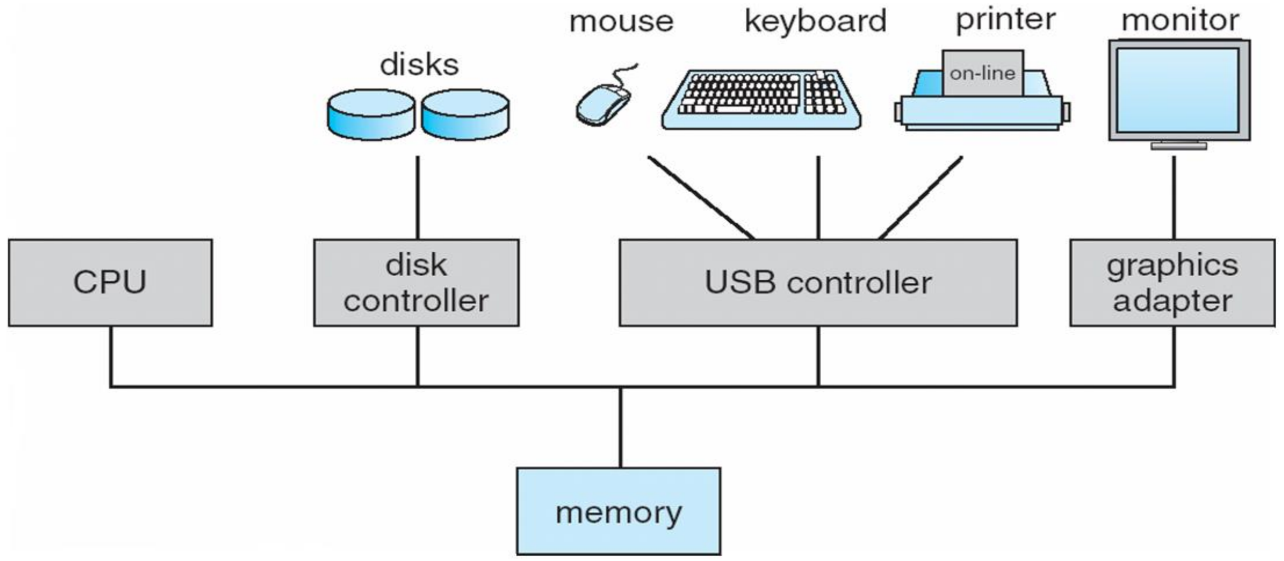
OPERATING SYSTEMS

- A modern computer consists of one or more processors, main memory, storage devices, network interfaces and various input/output devices.
- Writing programs that keep track of all these components is an extremely difficult job.

Computer System Organization

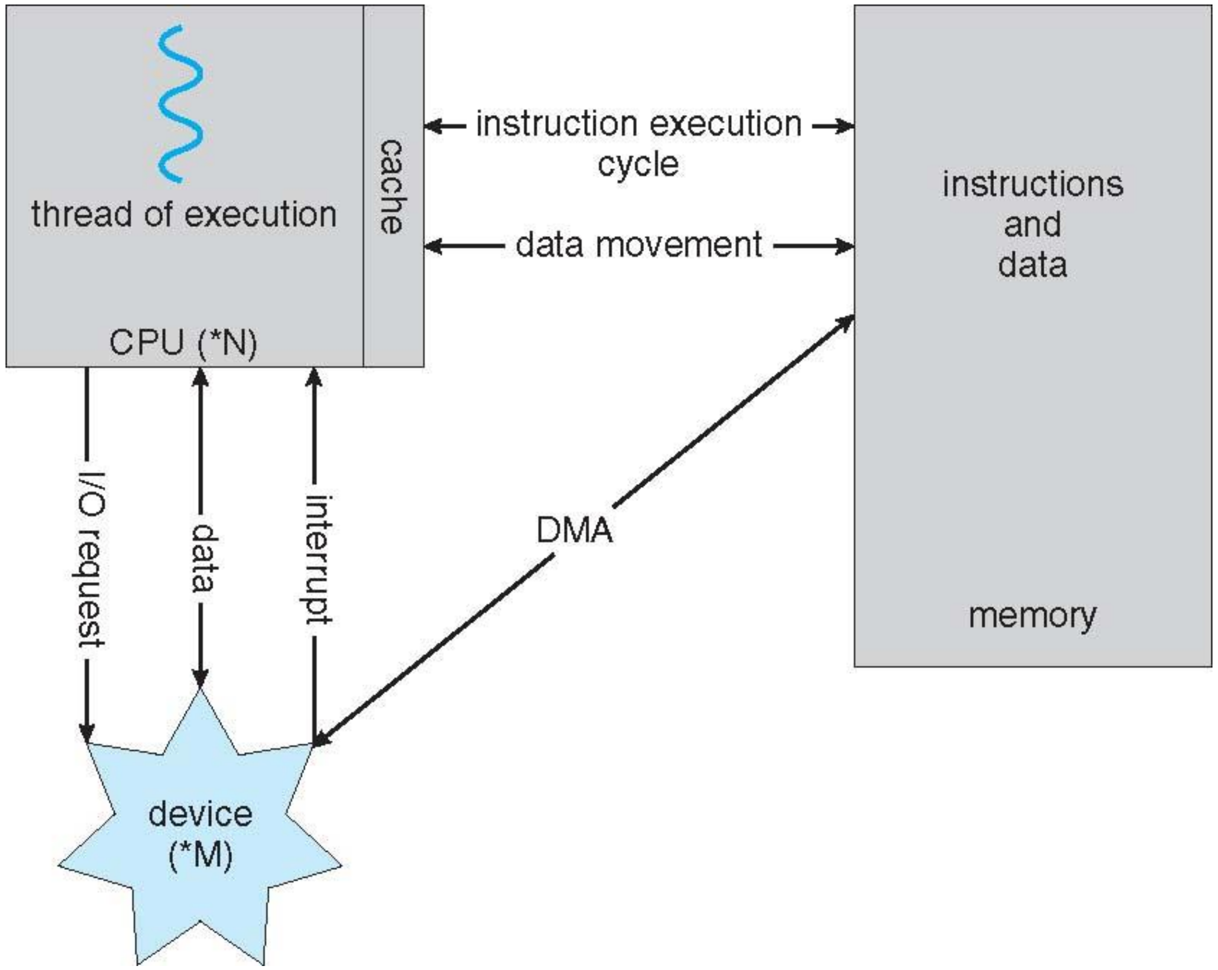
- Computer-system operation
  - One or more CPUs, device controllers connect through common bus providing access to shared memory
  - Concurrent execution of CPUs and devices competing for memory cycles



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*

How a Modern Computer Works



Components of Computer System

- **Hardware**
- **System Software**
- **Users**

Hardware

- It provides basic computing resources
  - CPU, memory, I/O devices
- It may be composed of two or more levels.
  - The lowest level consists of the basic resources.
  - The next level is the micro architecture level where physical resources are grouped together as functional units. The basic level operations involve internal registers, CPU and the operation of the data path is controlled either by micro program or by hardware control unit.
  - The next level is the Instruction set architecture level which deals with the execution of instructions.

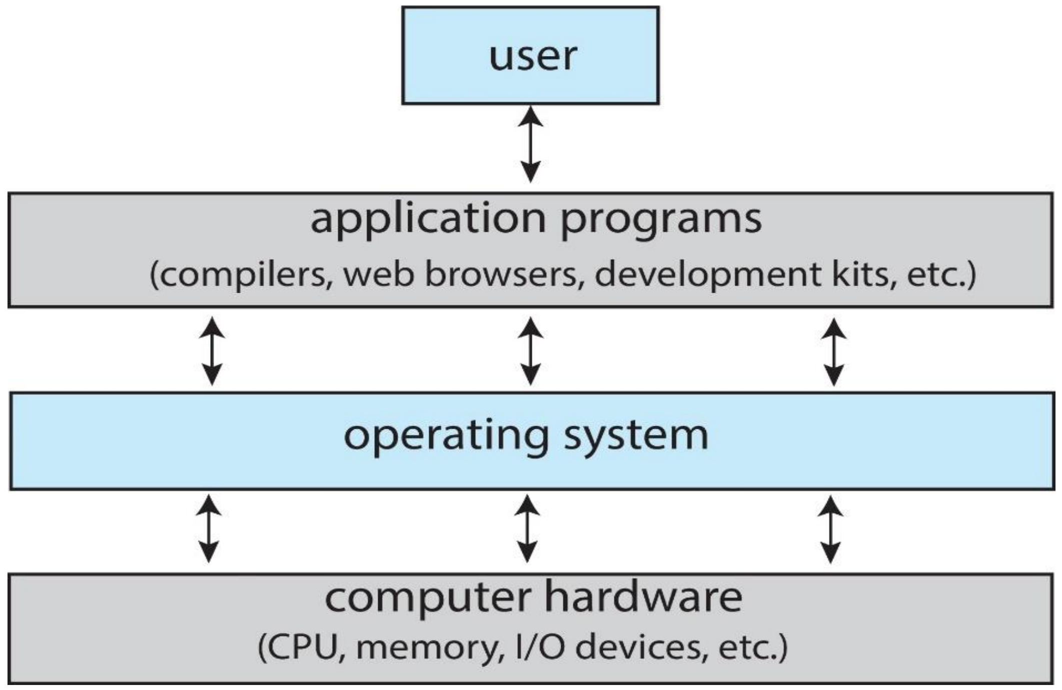
System Software:

- Operating System: It hides the underlying complex hardware. It controls and coordinates use of hardware among various applications and users.

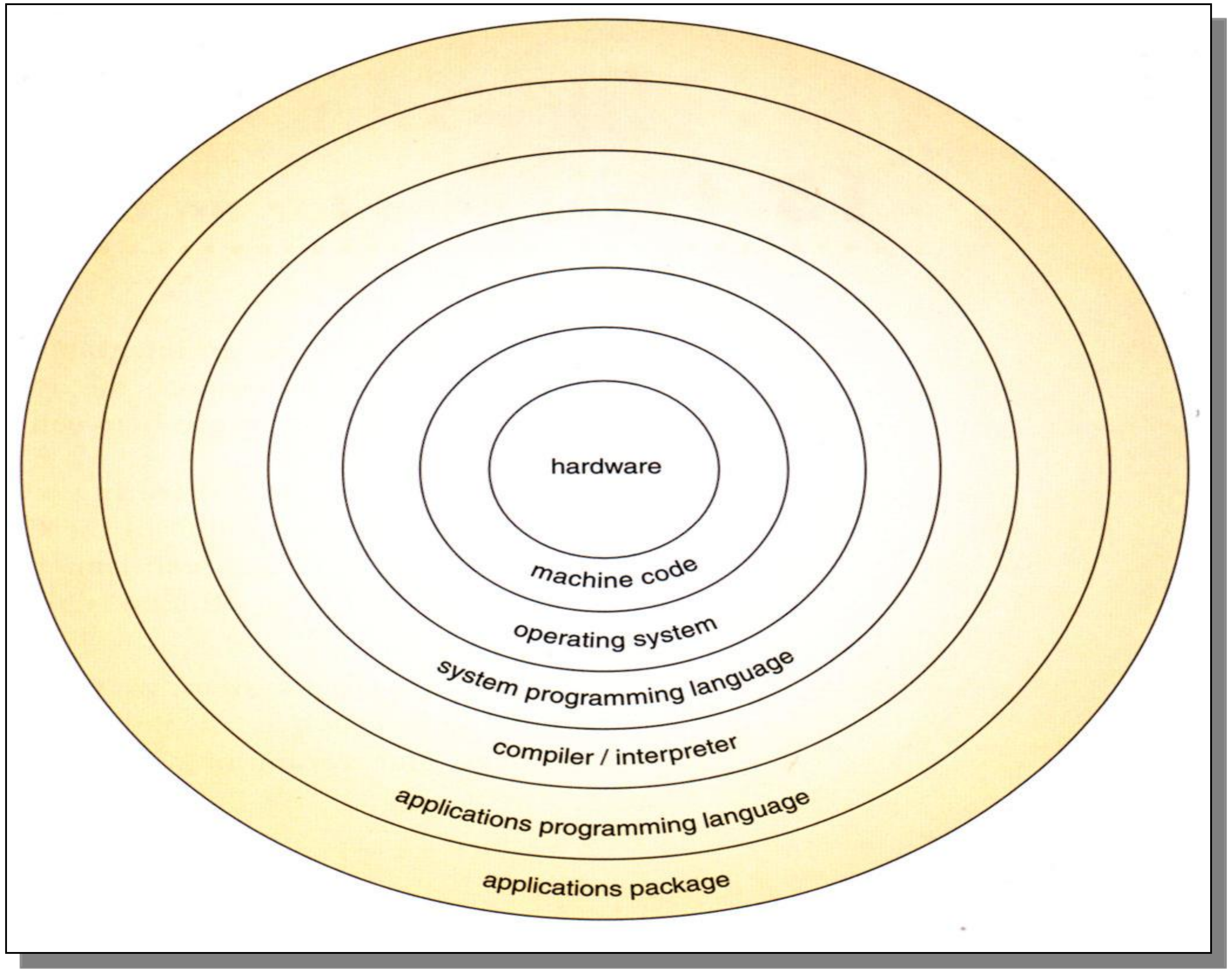
- Utilities and Application Software: define the ways of in which the system resources are used to solve the computing problems of the users. The system programs like compiler, assembler, editor etc. run in the user mode where the OS run in the supervisor mode or kernel mode.

Users: People, machines, other computers etc.

Abstract View of Components of Computer


user
application programs
(compilers, web browsers, development kits, etc.)
operating system
computer hardware
(CPU, memory, I/O devices, etc.)

Detail Layered View of Computer



Three types of programs

User / application programs

- programs used by the users to perform a task
- Performs specific tasks for users
  - Business application
  - Communications application
  - Multimedia application
  - Entertainment and educational software

System programs

- an interface between user and computer
- Performs essential operation tasks
  - Operating system
  - Utility programs

Driver programs (Device Drivers)

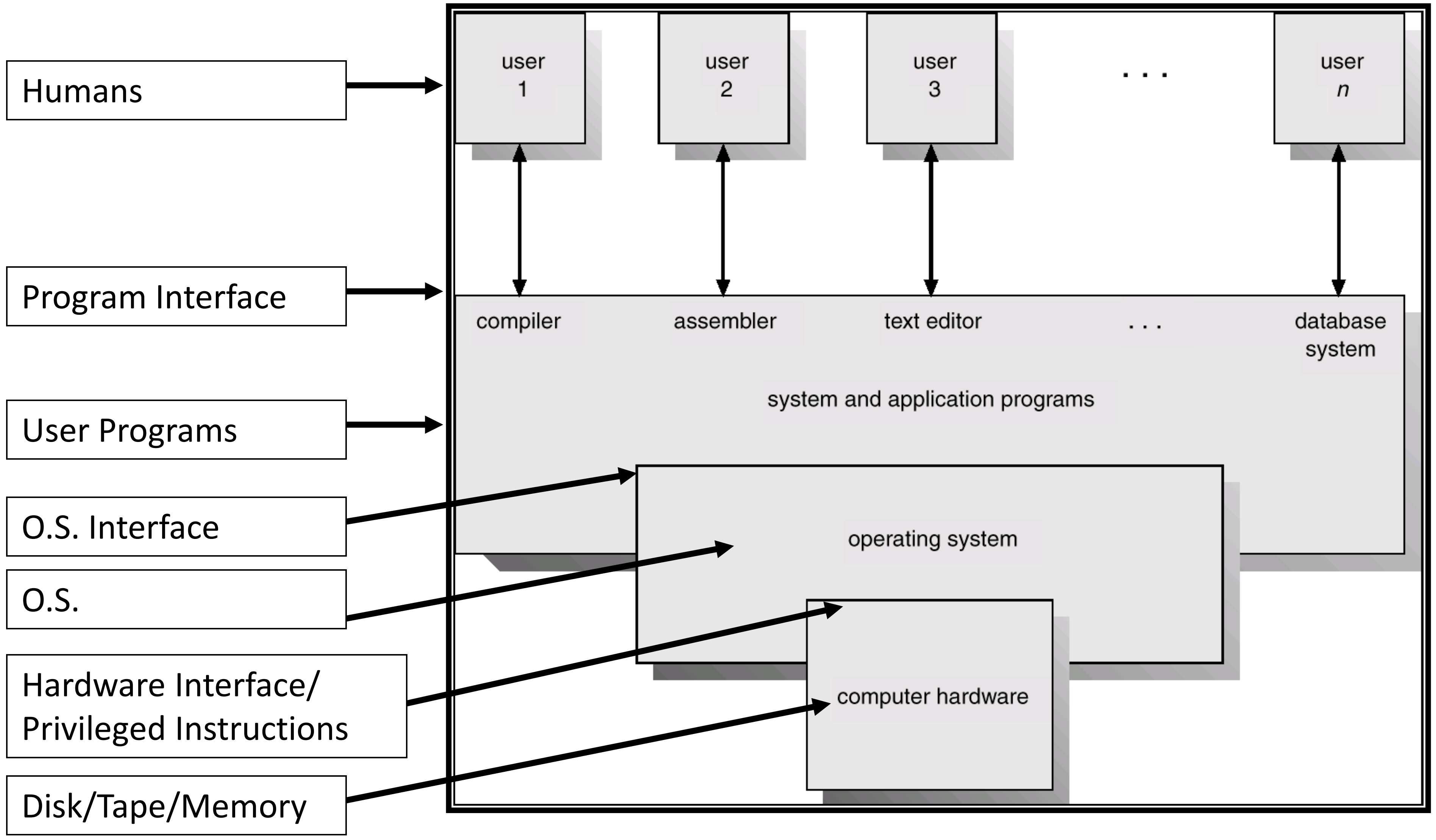
- small program that allows a specific input or output device to communicate with the rest of the computer system

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 including the CPU registers, device controllers, and memory contents
  - Locates and loads operating system kernel and starts execution of the first process (such as “init”) and waits for events to occur.

OPERATING SYSTEM OVERVIEW

The Layers of a System



OS

- Operating System (OS) is a program or set of programs, which acts as an interface between a user of the computer & the computer hardware.
- It is written in low-level languages (i.e. machine-dependent).
- When the computer is on, OS will first load into the main memory

Definitions

OS

- is a layer of software that manages all the hardware and software components in an effective and efficient manner.
- is a resource allocator that decides between conflicting requests for efficient and fair resource use.
- acts as a mediator, thereby making it easier for the programmer to access and use of the facilities and services of the computing system.
- implements a virtual machine that is easier to program than bare hardware.
- provides an abstraction of the hardware for all the user programs thereby hiding the complexity of the underlying hardware and gives the *user* a better view of the computer.
- is a control program that controls execution of programs to prevent errors and improper use of computer.

Why do we need operating systems?

- The primary need for the OS arises from the fact that user needs to be provided with services and OS ought to facilitate the provisioning of these services.
- Convenience: OS provides a high-level abstraction of physical resources; make hardware usable by getting rid of warts and specifics; enables the construction of more complex software systems and enables portable code.
- Efficiency: It shares limited or expensive physical resources and provides protection.

What is an Operating System?

- It is a *resource manager*
    - provides orderly and controlled allocation for programs in terms of time and space, *multiplexing*
    - Resource management keeps track of the resource, decides who gets the access and resolves conflicting requests for resources by enforcing policy, allocates the resource and after use reclaims the resource.

What is an Operating System?

- It is an *extended*, or *virtual*, *machine*
    - creates the functionalities of a computer in software i.e. creating copies of processors, memory etc. using software.
    - provides a simple, high-level abstraction, i.e., hides the “messy details” which must be performed
    - presents user with a virtual machine, easier to use
    - provides services; programs obtain these by *system calls*

Services Provided by the OS

- An OS provides standard **services** (an interface), such as Processes, CPU scheduling, memory management, file system, networking, etc. In addition to this, OS provides services in the following areas:
- Program development: These services are in the form of utility programs such as editors, debuggers to assist programmers in creating programs. These are referred as application development tools.
- Program execution: The tasks that are performed to execute a program are loading of instructions and data into the main memory, initializing I/O devices and files and preparing other resources.

- Access to I/O devices: Each I/O device requires its own peculiar set of instructions or control signals for operation. OS provides a uniform interface that hides these details so that the programmer can access such devices using simple reads and writes.
- Controlled access to files: For accessing files, control must include a detailed understanding of the nature of I/O device as well as the structure of the data contained in the files on the storage medium. For system with multiple users, OS should provide protection mechanisms to control access to the files.

- System access: For shared or public systems, the access function must provide protection of resources and data from unauthorised users and must resolve conflicts for resource contention.
- Error Detection and Responses: OS must provide responses to runtime errors such as internal or external hardware errors, software errors like arithmetic overflow etc. The response may be terminating the program or reporting error to the application.
- Accounting: OS collects usage statistics for various resources and monitor performance parameters to improve performance. On a multiuser system, the information can be used for billing purposes.

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
- 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 such as divide by 0 or a user request (system call).
- An operating system is **interrupt driven**

Interrupt Handling

- The operating system preserves the state of the CPU by storing registers and the program counter
- 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 without waiting for I/O completion
  - **System call** – request to the operating system to allow 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 device status and to modify table entry to include interrupt

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
- Programs and data cannot reside in main memory permanently because:
  - Main memory is limited (too small) to store all programs and data permanently
  - Main memory is volatile
- So secondary storage is provided – 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 Hierarchy

- Storage systems organized in hierarchy by
  - Speed
  - Cost
  - Volatility

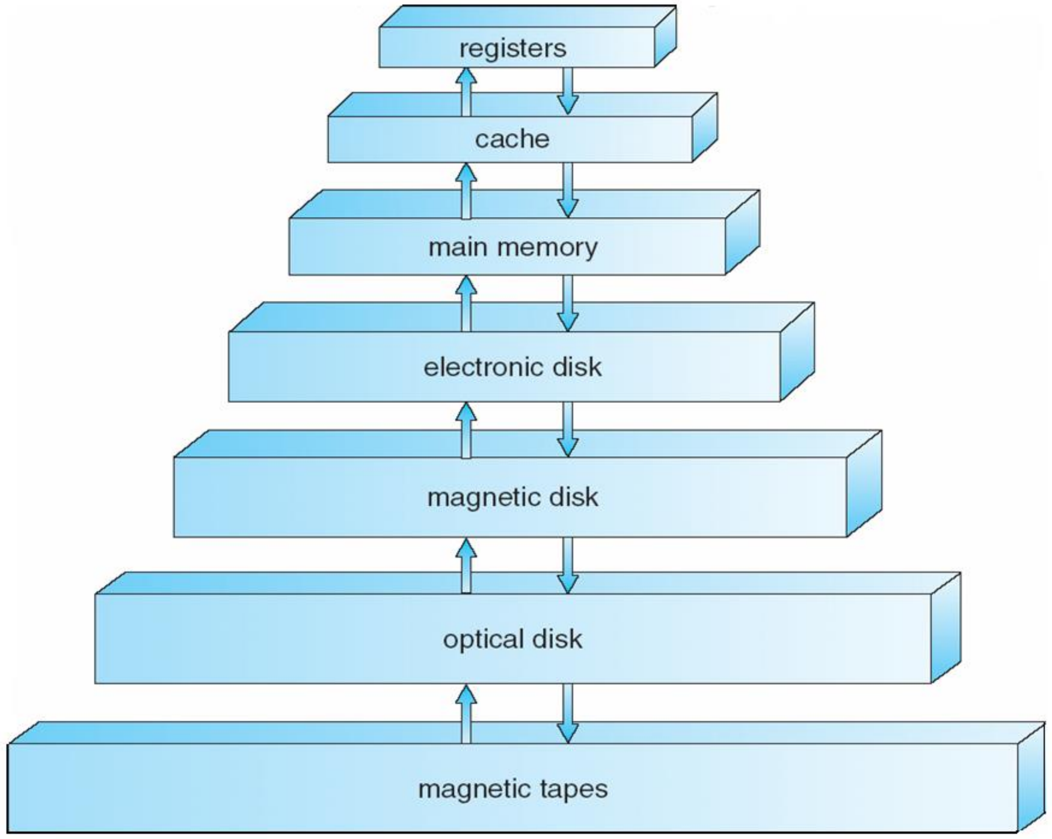
- The higher levels in the hierarchy are expensive but fast. As we move down the hierarchy, the cost per bit generally decreases, whereas the access time generally increases.

- **Caching** – copying information into faster storage system; main memory can be viewed as a last *cache* for secondary storage

Caching

- 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

Storage-Device Hierarchy



Major components of OS

1. Process Management
2. Main Memory Management
3. File Management
4. Mass Storage Management
5. I/O System Management
6. Network Management
7. Security and Protection system
8. Command Interpreter System

Process Management

- A process is an instance of a program in execution and is the fundamental unit of computation in a computer.
- A process needs certain resources, including CPU time, memory, files, and I/O devices, to accomplish its task. Processes can create sub-processes to execute concurrently.
- Programs can be executed concurrently as the number of processes active at a time may be more than one. These multiple processes which are active may communicate with each other and some may access resources which are mutually exclusive in nature.

The activities of a process manager include

- creating and deleting of user and system processes,
- suspending and resuming processes,
- scheduling processes,
- providing mechanisms for process synchronization,
- providing mechanisms for inter process communication
- handling deadlocks during concurrent execution.

Main Memory management

- Main memory is a volatile storage device. It loses its contents in the case of system failure.
- One can view memory as large array of words or bytes, each with its own address.
- It is storage of quickly accessible data shared by the CPU and I/O devices.
- A program resides on a disk as a binary executable file and it is brought to the main memory for execution.

The functions of memory management are

- keeping track of memory in use,
- allocation of memory to processes by deciding which processes and data are to be moved into and out of memory and move processes between disk and memory.

File Management

- The other names for this management are information management and storage management.
- OS provides uniform, logical view of information storage. It abstracts physical properties to logical storage unit called file.
- A file is a collection of related information defined by its creator. The information may be a sequence of bits, bytes, lines, or records whose meanings are defined by their creators. Commonly, files represent programs (both source and object forms) and data.

- The File management is responsible for allocation of space for programs and data on disk, creation and deletion of files, creation and deletion of directories, providing primitives to support for manipulating files and directories, mapping files onto secondary storage, maintaining file backup on stable storages.
- In addition, it keeps track of the information, its location, its usage, status using file system, decides who gets hold of information, enforce protection and provide access mechanism.
- Allocation and de-allocation of files are done using open and close system calls.

Mass Storage Management

- Since main memory is volatile in nature and too small to accommodate all programs and data, disks are used to store data and programs that does not fit in main memory or data that must be kept for a long period of time.
- Most modern computer systems use disks as the principle on-line storage medium, for both programs and data.

- Functions of this storage management includes free space management, storage allocation, disk scheduling, disk buffer management to reduce the disk access time as some storage devices are slow in nature.
- It also manages the tertiary storage such as optical storage devices, magnetic tapes for back up purpose. These devices vary between WORM (Write Once Read Many times) and RW (Read Write).

I/O System management

- The I/O system consists of a buffer-caching system, a general device-driver interface and drivers for specific hardware devices.
- The buffer caching system is to reduce the I/O access time.
- When a process wishes to access an I/O device it must issue a system call to OS which has device drivers to facilitate I/O functions involving I/O devices. These device drivers are software routines that control respective I/O devices through their controllers.
- The OS hides the peculiarities of specific hardware devices from the user.

- The I/O system management functions are keeping track of the I/O devices, I/O channels, etc. using I/O traffic controller.
- It performs I/O scheduling by deciding what is an efficient way to allocate the I/O resource and if it is to be shared, then deciding who gets it, how much of it is to be allocated, and for how long.
- It allocates the I/O device and initiates the I/O operation and when the use of the device is through reclaiming is done by the I/O system or in some I/O operation it terminates automatically.

Network Management

- An OS is responsible for the computer system networking via a distributed environment which allows computers to work together.
- A distributed system is a collection of autonomous computers and do not have a common clock or shared memory. Each computer has their own clock and memory and communicates with each other through networks.
- Several networking protocols such as TCP/IP (Transmission Control Protocol/ Internet Protocol), UDP (User Datagram Protocol), FTP (File Transfer Protocol), HTTP (Hyper Text Transfer protocol), NFS (Network File System) etc. are used.
- The resources like processors, memory etc., are shared and access to these shared resources improves the computation speed-up, increases availability and enhances reliability.

Security and Protection System

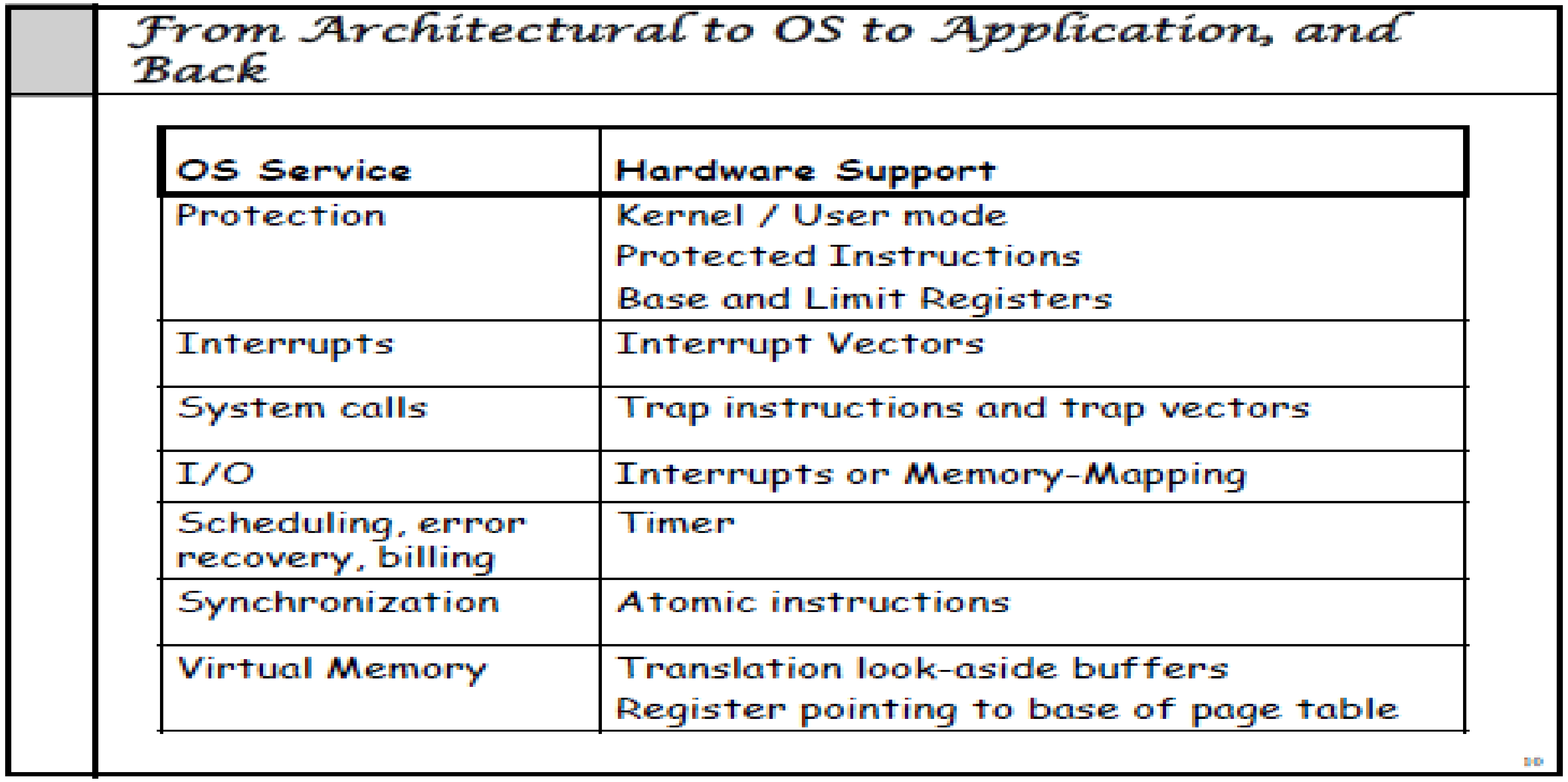
- Security refers to defense of the system against internal and external attacks which is of huge range including worms, viruses, Trojan horses, denial of services, identity theft, etc.
- OS is responsible for detecting and preventing these attacks.
- Protection refers to a mechanism for controlling access of processes or users to resources defined by OS.
- The protection mechanism provided by OS must distinguish between authorized and unauthorized usage, specify the controls to be imposed and provide a means of enforcement.

Command-Interpreter System

- A user interacts with OS via one or more user applications through a special application called a shell or command interpreter which acts as an interface between the user and the operating system.
- Today almost all OS provides a user friendly interface, namely Graphical User Interface (GUI).
- The commands given to OS are control statements that deal with process creation and management, I/O handling, secondary-storage management, main memory management, file system access, protection and networking.

From Architecture to OS to Application, and Back

Hardware	Example OS Services	User Abstraction
Processor	Process management, Scheduling, Traps, Protections, Billing, Synchronization	Process
Memory	Management, Protection, Virtual memory	Address space
I/O devices	Concurrency with CPU, Interrupt handling	Terminal, Mouse, Printer, (System Calls)
File system	Management, Persistence	Files
Distributed systems	Network security, Distributed file system	RPC system calls, Transparent file sharing



From Architectural to OS to Application, and Back

OS Service	Hardware Support
Protection	Kernel / User mode Protected Instructions Base and Limit Registers
Interrupts	Interrupt Vectors
System calls	Trap instructions and trap vectors
I/O	Interrupts or Memory-Mapping
Scheduling, error recovery, billing	Timer
Synchronization	Atomic instructions
Virtual Memory	Translation look-aside buffers Register pointing to base of page table