INTRODUCTION

Some basic questions

1. Definition of computer?
2. Father of computer?
3. Evolution of computer?
4. Generations of Computer?
5. Father of modern computer?
6. Application areas of computer?
7. Functional parts of a computer?
8. What is OS?

Computer

1. Hardware
2. Software

Types of software

1. System Software – essential for system operations
2. Application Software – developed for user applications

Definitions of OS

- It is a system software which act as an interface between the user and computer hardware

- It is the most important system software

- It act as a platform (base) on which application software can be developed and executed

- It is an organized collection of software modules that provides an effective interface between the user and computer **resources**

  Resources are hardware units like:
  - CPU
  - Main memory
  - Secondary storage
  - I/O devices
  - Files
  - Networks etc.

- OS depends on the hardware structure
  - Embedded systems
  - Mobile devices
  - PC
  - Servers
  - Mainframe etc
- Operating system is large and complex, it must be created piece by piece. Each of these pieces should be a well-delineated portion of the system, with carefully defined inputs, outputs, and functions.

Components of a computer system

1. Hardware
2. OS
3. System/ Application programs
4. User

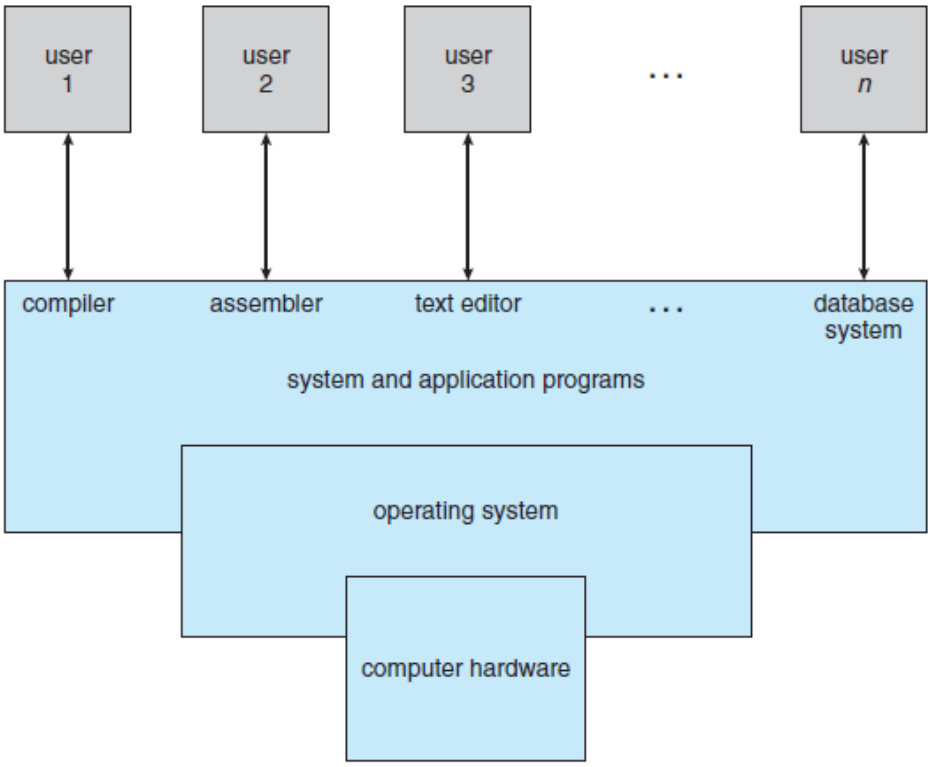

Figure 1.1 Abstract view of the components of a computer system.

Goals of OS

1. Optimize the use of resources to maximize the efficiency
2. Create a user friendly computing environment
3. Permit effective development of new programs

Computer-System Architecture

Computer systems can be categorized based on the number of general purpose processors used

1. Single processor systems
2. Multi-processor systems
3. Clustered systems

Single-Processor Systems

- Majority of systems are in this category
- Only one general purpose processor
- Can have any number of special purpose processors, like
  - Keyboard processor: convert the keystrokes into ASCII codes to be sent to the CPU
  - Graphics controller
  - I/O processor
  - Network processor etc
- They do their jobs autonomously, so that CPU can concentrate on main jobs.

Multiprocessor Systems

- Systems having two or more processors in close communication
- Also known as **parallel systems** or **multicore systems**
- Sharing the computer bus, memory and peripheral devices.
- All processors run parallel

Advantages

1. Increased throughput: (o/p obtained in unit time)
   - Work can be shared among the processors
   - More work in less time
2. Economy of scale: (when changing the size)
   - Economic when adding more processors
   - Other peripherals are sharing
3. Increased reliability
   - Even one processor fails; the system will not stop functioning
   - Remaining processors will take over the works of failed processor

Core

- Recently multi-processor systems are designed as cores
- Core is a CPU with local cache and registers together
- If 2 cores on a single chip, it is called **dual core system**
- If 4 cores, **quad core system**

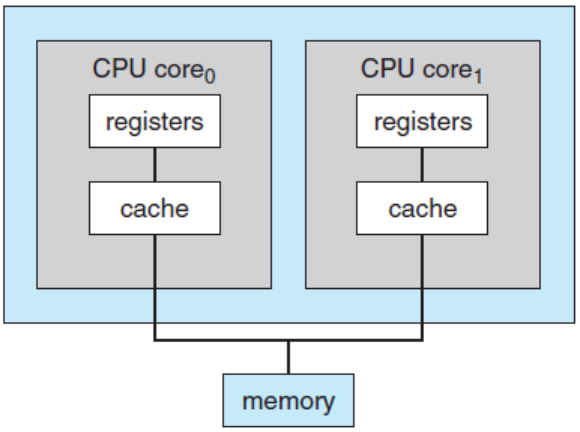

Figure 1.7 A dual-core design with two cores placed on the same chip.

Types of Multi-processor systems

1. Symmetric multiprocessing (SMP)
2. Asymmetric multiprocessing (ASMP)

SMP

- All processors run as identical copies
- Communicate each other through bus
- Peer to peer communication
- Each processor performs all tasks
- No boss–worker relationship
- All processors share physical memory

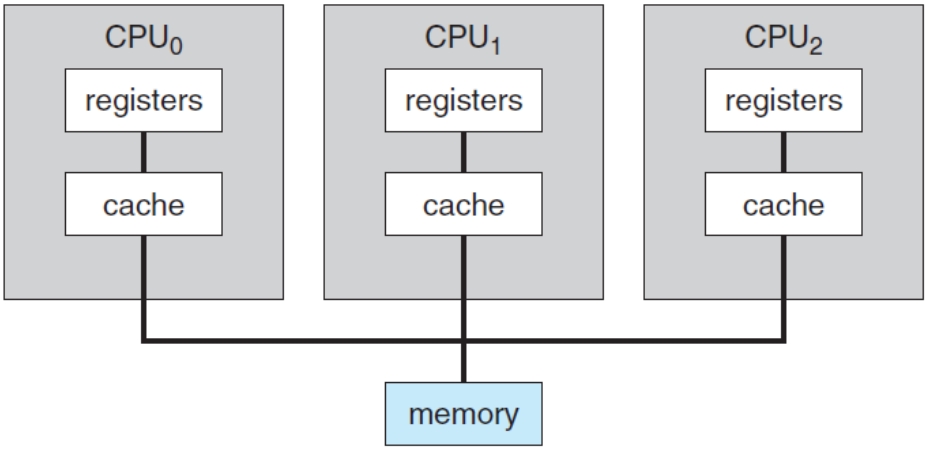

Figure 1.6 Symmetric multiprocessing architecture.

ASMP

- Each processor is assigned a specific task.
- A ***boss*** processor controls the system; the other processors either look to the boss for instruction or have predefined tasks.
- This scheme defines a boss–worker relationship.
- The boss processor schedules and allocates work to the worker processors.
- Eg: Master Slave configuration

Clustered systems

- Group of computer systems connected together through a network
- Loosely coupled systems
- Each node may be a single processor system or a multicore system.

- Clustered computers share storage and are closely linked via a local-area network LAN or a faster interconnect

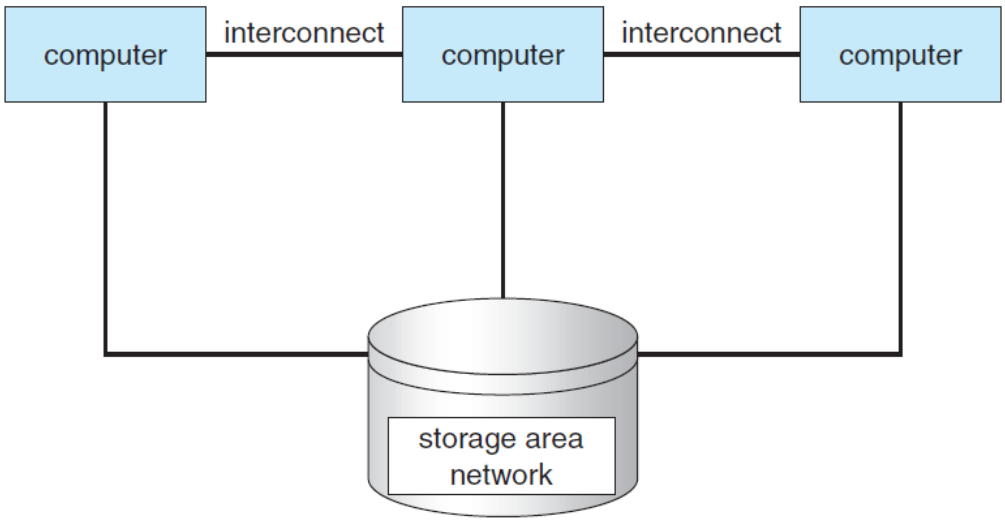

Figure 1.8 General structure of a clustered system.

Advantages

1. High availability
   - Service will continue even if one or more systems in the cluster fail.
   - Each node can monitor one or more of the others
   - If the monitored machine fails, the monitoring machine can take ownership of its storage and restart the applications that were running on the failed machine.
2. High Performance Computing (HPC)
   - Greater computational power than single-processor or even SMP systems because they can run an application concurrently on all computers in the cluster.

- This involves a technique known as parallelization, which divides a program into separate components that run in parallel on individual computers in the cluster.
- Once each computing node in the cluster has solved its portion of the problem, the results from all the nodes are combined into a final solution.

Types of Clustering

1. **Symmetric** Clustering: Two or more hosts are running applications and are monitoring each other.

2. **Asymmetric** Clustering: one machine is in **hot-standby mode** while the other is running the applications. The hot-standby host machine does nothing but monitor the active server. If that server fails, the hot-standby host becomes the active server.

Interrupts and Timer in OS

- Modern operating systems are **interrupt driven**.
- The occurrence of an event is usually signalled by an **interrupt** from either the hardware or the software.
- Hardware may trigger an interrupt at any time by sending a signal to the CPU, usually by way of the system bus.
- Software may trigger an interrupt by executing a special operation called a **system call.**

- A **trap** (or an **exception**) is a software-generated interrupt caused either by an error or by a specific request from a user program.
- An interrupt service routine is provided to deal with the interrupt.
- A timer can be set to interrupt the computer after a specified period.
- The period may be fixed (for example, 1/60 second) or variable (for example, from 1 millisecond to 1 second).
- A **variable timer** is generally implemented by a fixed-rate clock and a counter.
- The operating system sets the counter. Every time the clock ticks, the counter is decremented.
- When the counter reaches 0, an interrupt occurs.
- Before turning over control to the user, the operating system ensures that the timer is set to interrupt.
- If the timer interrupts, control transfers automatically to the operating system
- We can use the timer to prevent a user program from running too long.

Modes of Operations of OS

- **Dual mode** of OS: Two separate ***modes*** of operation:
  1. **User mode**
  2. **Kernel mode** (also called **supervisor mode**, **system mode**, or **privileged mode**).
- A bit, called the **mode bit**, is added to the hardware of the computer to indicate the current mode: kernel (0) or user (1).
- With the mode bit, we can distinguish between a task that is executed on behalf of the operating system and one that is executed on behalf of the user.
- When the computer system is executing on behalf of a user application, the system is in user mode.
- When a user application requests a service from the operating system (via a system call), the system must transition from user to kernel mode to fulfil the request.
- At system boot time, the hardware starts in kernel mode.
- The operating system is then loaded and starts user applications in user mode.
- Whenever a trap or interrupt occurs, the hardware switches from user mode to kernel mode (that is, changes the state of the mode bit to 0).
- Whenever the operating system gains control of the computer, it is in kernel mode.

- The system always switches to user mode (by setting the mode bit to 1) before passing control to a user program.
- The dual mode of operation provides us with the means for protecting the operating system from errant users
- The concept of modes can be extended beyond two modes (in which case the CPU uses more than one bit to set and test the mode).
- CPUs that support **virtualization** frequently have a separate mode to indicate (**Multi-mode**)

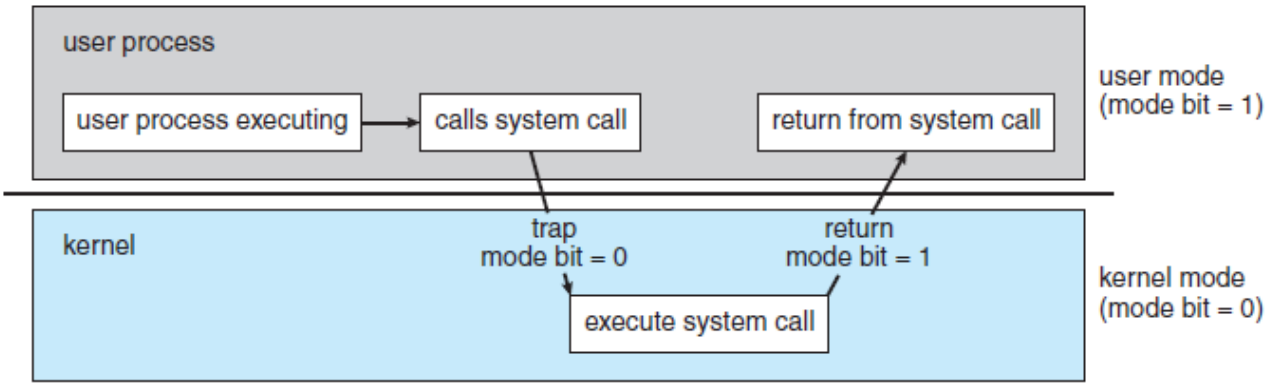

Figure 1.10 Transition from user to kernel mode.

Functions of OS

1. Process Management
2. Memory Management
3. Storage Management
4. I/O device Management
5. File Management
6. Network Management
7. System Protection & Security

1. Process Management

- Process is the instance of program under execution
- Program is a passive entity, but process is an active entity
- Process is the smallest unit of work that is independently schedulable.
- A process needs resources and this resource allocation is done by OS
- Major activities under process management are
  - Process creation: loading of program from secondary memory to primary memory
  - Process scheduling: allocating CPU for a process for execution
  - Process suspending: wait for some event or for some time
  - Process resuming: continue the suspended process
  - Process synchronization: effective running of concurrent processes without collision
  - Inter Process Communication (IPC): facilitate communication among concurrent processes
  - Deadlock handling: smooth execution of concurrent processes without a locked situation
  - Process termination: finish a process successfully or delete forcefully due to error

2. Memory Management

- Main memory is the workspace of computer
- It stores all data that are ready to be accessed by CPU

- Main memory is a large array of bytes, ranging in size from hundreds of thousands to billions. Each byte has its own address.
- For a program to be executed, it must be loaded into memory.
- When the program terminates, its memory space is declared available, and the next program can be loaded and executed.
- CPU reads instructions and data from memory and perform **read** and **write** operations during execution
- Major activities are:
  - Keeping track of which parts of memory are currently being used and who is using them
  - Deciding which processes and data to move into and out of memory
  - Allocating and deallocating memory space as needed

3. Storage Management

- Secondary memory is referred as **storage**
- Main memory is too small to accommodate all the data
- Also it is **volatile** (loses data when power loses)
- Secondary memory provides a permanent storage (non-volatile)
- Programs are saved in storage and later loaded into memory
- Usually disk structures are used for storage
- Magnetic tape drives, CD, DVD drives and platters are typical **tertiary storage** devices. (For archives and backups)

- Main activities are:
  - Free-space management
  - Storage allocation and deallocation
  - Disk scheduling
  - Mounting and unmounting media in devices
  - Tertiary device management
  - Cache memory management

4. I/O Device Management

- I/O devices are managed through device drivers
- Device driver is a software routine that controls respective I/O device. Eg: Keyboard driver, audio driver, printer driver etc.
- Device driver hides the features of I/O devices
- It forms an I/O sub-system with following activities
  - Memory-management that includes buffering, caching, and spooling
  - Provide a general device-driver interface
  - Provide Drivers for specific hardware devices

5. File Management

- It is a sub-part of storage management
- Storage may be provided by various kinds of physical devices like disks, magnetic tapes, optical devices etc
- Each of these media has its own characteristics and physical organization
- OS provides a uniform view to the user for user convenience, by hiding implementation details

- A file is a collection of related information defined by its creator.
- Commonly, files represent programs and data. Data files may be numeric, alphabetic, alphanumeric, or binary.
- Files may be free-form (for example, text files), or they may be formatted rigidly.
- Files are normally organized into directories to make them easier to use.
- Similar kinds of files are organized in the same directory/sub-directory.
- Main activities are:
  - Creating and deleting files
  - Creating and deleting directories to organize files
  - Supporting primitives for manipulating files and directories
  - Mapping files onto secondary storage
  - Backing up files on stable (non-volatile) storage media

6. Network Management

- network is a group of computers that use a set of common communication protocols over interconnections for the purpose of sharing resources
- Resource sharing: Eg: Printer sharing
- OS provides communication scheme that allows different processes on different computers exchange messages through the network.
- Network driver software is required
- Main activities are

- Establish connection
- Facilitate message passing
- Delete connection

7. System Protection and Security

- If a computer system has multiple users and allows the concurrent execution, then access to data must be regulated.
- Mechanisms ensure that files, memory segments, CPU, and other resources can be operated on by only those processes that have gained proper **authorization** from OS
- **Protection** is the mechanism for controlling the access of processes or users to the resources defined by a computer system.
- Protection can improve reliability by detecting latent errors at the interfaces between component subsystems.
- An unprotected resource cannot defend against use (or misuse) by an unauthorized or incompetent user.
- A protection-oriented system provides a means to distinguish between authorized and unauthorized usage
- **Security** is to defend a system from **external and internal attacks**.
- Such attacks spread across a huge range and include viruses and worms, denial-of-service attacks, identity theft, and theft of service.

- Prevention of some of these attacks is considered an OS function on some systems, while other systems leave it to policy or additional software.
- Due to the alarming rise in security incidents, most operating systems maintain a **list of user names and associated user identifiers** (user IDs).
- These numerical IDs are unique, one per user. When a user logs in to the system, the authentication stage determines the appropriate user ID for the user. That user ID is associated with all of the user's processes and threads.
- In some circumstances, we wish to distinguish among sets of users rather than individual users.
- For example, the owner of a file on a UNIX system may be allowed to issue all operations on that file, whereas a selected set of users may be allowed only to read the file.
- To accomplish this, we need to define a **group name and the set of users belonging to that group.**
- Group functionality can be implemented as a system-wide list of group names and group identifiers.
- A user can be in one or more groups, depending on operating-system design decisions.
- A user sometimes needs to **escalate privileges** to gain extra permissions for an activity. Eg: The user may need access to a device that is restricted.

OS OPERATIONS

- Interrupts
- Timer
- Modes of OS
    - Dual mode
    - Multi-mode

OS SERVICES

1. User Interface
2. Program execution
3. I/O operations
4. File system manipulations
5. Communication
6. Error detection
7. Resource Allocation
8. Accounting
9. Protection and security

1. User Interface (UI)

- Command line interface (CLI)
    - It is a form of interface between user and OS
    - Usually programmers use commands
    - Commands are given by the user and interpreted and executed by OS
    - Eg for DOS commands: dir, copycon, md, cls etc

- Eg for UNIX commands: cc, ls, ./ etc
- Command interpreters are also called **shells**

- **Batch interface**
  - Commands and directives to control those commands are entered into files, and those files are executed.
- **Graphical user interface (GUI)**
  - The interface is a window system with a pointing device to direct I/O, choose from menus, and make selections and a keyboard to enter text.
- Some systems provide two or all three of these variations.

2. Program execution

- The system must be able to load a program into memory and to run that program.
- The program must be able to end its execution, either normally or abnormally (indicating error).

3. I/O operations

- A running program may require I/O, which may involve a file or an I/O device.
- For specific devices, special functions may be desired (such as recording to a CD or DVD drive).
- For efficiency and protection, users usually cannot control I/O devices directly.
- Therefore, OS must provide a means to do I/O.

4. File-system manipulation

- Programs need to read and write files and directories
- They also need to create and delete them by name, search for a given file, and list file information.
- Some OS include permissions management to allow or deny access to files or directories based on file ownership.

5. Communications

- One process may needs to exchange information with another process.
- Such communication may occur between processes that are executing on the same computer or between processes that are executing on different computer systems through a computer network.
- Communications may be implemented via
  - **Shared memory**, in which two or more processes read and write to a shared section of memory, or
  - **Message passing**, in which packets of information in predefined formats are moved between processes by the OS.

6. Error detection

- OS needs to be detecting and correcting errors constantly.
- Errors may occur in the CPU and memory hardware (such as a memory error or a power failure), in I/O devices

(such as a parity error on disk, a connection failure on a network, or lack of paper in the printer), and in the user program (such as an arithmetic overflow, an attempt to access an illegal memory location, or a too-great use of CPU time).

- OS should take the appropriate action to ensure correct and consistent computing.
- Sometimes, it has to halt the system.
- Or, it might terminate an error-causing process or return an error code to a process for the process to detect and possibly correct.

First 6 services are helpful to the user. Remaining 3 services are for ensuring the efficient operation of the system

7. Resource allocation

- When there are multiple users or multiple jobs running at the same time, resources must be allocated to each of them without collision.
- **CPU-scheduling** routines take into account the speed of the CPU and the jobs that must be executed.
- There may also be routines to allocate printers, USB storage drives, and other peripheral devices.

8. Accounting

- Want to keep track of which users use how much and what kinds of computer resources.
- This record keeping may be used for accounting

9. Protection and security

(Already discussed)