Chapter 1: Introduction

Chapter 1: Introduction

- What Operating Systems Do
- Computer-System Organization
- Computer-System Architecture
- Operating-System Structure
- Operating-System Operations
- Process Management
- Memory Management
- Storage Management
- Protection and Security
- Kernel Data Structures
- Computing Environments
- Open-Source Operating Systems

Objectives

- To describe the basic organization of computer systems
- To provide a grand tour of the major components of operating systems
- To give an overview of the many types of computing environments
- To explore several open-source operating systems

What is an Operating System?

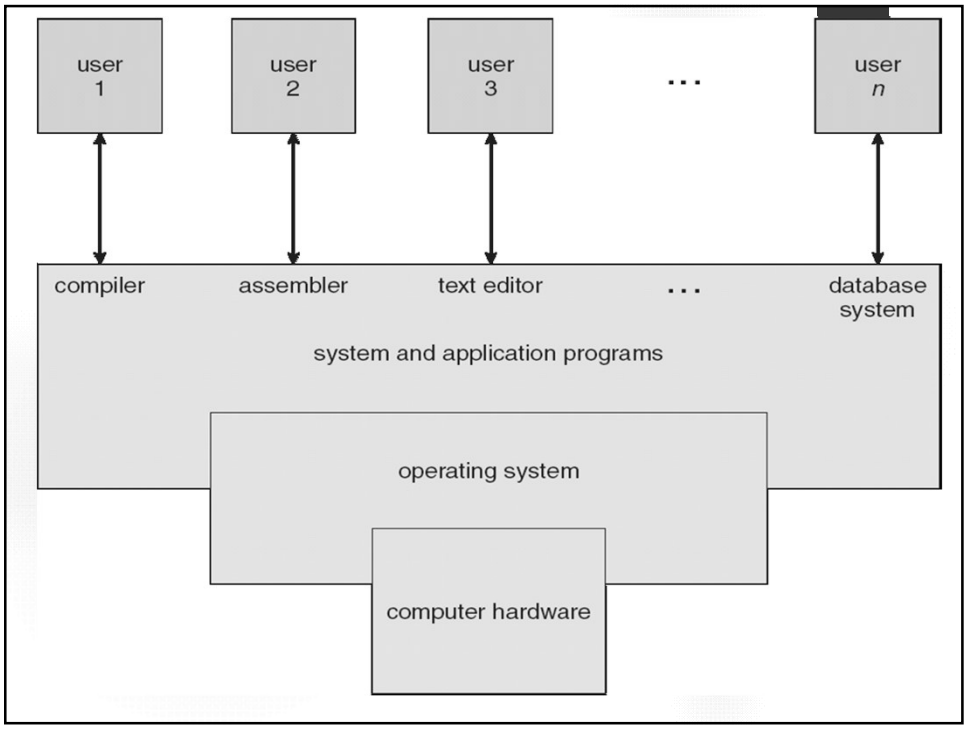
- A program that acts as an intermediary between a user of a computer and the computer hardware
- Operating system goals:
  - Execute user programs and make solving user problems easier
  - Make the computer system convenient to use
  - Use the computer hardware in an efficient manner

What is an Operating System?

- An operating system (OS) is the interface between user and computer hardware. It manages the hardware attached to the computer system.
- An operating system is a system software.
- Operating system perform various task regarding computer system like memory management, controlling peripheral devices such as disk drives and printers, handling input and output, file management etc.

Computer System Structure

- Computer system can be divided into four components:
  - Hardware – provides 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 systems, video games
  - Users
    - People, machines, other computers



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

Basic functions of operating systems:

1.Memory Management
It manages the allocation of memory of system for different processes. It manages both the primary memory and secondary memory.

2.Processor Management
It manages all the running processes in computer system.
A process is simply a program that is run by a user on computer system.

3.Security Management
It ensures the security of computer system from the various threats and viruses attacks.
An operating system uses various techniques such as authentication, authorization, cryptography etc. for ensuring security of computer system.

Basic functions of operating systems:

4.Device Management
This function of operating system is used to manage different devices that are connected with the computer system.
An operating system interact with hardware device through specified device drivers.

5.File Management
An operating system manages the files and directories of computer system.
A file can be defined as a collection of information or data that is store in the memory of computer system.
An operating system allow us to create, delete, save, edit files in a computer system.

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

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 de-allocating 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

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

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

Types of operating system:

- Many different **types of operating systems** are involved till date. The operating systems are improved in terms of their capabilities.
- The modern days operating systems allows multiple user to carry out multiple tasks simultaneously.
- Based on their capabilities and the types of application supported, the operating systems can be divided into following six major categories.

1. Batch processing operating systems
2. Multi user operating systems
3. Multitasking operating systems
4. Real time operating systems
5. Multi processors operating systems
6. Embedded operating systems

Batch processing operating system

- The batch processing operating system is capable of executing one job at a time.
- In batch processing operating system the jobs are combined in form of batch and then these batches are given to system as an input data.
- The job in batches are processed on first com first serve basis.
- After execution of one job operating system fetches another job from input data.
- There is no need of human interaction before fetching the next job after completion of any job.

Multi user operating system

- The multi user operating system uses to use a system by multiple users.
- In other word multi user operating system allows a number of users to work simultaneously on a single computer system.
- These types of operating systems are specially designed for the multi user system.
- Examples of multi user operating systems includes Unix, Linux, Windows 2000 and VM – 386.

Multitasking operating system

- The multitasking operating system allow a user to give multitask at a same time on a single computer system multitasking operating system are also known as multiprocessing operating system and multiprogramming operating system.
- The first multitasking operating system is created in 1960s.
- The number of tasks processed simultaneously by this operating system is depending upon speed of CPU, the capacity of memory and size of programs.
- Examples of multitasking operating systems are Linux, Unix, windows 2000, windows XP, windows 10

Real time operating system

- The real time operating system is similar as multitasking operating system.
- However, these operating systems are specially designed to handle real time applications.
- Real time applications are those applications which have to execute within a specific period of time.
- Therefore, time is major constraint for these applications. The different examples of real time applications are robots, machine learning etc.
- There are mainly two types of real time operating system. Hard real time operating system and soft real time operating system. Examples of real time operating systems are MTOS, Lynx, RTX

Multiprocessor operating system

- The multiprocessor operating system allows the computer system to use more than one CPU in a single system for executing more than one or multiple processes at a time.
- A computer system having multiple CPU process faster than a system which contains a single CPU.
- Examples of multiprocessor operating systems are Linux, Unix, windows 2000 etc.

Embedded operating system

- The embedded operating system is similar to real time operating system.
- This operating system is installed on an embedded computer system which is primary used to perform computational tasks in electronic devices.
- Examples of embedded operating systems are palm operating system, windows CE

Real-time operating system (**RTOS**)

- A **real-time operating system** (**RTOS**) is an operating system (OS) intended to serve real-time applications that process data as it comes in, typically without buffer delays.
- Processing time requirements (including any OS delay) are measured in tenths of seconds or shorter increments of time.
- A real time system is a time bound system which has well defined fixed time constraints.
- Processing must be done within the defined constraints or the system will fail.
- They either are event driven or time sharing.
- Event driven systems switch between tasks based on their priorities while time sharing systems switch the task based on clock interrupts.

Real Time Operating Systems

- System is nothing but group of peripherals connected to each other to process the input data and give output.
- System which is time dependent that is to process the input data and give output in given time, such systems are called real time systems.

- Real time system is divided into two systems

•Hard Real Time Systems.

•Soft Real Time Systems.

Hard Real Time Systems

- Hard real time system is purely deterministic and time constraint system for example users expected the output for the given input in 10sec then system should process the input data and give the output exactly by 10^{th} second.
- Here in the above example 10 sec. is the deadline to complete process for given data.
- Hard real systems should complete the process and give the output by 10^{th} second.
- It should not give the output by 11^{th} second or by 9^{th} second, exactly by 10^{th} second it should give the output.

- In the hard real time system meeting the deadline is very important if deadline is not met the system performance will fail.
- Another example is defense system if a country launched a missile to another country the missile system should reach the destiny at 4:00 to touch the ground what if missile is launched at correct time but it reached the destination ground by 4:05 because of performance of the system, with 5 minutes of difference destination is changed from one place to another place or even to another country.
- Here system should meet the deadline.

Soft Real Time System:

- In soft real time system, the meeting of deadline is not compulsory for every time for every task but process should get processed and give the result.
- Even the soft real time systems cannot miss the deadline for every task or process according to the priority it should meet the deadline or can miss the deadline.
- If system is missing the deadline for every time the performance of the system will be worse and cannot be used by the users.
- Best example for soft real time system is personal computer, audio and video systems, etc