

5132 – OPERATING SYSTEM

MODULE 1

CO1 Outline the significance of operating systems and its functions

Contents:

Introduction to software: System software & application software, examples-System software – assembler, loader, compilers, interpreters and their functions, comparison.

Operating system – definition – functions of Operating system - different types of Operating Systems - batch systems - multiprogramming systems - time sharing – multiprocessor systems - real time systems.

An Operating System (OS) is a collection of software that manages computer hardware resources and provides common services for computer programs. When you start using a Computer System then it's the Operating System (OS) which acts as an interface between you and the computer hardware. The operating system is really a low level **Software** which is categorized as a **System Software** and supports a computer's basic functions, such as memory management, tasks scheduling and controlling peripherals etc.

What is Operating System?

An Operating System (OS) is an interface between a computer user and computer hardware. An operating system is a software which performs all the basic tasks like file management, memory management, process management, handling input and output, and controlling peripheral devices such as disk drives and printers.

Generally, a **Computer System** consists of the following components:

- **Computer Users** are the users who use the overall computer system.
- **Application Softwares** are the softwares which users use directly to perform different activities. These softwares are simple and easy to use like Browsers, Word, Excel, different Editors, Games etc. These are usually written in high-level languages, such as Python, Java and C++.
- **System Softwares** are the softwares which are more complex in nature and they are more near to computer hardware. These software are usually written in low-level languages like assembly language and includes **Operating Systems** (Microsoft Windows, macOS, and Linux), Compiler, and Assembler etc.
- **Computer Hardware** includes Monitor, Keyboard, CPU, Disks, Memory, etc.

So now let's put it in simple words:

If we consider a Computer Hardware is body of the Computer System, then we can say an Operating System is its soul which brings it alive ie. operational. We can never use a Computer System if it does not have an Operating System installed on it.

Operating System - Examples

There are plenty of Operating Systems available in the market which include paid and unpaid (Open Source). Following are the examples of the few most popular Operating Systems:

- **Windows:** This is one of the most popular and commercial operating systems developed and marketed by Microsoft. It has different versions in the market like Windows 8, Windows 10, Windows 11 etc and most of them are paid.
- **Linux** This is a Unix based and the most loved operating system first released on September 17, 1991 by Linus Torvalds. Today, it has 30+ variants available like Fedora, OpenSUSE, CentOS, UBuntu etc. Most of them are available free of charges though you can have their enterprise versions by paying a nominal license fee.
- **MacOS** This is again a kind of Unix operating system developed and marketed by Apple Inc. since 2001.
- **iOS** This is a mobile operating system created and developed by Apple Inc. exclusively for its mobile devices like iPhone and iPad etc.
- **Android** This is a mobile Operating System based on a modified version of the Linux kernel and other open source software, designed primarily for touch screen mobile devices such as smart phones and tablets.

Some other old but popular Operating Systems include Solaris, VMS, OS/400, AIX, z/OS, etc.

System Softwares

A **compiler** is a program that converts high-level language to assembly language. Similarly, an assembler is a program that converts the assembly language to machine-level language.

Let us first understand how a program, using C compiler, is executed on a host machine.

User writes a program in C language (high-level language). The C compiler, compiles the program and translates it to assembly program (low-level language). An assembler then translates the assembly program into machine code (object). A linker tool is used to link all the parts of the program together for execution (executable machine code). A loader loads all of them into memory and then the program is executed.

Interpreter

An interpreter, like a compiler, translates high-level language into low-level machine language. The difference lies in the way they read the source code or input. A compiler reads the whole source code at once, creates tokens, checks semantics, generates intermediate code, executes the whole program and may involve many passes. In contrast, an interpreter reads a statement from the input, converts it to an intermediate code, executes it, then takes the next statement in

sequence. If an error occurs, an interpreter stops execution and reports it. whereas a compiler reads the whole program even if it encounters several errors.

Assembler

An assembler translates assembly language programs into machine code. The output of an assembler is called an object file, which contains a combination of machine instructions as well as the data required to place these instructions in memory.

Linker

Linker is a computer program that links and merges various object files together in order to make an executable file. All these files might have been compiled by separate assemblers. The major task of a linker is to search and locate referenced module/routines in a program and to determine the memory location where these codes will be loaded, making the program instruction to have absolute references.

Loader

Loader is a part of operating system and is responsible for loading executable files into memory and execute them. It calculates the size of a program (instructions and data) and creates memory space for it. It initializes various registers to initiate execution.

Cross-compiler

A compiler that runs on platform (A) and is capable of generating executable code for platform (B) is called a cross-compiler.

Source-to-source Compiler

A compiler that takes the source code of one programming language and translates it into the source code of another programming language is called a source-to-source compiler.

Operating System Definition: An operating system is a program that acts as an interface between the user and the computer hardware and controls the execution of all kinds of programs.

Functions of Operating system

Following are some of important functions of an operating System.

- Memory Management
- Processor Management
- Device Management
- File Management
- Network Management
- Security

- Control over system performance
- Job accounting
- Error detecting aids
- Coordination between other software and users

Memory Management

Memory management refers to management of Primary Memory or Main Memory. Main memory is a large array of words or bytes where each word or byte has its own address.

Main memory provides a fast storage that can be accessed directly by the CPU. For a program to be executed, it must be in the main memory. An Operating System does the following activities for memory management –

- Keeps tracks of primary memory, i.e., what part of it is in use by whom, what part is not in use.
- In multiprogramming, the OS decides which process will get memory when and how much.
- Allocates the memory when a process requests it to do so.
- De-allocates the memory when a process no longer needs it or has been terminated.

Processor Management

In multiprogramming environment, the OS decides which process gets the processor when and for how much time. This function is called **process scheduling**. An Operating System does the following activities for processor management –

- Keeps tracks of processor and status of process. The program responsible for this task is known as **traffic controller**.
- Allocates the processor (CPU) to a process.
- De-allocates processor when a process is no longer required.

Device Management

An Operating System manages device communication via their respective drivers. It does the following activities for device management –

- Keeps tracks of all devices. Program responsible for this task is known as the **I/O controller**.
- Decides which process gets the device when and for how much time.
- Allocates the device in the efficient way.
- De-allocates devices.

File Management

A file system is normally organized into directories for easy navigation and usage. These directories may contain files and other directories.

An Operating System does the following activities for file management –

- Keeps track of information, location, uses, status etc. The collective facilities are often known as **file system**.
- Decides who gets the resources.
- Allocates the resources.
- De-allocates the resources.

Other Important Activities

Following are some of the important activities that an Operating System performs –

- **Security** – By means of password and similar other techniques, it prevents unauthorized access to programs and data.
- **Control over system performance** – Recording delays between request for a service and response from the system.
- **Job accounting** – Keeping track of time and resources used by various jobs and users.
- **Error detecting aids** – Production of dumps, traces, error messages, and other debugging and error detecting aids.
- **Coordination between other softwares and users** – Coordination and assignment of compilers, interpreters, assemblers and other software to the various users of the computer systems.

Operating System Properties

Following are the different properties of an Operating System.

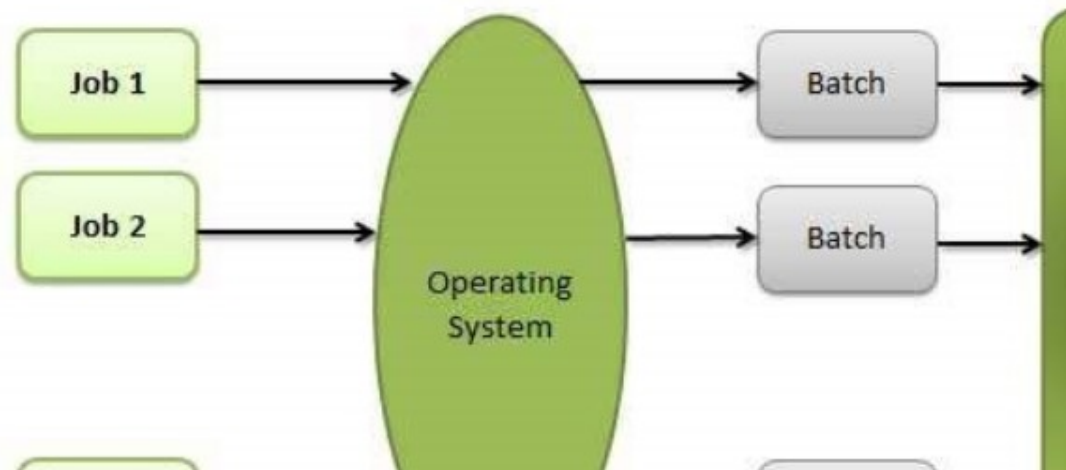
1. Batch processing
2. Multitasking
3. Multiprogramming
4. Interactivity
5. Real Time System
6. Distributed Environment
7. Spooling

Batch operating system

The users of a batch operating system do not interact with the computer directly. Each user prepares his job on an off-line device like punch cards and submits it to the computer operator. To speed up processing, jobs with similar needs are batched together and run as a group. The programmers leave their programs with the operator and the operator then sorts the programs with similar requirements into batches.

Batch processing is a technique in which an Operating System collects the programs and data together in a batch before processing starts. An operating system does the following activities related to batch processing –

- The OS defines a job which has predefined sequence of commands, programs and data as a single unit.
- The OS keeps a number of jobs in memory and executes them without any manual information.
- Jobs are processed in the order of submission, i.e., first come first served fashion.
- When a job completes its execution, its memory is released and the output for the job gets copied into an output spool for later printing or processing.



Advantages

- Increased performance as a new job get started as soon as the previous job is finished, without any manual intervention.

Disadvantages

- Lack of interaction between the user and the job.
- CPU is often idle, because the speed of the mechanical I/O devices is slower than the CPU.
- Difficult to provide the desired priority.
- Difficult to debug program.
- A job could enter an infinite loop.
- Due to lack of protection scheme, one batch job can affect pending jobs.

Time-sharing operating systems

Time-sharing is a technique which enables many people, located at various terminals, to use a particular computer system at the same time. Time-sharing or multitasking is a logical extension

of multiprogramming. Processor's time which is shared among multiple users simultaneously is termed as time-sharing.

The main difference between Multiprogrammed Batch Systems and Time-Sharing Systems is that in case of Multiprogrammed batch systems, the objective is to maximize processor use, whereas in Time-Sharing Systems, the objective is to minimize response time.

Multiple jobs are executed by the CPU by switching between them, but the switches occur so frequently. Thus, the user can receive an immediate response. For example, in a transaction processing, the processor executes each user program in a short burst or quantum of computation. That is, if n users are present, then each user can get a time quantum. When the user submits the command, the response time is in few seconds at most.

Advantages of Timesharing operating systems are as follows –

- Provides the advantage of quick response.
- Avoids duplication of software.
- Reduces CPU idle time.

Disadvantages of Time-sharing operating systems are as follows –

- Problem of reliability.
- Question of security and integrity of user programs and data.
- Problem of data communication.

Distributed operating System

Distributed systems use multiple central processors to serve multiple real-time applications and multiple users. Data processing jobs are distributed among the processors accordingly.

The processors communicate with one another through various communication lines (such as high-speed buses or telephone lines). These are referred as **loosely coupled systems** or distributed systems. Processors in a distributed system may vary in size and function. These processors are referred as sites, nodes, computers, and so on.

The advantages of distributed systems are as follows –

- With resource sharing facility, a user at one site may be able to use the resources available at another.
- Speedup the exchange of data with one another via electronic mail.
- If one site fails in a distributed system, the remaining sites can potentially continue operating.
- Better service to the customers.
- Reduction of the load on the host computer.
- Reduction of delays in data processing.

Network operating System

A Network Operating System runs on a server and provides the server the capability to manage data, users, groups, security, applications, and other networking functions. The primary purpose of the network operating system is to allow shared file and printer access among multiple computers in a network, typically a local area network (LAN), a private network or to other networks.

Examples of network operating systems include Microsoft Windows Server 2003, Microsoft Windows Server 2008, UNIX, Linux, Mac OS X, Novell NetWare, and BSD.

The advantages of network operating systems are as follows –

- Centralized servers are highly stable.
- Security is server managed.
- Upgrades to new technologies and hardware can be easily integrated into the system.
- Remote access to servers is possible from different locations and types of systems.

The disadvantages of network operating systems are as follows –

- High cost of buying and running a server.
- Dependency on a central location for most operations.
- Regular maintenance and updates are required.

Real Time operating System

A real-time system is defined as a data processing system in which the time interval required to process and respond to inputs is so small that it controls the environment. The time taken by the system to respond to an input and display of required updated information is termed as the **response time**. So in this method, the response time is very less as compared to online processing.

Real-time systems are used when there are rigid time requirements on the operation of a processor or the flow of data and real-time systems can be used as a control device in a dedicated application. A real-time operating system must have well-defined, fixed time constraints, otherwise the system will fail. For example, Scientific experiments, medical imaging systems, industrial control systems, weapon systems, robots, air traffic control systems, etc.

There are two types of real-time operating systems.

Hard real-time systems

Hard real-time systems guarantee that critical tasks complete on time. In hard real-time systems, secondary storage is limited or missing and the data is stored in ROM. In these systems, virtual memory is almost never found.

Soft real-time systems

Soft real-time systems are less restrictive. A critical real-time task gets priority over other tasks and retains the priority until it completes. Soft real-time systems have limited utility than hard real-time systems. For example, multimedia, virtual reality, Advanced Scientific Projects like undersea exploration.