

---

## UNIT 2 OPERATING SYSTEM

---

Structure	Page No.
2.0 Introduction	46
2.1 Objectives	47
2.2 The Concept of an Operating System	47
2.3 Operating System Services	50
2.3.1 Command Processor and User Interface	50
2.3.2 File Management	51
2.3.3 Input/Output Services	52
2.3.4 Process Control Management	52
2.3.5 Memory Management	52
2.4 A Bit of History	53
2.5 Summary	57
2.6 Answers to Check Your Progress	57
2.7 Further Readings	57

---

### 2.0 INTRODUCTION

---

An operating system is system software which may be viewed as an organized collection of software consisting of procedures for operating a computer and providing an environment for execution of programs. It acts as an interface between users and the hardware of a computer system.

There are many important reasons for studying operating systems. Some of them are:

- 1) User interacts with computer through the operating system in order to accomplish his/her task since it is his/her primary interface with a computer.
- 2) It helps users to understand the inner functions of a computer very closely.
- 3) Many concepts and techniques found in the operating system have general applicability in other applications.

The introductory concepts of an operating system will be the main focus in this unit. The unit introduces several OS concept such as command interface, bootstrap loader, kernel, nanokernel, thin clients, multitasking and multiprogramming OS. To understand OS one needs to understand what kinds of services are provided by process management, I/O device management, memory management and file system. This topic is an important part of the unit. The end of unit goes into a brief history of evolution of OS, starting from serial processing, batch processing till date.

## 2.1 OBJECTIVES

After going through this unit, you should be able to:

- describe Operating Systems concepts;
- list components of an operatin system;
- Describes services provided by OS (Operating System) components;
- Describe in brief evolution of an operating system.

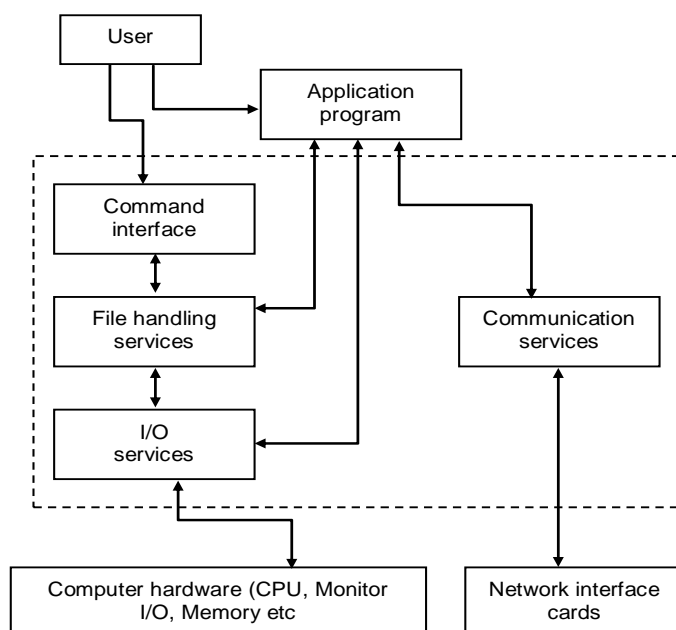
## 2.2 THE CONCEPT OF OPERATING SYSTEM

An operating system is an essential software component of a computer system. The basic objectives of an operating system are to make the computer system convenient to use and to utilize computer hardware in an efficient manner.

We can define an operating system as a large collection of software, which manages the resources of the computer system, such as **memory, processor, file system** and **input/output devices**. It keeps track of the status of each resource and decides which will have control over computer resources, for how long and when. In doing so, it provides two basic types of services:

1. It accepts requests from users and from the user's programs and executes their commands.
2. It optimally manages the hardware resources of the computer which may include CPU, main memory, hard disks, I/O devices, network interface card, etc.

The relationship between the various components of a computer system is shown schematically in Figure 2.1.



The figure 2.1 shows the **command interface** as part of the operating system. In some systems, this is viewed as a shell outside of the operating system.

In a real situation, the operating system does more complex tasks: It controls not only one I/O device but many such devices; it allows many programs to share the H/W resources of a system and not one. There could be many CPUs in a system. Operating system has to optimally manage all of them. So how does operating system execute a command? In order to execute a command OS does the following steps:

- It accepts a command from a user through mouse clicks or from the keyboard. Through a port number it is able to recognize what is the source of input.
- It must interpret these commands and take actions accordingly.
- It must provide a file system that can interpret the name of the program being requested and determine the location of the file on secondary storage device for example.
- It must read the appropriate blocks from the device into memory. Only then can the operating system transfer control to the program being executed.

In case of multiple programs executing simultaneously, CPU does some extra jobs to allot the CPU time in an equitable way to each program, and also maintain the integrity of each program, etc.

With this example one can make out what kind of facilities operating system provides to the users and programs:

- The operating system provides **interfaces** for the user (keyboard, mouse, clicks, and pen drive) and also for the user's programs.
- It provides file system support to manage huge volume of data in to secondary storage device.
- It provides I/O services that can be used by every program.
- It provides **boot-strapping** or **Initial Program Load (IPL) to start a computer**
- It manages all kinds of errors and also supports error recovery mechanisms
- It provides networking services.
- It provides the environment (allocation of memory, I/O devices and CPU time) for **concurrent processing**. As you are aware, a single processor is capable of executing only one instruction at a time. Concurrent processing is the means used to simulate the simultaneous execution of multiple programs to provide multitasking and multiuser support.

The operating system is commonly divided into **resident** and **non resident parts**. Some operating system services are critical to the operation of the system and must be **resident** in memory all the time. Others can be loaded into memory only when they are needed, and executed just like other programs.

The critical programs are loaded into memory by the **bootstrap loader** at start-up time and will remain resident as long as the computer is running. The bootstrap for most modern computers is stored in read-only memory; on some computers, part of the resident operating system will also be contained in ROM, so that it is permanently resident in memory and always available for use. The memory resident components of an operating system are commonly known as the **kernel** of the operating system. For example, the operating system program accepts user commands must always be present, as well as the programmes that handle request from the user systems and manage resources in a multitasking system. On the other hand, an operating system command that formats a new disk is only used occasionally; it can be loaded and executed only when it is required. Today, the trend is towards design of **nano kernel** or **micro kernel**. The essence of these kernels is to keep the basic kernel smaller.

Most people assume that the operating system is stored on a disk that is connected directly to the computer, but this is not necessarily true. If the computer is attached to a network, it may obtain its programs, including the operating system, from another computer on the network. This has led to the concept of the diskless workstation, a personal computer that relies completely on the network for its data and program storage and access. **Diskless workstations** are also known as **thin clients**. The size of the kernel and the particular services provided within a kernel vary from one operating system to another operating system, depending on the organization and capabilities of the system, as well as by the type of system.

- UNIX, WINDOWS 2000, WINDOWS XP and Linux are examples of operating systems that operate on a variety of different hardware platforms (i.e., microprocessor). There is a strong advantage providing a standard operating system that works on different hardware platforms. Such a system provides portability for programme as well as for file, therefore it is easy for the users to move comfortably from one machine to another. They are also called **multiuser** and **multiprogramming** or **multitasking operating** system which makes it possible for many users or tasks to share the computer resources, providing fuller utilization of the system resources.
- The operating system extends the capability of the computer to include features that require special coordinated hardware and software that is invisible to the user. These features include virtual memory, cache memory, multiprocessing.
- Systems can also be categorized by the degree of activity between the user and the system during program execution. As a student, you are probably most familiar with **online, interactive systems**. “Online” simply means that the user is connected directly to the computer. When the system is interactive, the user interacts directly with the program to provide input data and guidance during program execution. This is called **online processing**. Interactive systems are sometimes known as **conversational systems**.
- Many business tasks such as banking operations are performed more effectively in a batch, where the data input for the program is collected together into a file on disk or tape. It does not make sense to have a user enter

data one record at a time if an entire set of data is to be sorted, for example. Instead, the user *submits* the program(s), or **job(s)**, to the computer for processing. This type of processing is known as **batch processing**. The user does not interact with the program during batch processing. The OS system user interface provides a means for the user to get work done more quickly and efficiently. This is especially true for the user interface such as **menu**, icons found on modern operating systems. The modern operating systems combine **graphical user interface** simplicity with sophisticated text command input capability to provide the user with powerful access to the facilities of the computer.

### Check Your Progress 1

1. Define the following terms:

(i) Online Processing:

.....  
.....  
.....

(ii) Batch Processing:

.....  
.....  
.....

2. How does multiprogramming improve performance of a system?

.....  
.....  
.....

---

## 2.3 OPERATING SYSTEM SERVICES

---

In this section we consider basic services supported by an operating system. There are many building blocks to be considered, not all of which will necessarily be found in any particular operating system. Therefore, we will consider few essential ones.

- Command processor and user interface
- File management system
- Input/output control system
- Process management
- Memory management

### 2.3.1 Command Processor and User Interface

To the user, the most important and visible services provided by the OS is the user interface and the capabilities that it provides to execute commands which may not be a part of OS.

Instead these systems consider the user interface as a separate shell that is provided with the operating system and that interacts with the kernel to provide the necessary-user command capabilities. In UNIX, three different shells, the C shell, the Bourne shell, and the Korn shell are in common use, and many other shells for UNIX are available. Each of these shells provides different command structures and different capabilities.

Different types of user interface exist. The most common are the **graphical user interface**, or **GUI**, and the **command line interface**. The graphical user interface accepts commands primarily in the form of drop-down menus, mouse movements, and mouse clicks. The command line interface relies on typed commands which provide direct access to various methods within operating system such as File system, I/O system, and network services. UNIX allows certain class of users called **superusers** to use some kind of commands for changing the platform or access rights.

Command languages extend the power and flexibilities of the operating system and simplify use of the system for less sophisticated users.

### 2.3.2 File Management

The concept of a file is central to the effective use of a computer system. A file is generally loosely defined as a collection of related information such as students records employee database. It might contain graphical usage. A file may be organized internally into records or it may simply be a stream of bytes. A file constitutes a *logical unit* of storage, that is, logical to the person or program using the file.

The file management system provides and maintains the mapping between a file logical storage needs and the physical location where it is stored. Users and programs simply access the files by the name, and the file management system handles the details. The file management system identifies and manipulates files by the names provided by their users determines the physical requirements of the file, allocate space for it, stores it in that space, and maintains the information about the file so that it may be retrieved partially or in full, later. The file management system keeps track of the available space on each device connected to the system. The user and the user's program need not be aware of the underlying physical storage issues.

The file management system allows the retrieval and storage of files by name, keeps track of the mappings, allocates and frees space, allows the mounting and unmounting of file structures, and provides other functions required to maintain the structures of the file system.

- Directory structures for each I/O device in the system and tools to access and move around these structures. The directory structure provisions are made to move easily from one structure to another.
- It also protects files and limit file access to authorized users.

File management systems are particularly important in systems in which secondary storage devices are shared in common by multiple users, since they provide a directory system that assures that there is no duplicate use of physical storage.

### 2.3.3 Input/Output Services

Every operating system, large or small, provides input/output services for each device in the system. The operating system includes I/O device driver programs for each device installed on the system. These drivers provide services to the file management system and are also available, through the API, to other programs for their use. The I/O device drivers accept I/O requests and perform the actual data transfers between the hardware and specified areas of memory.

Devices drivers for newly installed devices are added and integrated into the operating systems. In Windows, this capability is known as **plug-and-play**.

### 2.3.4 Process Control Management

A **process** is an executing program. It is considered the standard unit of work within a computer system. *Every* executing program is treated as a process. This includes not only application programs, but the programs within the operating system itself. The process concept considers the program, together with the resources that are assigned to it, including memory, I/O devices, time for execution, and the like. When admitted to the system, each program is assigned memory space and the various resources that it initially requires to complete its work. As the process executes, it may require additional resources, or it may release resources that it no longer needs. The operating system performs various functions with processes, including scheduling and memory management, by providing the various services. Processes must often be synchronized, so that processes sharing a common resource do not step on each other's toes by altering critical data or denying each other needed resources. Systems also provide communication capability between different processes. Processes may cooperate with each other by sending messages back and forth using **interprocess messaging services**.

In multitasking operating system. OS determines which job will be admitted to the system and in what order. This process is called job scheduling. Many modern systems further break the process down into smaller units called **threads**. A thread is an individually executable part of a process. It shares memory and other resources with all other threads in the same process, but can be scheduled to run separately from other threads.

### 2.3.5 Memory Management

The purpose of the memory management system is to load programs into memory in such a way as to give each program loaded the memory that it requires for execution.

In multiprogramming operating system there are many programs residing in the memory simultaneously in the memory.

The memory management system has three primary tasks. It attempts to perform these tasks in a way that is fair and efficient to the programs that must be loaded and executed.

1. It keeps track of which parts of the memory are currently being used and by which process into memory together with the space being used and also keeps track of available space.
2. It maintains one or more queues of programs waiting to be loaded into memory as space becomes available, based on such program criteria as priority and memory requirements.
3. When space is available, it allocates memory to the programs that are next to be loaded. It also de-allocates a program's memory space when it completes execution. The de-allocated space is made available for other programs.

### Check Your Progress 2

1. What are the important components of an OS?

.....

.....

.....

2. What is job scheduling?

.....

.....

.....

3. What is a thread?

.....

.....

.....

---

## 2.4 A BIT OF HISTORY

---

In attempting to understand the key requirement for an operating system and the significance of the major features of a contemporary operating system, it is useful to consider how operating systems have evolved over the years.

### Serial Processing

During the late 1940s to the mid-1950s, the programmer interacted directly with computer hardware; there was no operating system. These machines were run by operators from a console, consisting of displays lights, toggle switches, some



form of input device and a printer. Programs in machine code were loaded via a card reader.

This mode of operation could be termed **serial processing**, reflecting the fact that users have access to the computer in series. Gradually, various system software tools were developed which included libraries of common functions, linkers, loaders, debuggers, and I/O driver software to attempt to make serial processing more efficient.

### Simple Batch Systems

Early machines were very expensive, and therefore it was important to maximize machine utilization. To improve utilization, the concept of batch operating system was developed. It appears that the first batch operating system (and the first operating system of any kind) was developed in the mid-1950s by General Motors for use on an IBM 701. The concept was subsequently refined and implemented on the IBM 704 by a number of IBM customers.

The central idea behind the **batch processing** system was the use of a piece of software known as the **monitor**. With the use of this type of operating system, the user no longer had direct access to the machine. Rather, the user submitted the job on cards or tape to a computer operator, who batches the jobs together sequentially and places the entire batch on an input device, for use by the monitor. Many important breakthroughs in operating system design occurred in the early 1960s which laid the foundation for design of modern operating system.

In 1963, Burroughs design of Master Control Program (MCP), which included design of many of the features of modern systems such as support for multiprocessing (with two identical CPUs) as well as multitasking capabilities.

IBM introduces OS/360 as the operating system for its new System/360 in 1964. OS/360 provided a powerful language to expedite batch processing known as **Job Control Language**. It introduced a simple form of **multiprogramming** or multitasking feature that facilitated loading several jobs into main memory, so that other jobs programs could use the CPU when one job was busy with I/O. By this time, disks were also becoming available. To take its advantage the OS introduced features to cards onto disk while the CPU executed its jobs; thus, when a job completed, the OS could load another job from disk into memory, ready to run. This improved the OS scheduling capability. Multiprogramming or multitasking is the central theme of modern operating system.

### Time Sharing System

With the use multiprogramming, the batch processing can be quite efficient. However, for many jobs, it is desirable to provide a mode in which the user interacts directly with the computer. Indeed for some jobs, such as transaction processing, an interactive mode is essential.

Today, the requirement for an interactive computing facility can be, and often is, met by the use of a dedicated microcomputer. That option was not available in the 1960s when most computers were big and costly. Instead time sharing was developed. Just as multiprogramming allows the processor to handle multiple batch jobs at a time, multiprogramming can be used to handle multiple interactive jobs. In this latter case, the technique is referred to as time sharing, because the processor's time is shared among multiple users. In time sharing system, multiple users simultaneously access the system through terminals, with the operating system interleaving the execution of each user program in a short burst or quantum of computation.

Both batch processing and time sharing use multiprogramming. The key differences are shown in the following table.

**Table 2.1 : Multiprogramming versus Time Sharing**

	<b>Multiprogramming</b>	<b>Time Sharing</b>
<b>Principal objective</b>	Maximize processor use	Minimize response time
<b>Source of directives to operating system</b>	Job control language commands provided with the job	Commands entered at the terminal

MIT, Bell Labs, and GE formed a partnership to develop a major time-sharing system. The system, was called MULTICS (Multiplexed Information and Computer Service), and although MULTICUS never became a full fledged operating system but many of the most important multitasking concepts and algorithms were developed by the MULTICS team.

- During 1960-1970, many other developments in OS design took place:
  - (a) Hierarchical approach to OS, for example the THE OS designed by Dijkstra.
  - (b) It also showed the advantages of modular programming design in the implementation of OS.

## **Development of UNIX**

- The original UNIX development was performed on a Digital PDP-7 minicomputer and later moved to a PDP-11 minicomputer, the forerunner of the VAX computer. Originally, the system was written in assembly language, but at the later stage the operating system was largely rewritten in C.
- UNIX introduced many important OS concepts which have become a trend in OS design today, including the hierarchical file system, the shell concept, redirection, piping, and the use of simple commands that can be combined to perform powerful operations. The authors Ken Thompson and Denis Ritchie of

the OS also developed techniques for interprocess communication, and even provided for networked and distributed processing.

- UNIX earned a reputation for power and flexibility. Because it was written in C, it was also easy to **port** it that is to make it run on other computers. As a result of these factors, UNIX became an important operating system for universities and was ultimately adopted, in many versions.
- Another important innovation, some would say the most important development in making the computer accessible to nontechnical users, was the development of the concept of graphical user interfaces.

### **Development of Personnel Computer Operating System**

- The next important breakthrough in computer use occurred in 1982, with the introduction of the IBM personal computer. The IBM PC was designed as a stand-alone, single-user computer for the mass market. The IBM PC was supplied with a reasonably easy-to-use operating system, PC-DOS, which was developed, and also later marketed, by Microsoft as MS-DOS. PC-DOS was actually derived from an earlier personal computer operating, CP\M (Control Program for Microcomputers).

### **Other Developments during 1980-1990s**

- The evolution of MS-DOS into Windows 2000, a GUI-based operating system capable of supporting a wide range of users and applications, from the individual working at a PC to a networked enterprise system capable of managing every aspect of a large organization. This capability was formerly limited to large, mainframe-based multiuser systems.
- The development of Linux operating system client-server configuration model, which took place at a number of different universities during the 1980s, and the use of these microkernels, particularly Mach, as the basis for the development of new operating systems of unparalleled power, stability, and simplicity.
- The development of the Distributed Computing Environment, plus object-based extensions that make distributed processing practical.
- Creation of the JAVA Virtual Machine and other aspects of the JAVA language and environment as a means to share and execute objects in a way that is truly machine independent and network transparent.
- There have been many other less obvious developments in operating system technology, both subtle and sophisticated, that have impacted computer system design and operation.

This unit has laid down to foundation for understanding the OS concepts which will be discussed in advanced operating system courses. OS accepts requests from users and from users programs and executes their cards in optional manner. The unit introduced several OS concepts such command interface, boot straping, concurrent processing, nanokernels, microkernels, thin clients, multitasking, multiprogramming operating system. To understand an operating system are needs to understand the services provided by the OS. It briefly describes services provided by command processor, file management system, I/O control system, process management and memory management. In the last section of the unit we trace the evolution of OS since 1940s onwords.

---

**2.6 ANSWERS TO CHECK PROGRESS**

---

**Check Your Progress 1**

1. (i) Online simplfy mens that the user is directly connected to the computer and when the system is interchanges, the user interacts directly with the program to provide important data and other works.
- (ii) In batch processing, an user directly interact with the system. Rather than they collect their programs or job(s) in a file or a disk or tape and submit it for later execution. Many business tasks such as banking and insurance operations are performed much effectively in a batch.
2. Multiprogramming improves performance by overlapping CPU and I/O operation on a single machine.

**Check Your Progress 2**

1. The following are the important components of an OS:
  - Command processor
  - File management system
  - Process management
  - Memory management
2. In multitasking OS system, OS determine which job well be admitted into a system and in order. This process is called job scheduling.
3. A thread is an individually executable part of a process. It shares memory and other resources with all other threads in the same process but can be scheduled from other threads.

---

**2.7 FURTHER READINGS**

---

William Stalling, Operating Systems, 4<sup>th</sup> edition, Prentice Hall of India – 2003  
Irv Englander, Architecture of Computer Hardware and System Software, Wiley 2000