

Unit 1 : Introduction to Operating System

An operating system (OS) is an important part of almost every computer system. Most computer users have had some experience with an operating system, but it is difficult to pin down precisely what an operating system. An OS is an integrated set of programs that directs and manages the components and resources of a computer system, including main memory, the CPU and the peripheral devices. The OS¹ is somewhat like a house keeper in that it tidies, organizes and maintains the functioning of various devices. The task of an operating system is to manage the hardware carefully, in order to achieve the best possible performance. This is accomplished by the operating system's controlling and coordinating such resources as the CPU, other processing units, both primary memory and secondary storage, and all input/output devices. The hardware provides raw computing power and the operating system makes this power conveniently accessible to the user. This unit presses what operating system do and basics of OS.

Lesson 1 : Introduction to OS and System Software

1.1. Learning Objectives

On completion of this lesson you will be able to know :

- ◆ the purpose of operating system.
- ◆ an OS and system software
- ◆ the control and service programs that constitute an operating system
- ◆ the operation of the command processor and interrupt handler
- ◆ the operation of the I/O control system
- ◆ system software and its classification.

1.2. Operating System

An operating system is an organized collection of software that controls the overall operations of a computer. In other words, an operating system is a program that acts as an interface between a user of a computer and computer hardware. The primary goal of an OS is to make the computer system convenient to use. A secondary goal is to use the computer hardware in an efficient manner.

1.3. Purposes of an Operating System

The purposes of an operating system are as follows.

¹ OS - means Operating system
S/W - means Software

An OS is a collection of system programs that controls and coordinates the over all operation of a computer.

Introduction to Operating System

Purposes

- ◆ It minimizes the computer user's intervention in and concern about machine's internal workings.
- ◆ It provides an environment in which a user may execute programs.
- ◆ It controls and co-ordinates the use of hardware among the various application programs.
- ◆ It acts as a managers of resources (hardware² and software) and allocates them to specific programs.
- ◆ It controls the various I/O devices and user programs.
- ◆ It maximizes the overall efficiency and effectiveness of the system.
- ◆ It provides an environment within which other programs can do useful work.

Computer System

A computer system consist of hardware (CPU, memory, I/O devices), system software (operating system), application programs (compilers, DBMS, Editor, spreadsheets) and the users

A computer system consist of hardware (CPU, memory, I/O devices), system software (operating system), application programs (compilers, DBMS, Editor, spreadsheets) and the users Fig. 1.1.

The hardware provides the basic computing resources. The applications programs define the ways in which these resources are used to solve the computing problems of the users. The operating system controls and co-ordinates the use of the hardware among the various application programs.

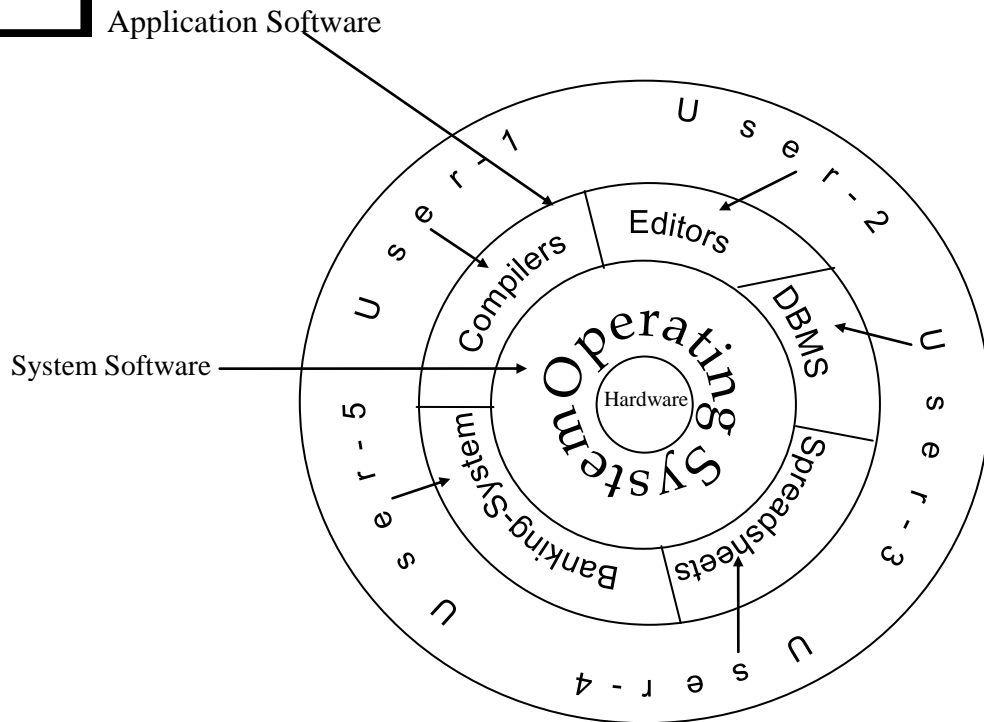


Fig. 1.1 : Elements of a computer system.

² h/w - means hardware

Operating System

1.4. Components of an Operating System

An operating system is made up of control program and service programs. These are as follows:

An operating system is made up of control program and service programs

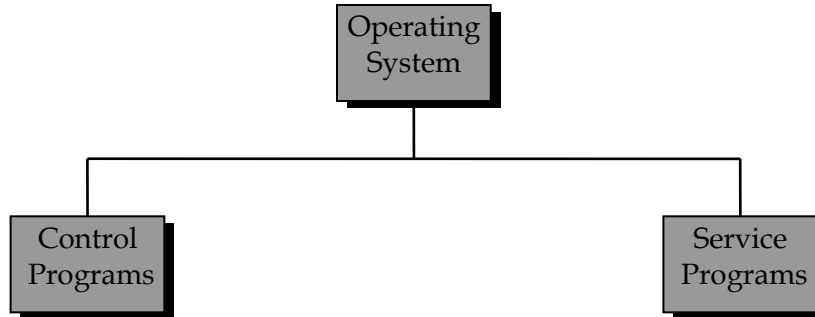


Fig. 1.2 : Components of an operating system.

1.4.1. Control Programs

Control programs permit user-computer communication, log jobs, and oversee the overall computer operation to ensure that the various activities run smoothly and to completion. The principal control programs are shown in Fig. 1.3.

Supervisor Program

The major operating system control program is commonly called the supervisor. The supervisor handles the overall management of a computer system. It is maintained in memory and it supervises the loading of other parts of the operating system from secondary storage into main memory as they are needed. It also supervises the loading of application programs for execution. The supervisor also interprets user messages and it keeps track of jobs.

The major operating system control program is commonly called the supervisor.

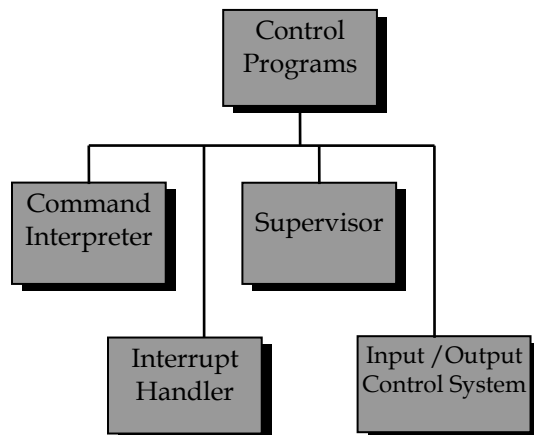


Fig.1.3 : The principal control programs of an operating system.

Command Interpreter

The portion of the operating system that can accept, interpret and carry out user commands is referred to as the command interpreter. The command interpreter consists of a number of individual program modules, each responsible for handling a single command.

The portion of the operating system that can accept, interpret and carry out user commands is referred to as the command interpreter.

Individual user commands to COPY a file, FORMAT a disk and so on, are handled by the command interpreter. Commands to the operating system of a microcomputer are actually requests to execute individual command interpreter programs. For example, the command

COPY SUM B :

directs the command processor to execute the memory-resident copy program, whereas the command

FORMAT B :

requests the externally stored format program to be executed.

Interrupt Handler

The interrupt handler acknowledges and processes all interruptions to the system. One of the most common sources of interrupts is from I/O devices such as the keyboard, printer, and secondary storage. These devices must communicate with the CPU through the operating system. Thus the operating system is never idle, but must constantly be on the alert for an interrupt triggered by internal or external events, such as an I/O device indicating that it has completed its task or that an error condition may have occurred.

The interrupt handler acknowledge and processes all interruptions to the system.

The function of the interrupt handler can vary greatly from a microcomputer system to a mainframe system. Most microcomputer systems handle only one task at a time. Hence, interrupts generally come from the particular device being used at that moment, or from a user-initiated activity, such as a command issued from the keyboard to load a program, execute a program, or abort job.

The handling of interrupts is more complex in a mainframe environment. Unlike microcomputer systems, mainframe systems must perform a number of tasks at the same time if the resources of the system are to be used efficiently. This generally required that the CPU jump back and forth between a number of tasks or application programs being processed in the computer at the same time.

Input/ Output Control System

The input/output control system (IOCS) schedules and activates the proper I/O device as well as the storage unit.

The input/output control system (IOCS) schedules and activates the proper I/O³ device as well as the storage unit. It also monitors the operation of input/output devices. If a needed device is not available, the IOCS will substitute another device in its place. It controls and coordinates the flow of data between I/O devices, for example, from a terminal keyboard to a display screen, or to other output devices like disk drives and printers.

On a microcomputer system IOCS control program is generally maintained in ROM and referred to as the BIOS (basic input/ output system). The IBM PC BIOS is stored on a set of two or four proprietary PROM chips and represents an important area of difference between an IBM PC and PC compatibles or clones. The role of the IOCS in loading a Pascal program, follows.

1. The Pascal compiler disk is inserted and a load command is entered.
2. The command processor interprets the command and executes the load program.
3. The load program assigns the job to IOCS.
4. IOCS directs the following operations :
 - a) Determine the location of the Pascal compiler from the disk directory.
 - b) Direct the read/write heads to the correct track on the disk.
 - c) Read the compiler from disk into memory.
5. The IOCS returns control to the load program (Step 3 reversed).
6. The operating system awaits the user's next command.

1.4.2. Service Programs

In addition top control programs operating system also include service program. There are two types of service programs: *utility* and *library programs*, which perform a variety of labor-saving tasks and functions for the programmer. Whenever a specific task is required, the appropriate service program is accessed and executed by the operating system.

Two types of service programs: utility and library programs, which perform a variety of labor-saving tasks and functions for the programmer.

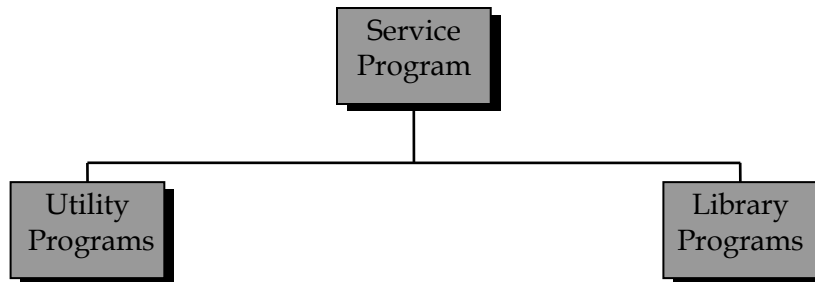


Fig. 1.4 : The principal service programs.

Utility Program

³ I/O - means input/output

Utility programs give the user greater control over the computer system through efficient file management. For example, file can be easily prepared, copied, deleted, merged, sorted, and updated by using the appropriate utility programs.

Library Program

The library program maintains a directory of frequently used software modules and their locations. These programs might consist of manufacturer-supplied or user-written routines or complete programs to compute mathematical functions, control input/output devices, maintain appointments, and so on. The library program makes these routines available when requested by the user, the operating system, or an application program.

The library program maintains a directory of frequently used software modules and their locations.

1.5. System Software

System Software consists of programs designed to facilitate the use of the computer by the user. These programs perform such standard task as organizing and maintaining data files, translating programs written in various languages to a language acceptable to the hardware, scheduling jobs through the computer as well as aiding in other areas of general computer operations.

System Software consists of programs designed to facilitate the use of the computer by the user.

System software refers to programs that are tools to assist the computer user to generate application programs, debug and test them, modify them and, finally execute them. These programs are generally written by the computer manufacturer for one specific computer or system. The same system programs can be used by many different users and many different application programs.

System Software can be classified as follows.

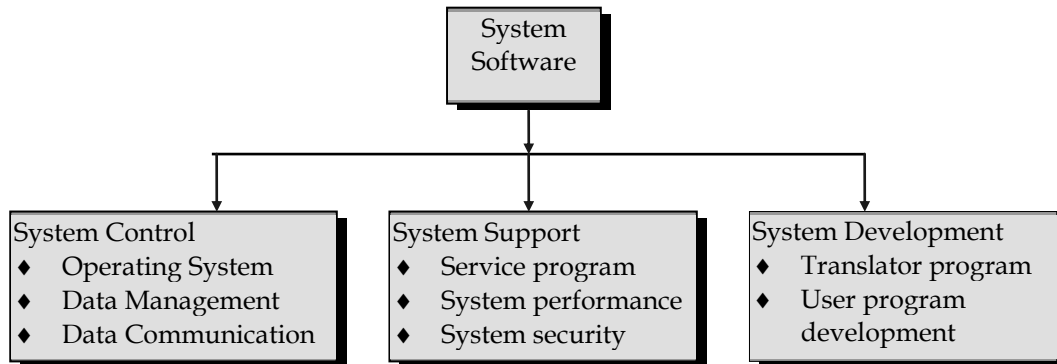


Fig. 1.5 : Classification of system software.

The most fundamental of all the system programs is the operating system, which controls all the computer's resources and provides the base upon which application programs can be written.

Operating System

1.6. Exercise

1.6.1. Multiple choice questions

1. System software consists of
 - i) programs
 - ii) hardware
 - iii) monitor
 - iv) all of the above.

2. The operating system acts as an interface between
 - i) The user of a computer and computer hardware.
 - ii) User and computer software.
 - iii) Hardware and software.
 - iv) None of the above.

3. IOCS stands for
 - i) Input/output communication system.
 - ii) Input/output control system.
 - iii) Interrupt output control system.
 - iv) None of the above.

4. How many types of service programs are there?
 - i) 2
 - ii) 3
 - iii) 4
 - iv) None of the above.

1.6.2. Questions for short answers

- a) What do you understand by operating system?
- b) What are the purposes of an operating system?
- c) What do you understand by computer system? List the components of a computer system.
- d) Define system software. Describe its classification.
- e) What do you understand by OS and system software?
- f) What are the tasks of a supervisor program?
- g) Explain the operation of the command interpreter.
- h) Describe the operations of I/O control system.
- i) What are the fundamental tasks of an OS?

1.6.3. Analytical questions

- a) What do you know about control programs? Describe briefly.
- b) What are the two general categories of an OS? Describe briefly.

Lesson 2 : Serial Batch Processing and Multiprogramming

Operating systems can be classified in a number of ways : by how they organize primary memory, by how many different programs they can execute concurrently, by what kind of secondary storage devices they use for work areas, by the setting in which they are to be used, or by the basic design of their components. So, operating system can be classified according to above outstanding characteristics, but some overlap occurs among the categories. We will divide operating system into six types :

- ◆ serial batch-processing
- ◆ multiprogramming
- ◆ time sharing
- ◆ multiprocessing
- ◆ real time and
- ◆ virtual storage operating system.

Let us start with the simple class, the serial batch processing operating system.

2.1. Learning Objective

On completion of this lesson you will be able to know :

- ◆ serial batch processing system
- ◆ advantages and disadvantages of serial batch processing system
- ◆ multiprogramming system
- ◆ merits and demerits of multiprogramming system.

2.2. Serial Batch Processing Systems

Serial batch processing operating systems can run only a single user program at a time. These are simple systems generally used on mainframes that run in batch mode and on single-user microcomputers. In another words, a system in which a number of similar items or transactions to be processed are grouped (bathed) for sequential processing during a machine run. For example, suppose the operators received one FORTRAN job, one COBOL job and another FORTRAN job. If they ran them in that order, they would have to set up for FORTRAN (load the compiler tapes), then set up for COBOL and finally set up for FORTRAN again. If they ran the two FORTRAN programs as a batch, however, they could set up only once for FORTRAN, saving operator time. The following diagram shows how each program is run before the next begins.

Running one program at a time.

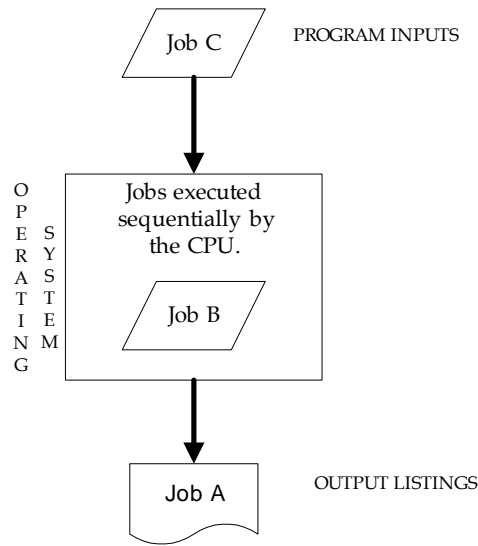


Fig.1.6 : Serial-batch processing operating systems.

advantages

- ◆ It allows a computer to be dedicated to a specific use.
- ◆ It is less complex.
- ◆ There is only one user at a time.
- ◆ There is no possibility that multiple programs will deadlock.

advantages

disadvantages

- ◆ A long turnaround time is needed.
- ◆ Batch systems are slow in both processing and output.
- ◆ In batch system the programs must be debugged statically.

disadvantages

CP/M, DOS, and Macintosh operating system are the example of serial batch-processing systems.

2.4. Multiprogramming Operating System

This system involves simultaneous handling of multiple independent programs by interleaving or overlapping their execution. Multiprogramming is similar to what a chef does in the preparation of a multi-course meal. First, one dish is worked on, then it is set aside while another is attended to, and so on until the entire meal is ready at the same time.

Running several program concurrently.

Multiprogramming is similar to the work of a lawyer. A lawyer does not have only one client at a time. Rather several clients may be in the process of being served at

the same time. While one case is waiting to go to trial or for papers to be typed, the lawyer can work on another case; with enough clients, a lawyer need never be idle.

Multiprogramming operating systems can execute several jobs concurrently by switching the attention of the CPU back and forth among them. This switching is usually prompted by a relatively slow input, output or storage request that can be handled by a buffer, spooler or channel, freeing the CPU to continue processing (Fig. 1.7)

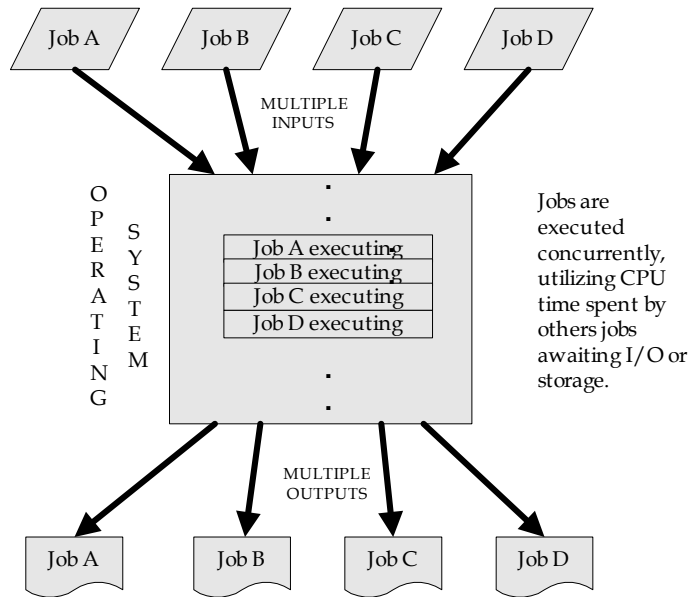


Fig. 1.7 : Multiprogramming operating systems.

The primary reason multiprogramming operating systems were developed, and the reason they are popular, is that they enable the CPU to be utilized more efficiently. If the operating system can quickly switch the CPU to another task whenever the one being worked on requires relatively slow input, output, or storage operations, then the CPU is not allowed to stand idle. This means that more can be accomplished during a given amount of time. For example, if a particular program needs to read data from a disk drive, that task can be delegated to channel and the CPU can be put to work on another program while the data are being read in. Multiprogramming is thus an effective way to keep the fast-working CPU busy with computations while slower input, output, and storage operations are being carried out.

advantages

Advantages

- ◆ it increases CPU utilization
- ◆ it decreases total real time needed to execute a job
- ◆ it maximizes the total job throughput of a computer.

Throughput is the amount of work accomplished in a given time interval.

Operating System

disadvantages

disadvantages

- ◆ it is fairly sophisticated and more complex than serial batch processing operating system.
- ◆ a multiprogramming operating systems must keep track of all the jobs it is concurrently running.

UNIX, Pick and IMB VM can be classified as multiprogramming operating system.

2.5. Exercise

2.5.1. Multiple choice questions

1. Which is appropriate for serial batch processing systems?

- i) Running several programs at a time.
- ii) Running one programs at a time.
- iii) Handling of multiple dependent programs at a time.
- iv) None of the above.

2. Which is the false statement?

- i) A long turnaround time is needed for serial batch system.
- ii) Batch systems are slow in both input and output.
- iii) Batch systems are slow in both processing and output.
- iv) A batch system is less complex

3. Multiprogramming increases

- i) real time
- ii) execution time
- iii) CPU utilization
- iv) minimizes throughput.

2.5.2. Questions for short answers

- a) What do you understand by serial batch processing?
- b) List some advantages of serial batch systems.
- c) What are the disadvantages of serial batch processing systems?
- d) List some of the disadvantages of the multiprogramming systems.

2.5.3. Analytical questions

- a) What do you understand by serial batch processing system? Illustrate with examples.
- b) What are the reason for developing multiprogramming systems?
- c) Describe the multiprogramming operating system.
- d) Why do multiprogramming systems remain popular?

Lesson 3 : Time Sharing and Multiprocessing Operating Systems

3.1. Learning Objectives

On completion of this lesson you will be able to :

- ◆ understand the time sharing systems
- ◆ distinguish between time sharing and multiprogramming systems
- ◆ know the different types of multiprocessing systems.

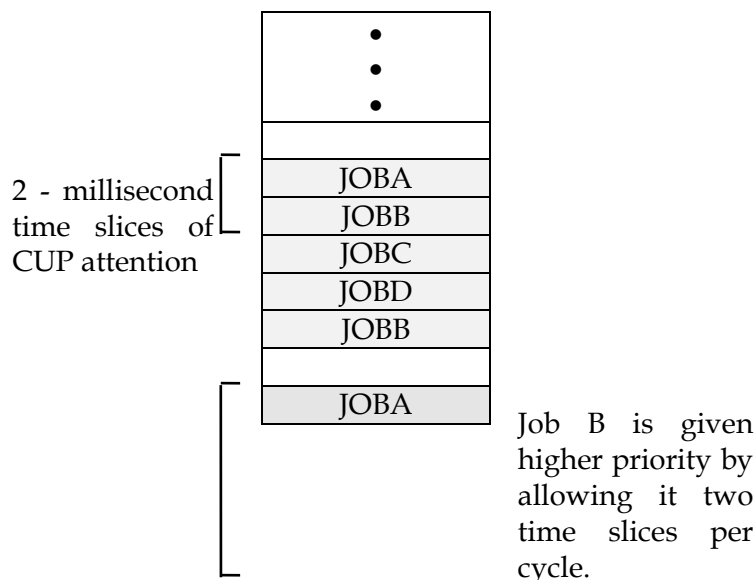
3.2. Time Sharing Systems

Serving several users concurrently.

Time sharing operating systems are time driven multiprogramming systems that serve several users concurrently by rapidly switching among them. With time sharing systems, many users simultaneously share computer resources. Each makes use of a tiny slice of CPU time and then relinquishes the CPU to another user. As the system switches rapidly from one user to the next, users are given the impression that they each have their own computer. Users are unaware that there are other users on the system.

The primary difference between time sharing and multiprogramming operating systems is the criterion that is applied for switching between jobs. Multiprogramming systems are described as being event driven, and time sharing systems are time driven. In other words, a multiprogramming system switches from one program to another on the basis of some event (such as I/O request). A time sharing system, on the other hand, switches to a different job when the clock says so.

Time-sharing operating systems are time-driven multiprogramming systems. Each active job in the system is given one or more fixed time slices of CPU attention per cycle. Jobs or users with higher priorities (such as computer operators or systems programmers) may be allocated more than one time slice per cycle (Fig. 1.8).



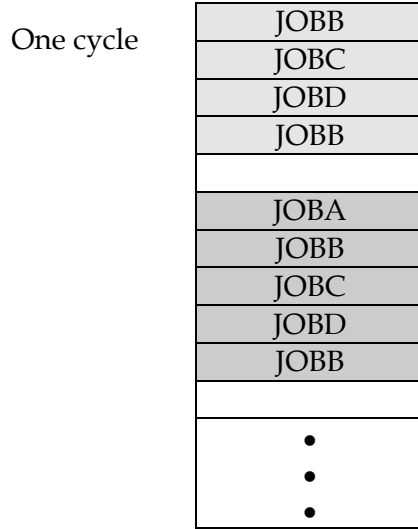


Fig. 1.8 : Time-sharing operating system.

3.3. Multiprocessing Operating Systems

Multiprocessing operating systems can execute several jobs simultaneously through the use of more than one processor. Multiprocessing is the execution of several instructions in parallel fashion on a single computer system having several central processing units (Fig. 1.9). Multiprogramming and time-sharing systems run jobs concurrently, but multiprocessing systems truly run jobs simultaneously.

Running several programs simultaneously with

Multiprocessing systems can be subdivided into four general types, all of which have more than one processor; they are briefly described below.

- ◆ **Homogeneous Multiprocessors** : These systems make use of multiple identical CPUs. The operating systems coordinates the use of storage by the CPUs so that no unresolved conflicts occur. Homogeneous multiprocessors are commonly used in general-purpose mainframe computers used for business applications of data processing.
- ◆ **Nonhomogeneous Multiprocessors** : These systems make use of special-purpose processors in the computing unit, which are actually CPUs in their own right. Nonhomogeneous multiprocessors are found in general-purpose mainframe computers.
- ◆ **Array Processor** : This system is composed of a set of identical processors (each is called a processing element, or PE) that are directed and synchronized by a single control unit. They are designed primarily for rapidly manipulating highly ordered sets of data, such as are encountered in scientific and mathematical applications.

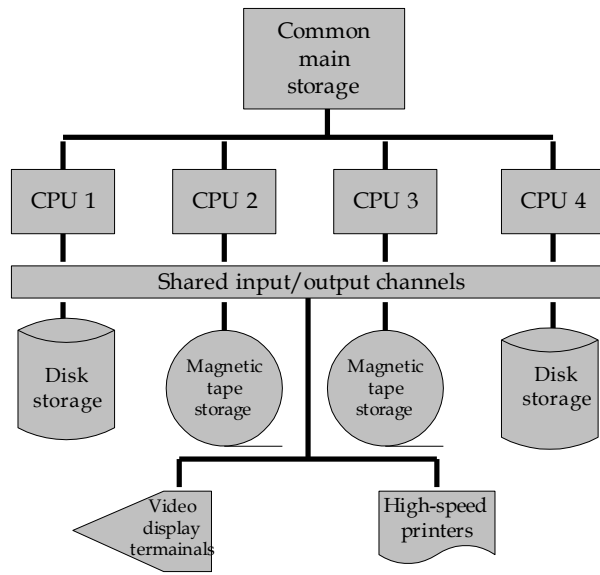


Fig. 1.9: Multiprocessing operating system.

♦ **Pipeline Processors** : In pipeline systems, multiple processors are used to perform different stages of consecutive computer instructions simultaneously. The processors are arranged like a factory production line, allowing several operands to be in different stages of execution at the same time. Like array processor systems, these systems perform calculations very quickly. They are primarily used for scientific and mathematical applications.

1. The main advantage of multiprocessing systems is speed; since more than one CPU is available, jobs can be processed faster than they can with only one CPU.
2. Multiprocessing systems are high-performance operating systems, implemented almost exclusively on mainframes and supercomputers.
3. In this system CPU will more likely be available when it is requested.

Advantages

3.4. Exercises

3.4.1. Multiple choice questions

1. Time sharing systems are
 - i) event driven
 - ii) time driven
 - iii) input driven
 - iv) output driven.

2. Which is true for time sharing operating systems ?
 - i) Running several programs concurrently.
 - ii) Serving several users concurrently.
 - iii) Running several programs simultaneously with one CPU.
 - iv) None of the above.

3. The multiprocessing operating systems can be subdivided into
 - i) 2 general types
 - ii) 3 general types
 - iii) 4 general types
 - iv) 5 general types.

3.4.2. Questions for short answers

- a) What is the goal of time sharing systems ? .
- b) What are the difference between time sharing and multiprogramming systems ?
- c) What is a pipeline processor?
- d) Distinguish between multiprogramming, time sharing and multiprocessing operating system?
- e) List some of the advantages of multiprocessing systems?

3.4.3. Analytical questions

- a) What do you understand by time-sharing Operating Systems? Describe briefly.
- b) What do you know about the multiprocessing operating system?
- c) Describe different types of multiprocessing systems.

Lesson 4 : Real-Time and Virtual Storage Operating System

4.1. Learning Objectives

On completion of this lesson you will be able to :

- ◆ explain real time operating systems
- ◆ describe different types of real-time operating systems
- ◆ describe virtual storage operating systems and their implementation.

4.2. Real-time Operating System

The processing of information or data in a sufficiently rapid manner so that the results of the processing are available in time to influence the process being monitored or controlled.

Controlling or monitoring external processes.

Real time operating systems control computers that interact with their environments to perform work. There are two major types of real-time operating systems : process control systems and process monitor systems.

Process control systems take input data from sensors, analyze them and then cause actions to occur that change the processes that they control.

Process monitor systems also take input data from sensors, but they merely report the data without actually affecting the processes that they are monitoring.

Both of these types of real-time operating systems are being used for more and more industrial and military applications. Real time operating systems are currently being used for such applications as automated environmental monitoring for air and water pollution, directing and monitoring the flows in a chemical plant, police inquiry system, airline reservation, microscopic assembly processes, medical analysis systems, air and automobile traffic control, factory production, oil pipeline regulation and some display systems.

Uses

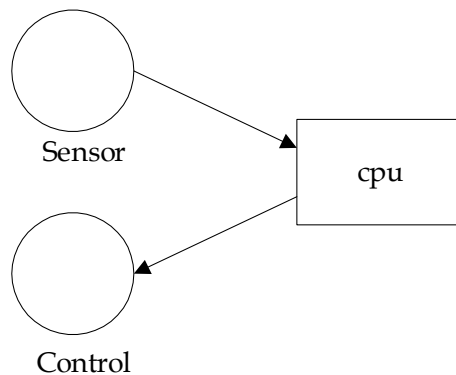


Fig. 1.10 : Real time operating system.

A real time system is often used as a control device in a dedicated application. Sensors bring data to the computer. The computer must analyze the data and possibly adjust controls to modify the sensor inputs. Real time operating systems have to work within strict time limits for critical jobs or the systems will fail. Critical jobs are locked in memory and receive the highest priority.

Real time systems are required to be highly reliable. For example, failure of a system which controls a space vehicle in motion may result in a fatal accident. In such cases, duplicate systems are used so that if one system fails, the other will take over.

4.3. Virtual Storage Operating System

Virtual storage is a technique that uses some secondary storage, by employing segmentation and/or paging, to augment primary memory. Virtual storage is a memory management tactic that employs an area of rapidly accessible secondary storage (e.g. a hard disk) as an extension of primary memory. Portions of programs are swapped into real storage (the actual primary memory) from virtual storage as needed. This gives users the illusion that more primary memory is available than is actually the case. Since this memory management is automatically taken care of by the operating system, users are freed from having to worry about how much memory their programs will require.

Virtual storage is a memory management tactic of using some secondary storage to augment primary memory. Users don't have to worry if their programs require more space in memory than is actually available because such systems can give them as much virtual storage as is needed. Virtual storage is usually implemented by segmentation, paging or a combination of them.

4.3.1. Segmentation

Segmentation is the process of dividing up a program that is to be run into a number of chunks (or segments) of different sizes and placing these segments in memory wherever they fit. Segmentation divides programs into pieces of different sizes, which are stored in secondary storage and transferred into primary memory (Fig. 1.11).

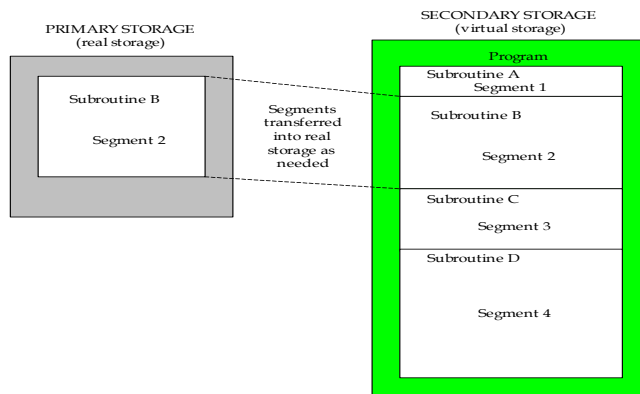


Fig. 1.11 : Segmentation.

Memory management to overcome space limitations.

4.3.2. Paging

Paging is similar to segmentation except that programs are divided into equal sized portions. As with segmentation, the operating systems keeps track of page locations by constructing page table. As pages are fixed sizes, the use of paging can result in less waste of real storage space. Since segments can be of different sizes, swapping in new segments from virtual storage can leave little fragments of unused space in real memory. By making all program pieces the same size, paging eliminates this type of waste.

The best memory management scheme is to combine segmentation and paging by first segmenting programs and then further subdividing each segment into pages.

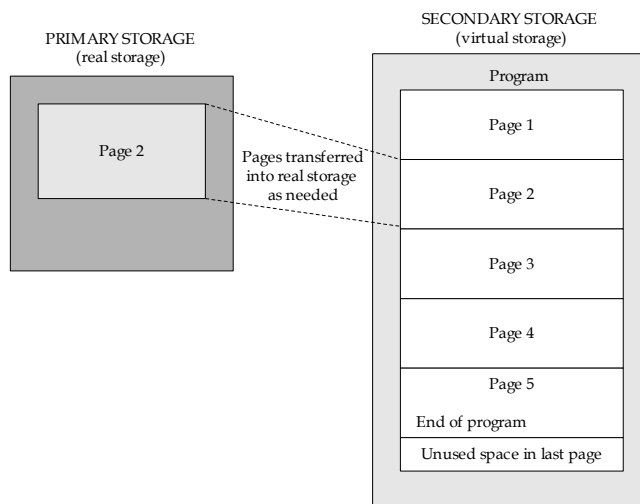


Fig. 1.12 : Paging.

4.4. Exercises

4.4.1. Multiple Choice Questions

1. How many types of real-time operating system are there in this lesson?

- i) 2
- ii) 3
- iii) 4
- iv) 5.

2. Which of the following is related to real-time operating systems?

- i) Execution of programs concurrently.
- ii) Controlling or monitoring external processes.
- iii) Serving several users at a time.
- iv) None of the above.

4.4.2. Questions for short answers

- a) List some of the uses of real time operating systems.
- b) Describe the real time operating system.
- c) What do you understand by segmentation?
- d) What is paging?
- e) What are the two types of operating systems?

4.4.3. Analytical questions

- a) What do you understand by real-time operating systems? Describe briefly.
- b) What is process control system and process monitor system?
- c) Describe a virtual storage operating system with their implementation.

Lesson 5 : Functions and Evaluation of Operating System

5.1. Learning Objectives

On completion of this lesson you will know :

- ◆ major functions of an operating system
- ◆ the evaluation of operating system.

5.2. Functions of Operating Systems

An operating system performs support functions. Before the advent of operating system in the early 1960s, computer operators had to perform the support functions that are now done by the operating system. Today, operating systems successfully perform many of the functions previously assigned to operators; further more, modern operating systems perform these functions better, faster and more economically. An operating system provides an environment for the execution of programs. An Operating System has a complex mixture of diversified functions. The specific functions provided will, of course, differ from one operating system to another, but there are some common classes of functions which can be identified. Its major functions are as follows :

- ◆ *Memory Management* : Memory management involves monitoring the various storage and retrieval operations in main memory. The operating system keeps track of vital information such as which areas are and are not in use and who is using the memory at any given time. In a large system this task can be very complicated. If virtual memory is being used, for example, portions of main memory are being transferred to and from the disk, and the operating system coordinates this process. Paging and segmentation are two common methods of realizing virtual memory. Another example of complex memory management is multiprogramming, a mode of operation in which two or more computer programs are executed by a single CPU in an interleaved manner. The operating system must then coordinate the memory requirements of the several programs.
- ◆ *CPU Management* : On smaller computers, the operating system keeps track of the status of the CPU at any point. It determines, for instance, whether the control unit is in the instruction cycle or the execution cycle. When many users are competing for the CPU, the operating system must prioritize and schedule its use. Larger minicomputers and mainframes often include several CPUs, which are together referred to as a multiprocessor. In multiprocessing (operating a multiprocessor system), CPU management becomes much more complex. The major advantage of multiprocessing, from the user's standpoint, is that a CPU will more likely be available when it is requested.

Memory management

CPU management

Introduction to Operating System

I/O and file management

- ◆ *Input/output and File Management* : The operating system efficiently and reliably controls many different types of input and output devices, such as keyboards, printers, monitors, and audio inputs. These devices have different specifications in terms of speed, printing density, control mechanism, and other variables. The operating system is also responsible for keeping track of the files and directories that reside on hard or floppy disks.

Management of communication

- ◆ *Management of Communication* : The operating system manages communication among computers connected on a network. Since data exchanged on the network may be received intermittently instead of continuously, they must be combined by the operating system. The received data is then converted into a form that computers can process. The operating system also keeps track of the status of the network, disconnecting faulty portions, reporting computer usage accounts, and so on.

Security

- ◆ *Security* : The operating system protects computers from access by illegal users and from data corruption introduced by unintentional mistakes made by legitimate users. Security is particularly important for computers that are connected to a communications network, because many users can freely access any computer. Authorized users are authenticated by entering an individual password, and then a computer usage fee is charged to the account of each user.

User interface

- ◆ *User Interface* : The operating system provides a convenient interface between a computer and its users. In the case of batch processing with mainframes, for example, users may want to run their large program only after midnight for several days until the execution of the programs is completed. In the case of personal computers and workstations, graphical user interfaces such as windows and icons displayed on a monitor are convenient, for users.

Detection of errors

- ◆ *Detection of Errors* : The operating system constantly needs to be aware of possible errors that may occur in the CPU and memory hardware (e.g. a memory error or power failure), in I/O devices (e.g. a parity error on tape or the printer out of paper), or in the user program (e.g. an arithmetic overflow, an attempt to access illegal memory location or using too much CPU time). For each type of error, the operating system should take the appropriate action to ensure correct and consistent computing.

Information management

- ◆ *Information Management* : The operating system also monitors system information, which is organized into records and files. There are several distinct tasks of information management. These are as follows :
 - ◆ Managing groups of file
 - ◆ Managing file directories
 - ◆ Processing and managing the records within a file.

Operating System

Allocation of resources

Allocation of Resources : If, there are multiple users or multiple jobs running at the same time, then an operating system must coordinate the use of all available resources. A good operating system accomplishes this in the most efficient manner possible.

Besides these functions, some of the functions provided by a modern operating system is as follows :

- ◆ Provides for human - computer interaction
- ◆ Boots or starts the computer operations
- ◆ Schedules job
- ◆ Manages data and file storage
- ◆ Assigns different task to the CPU
- ◆ Provides security and control.

Functions of modern OS

5.3. Development of Operating System

OS have been evolving through the years. In the following sections we will briefly look at this development.

The earliest computer systems had no operating systems; users had access to computer resources only via machine language programs. Programs were run one at a time by computer operators who manually entered the commands to initiate and complete each one. This pattern of usage wasted a great deal of computer time, since the CPU remained idle between the completion of one task and the initiation of the next.

History of OS

The 1950s were marked by the development of rudimentary operating systems designed to smooth the transitions between jobs (a job is any program or part of a program that is to be processed as a unit by a computer). This was the start of batch processing, in which programs to be executed were grouped into batches. While a particular program was running, it had total control of the computer. When it finished, control was returned to the operating system, which handed any necessary finalizations and read in and started up the next job. By letting the computer handle the transition between one job and the next instead of having it done manually, less time was taken up and the CPU was more efficiently utilized.

During the 1960s, operating systems became much more sophisticated, leading up to the development of shared system. These multiprogramming, time-sharing and multiprocessing systems (which we have defined and discussed in more detail in the previous lessons) allowed several user programs to be run on a single computer system, seemingly at the same time. Additionally, these systems were the first to allow usage to take place in interactive, or conversational, mode, in which the user communicates directly with the computer, rather than submitting jobs and passively waiting for their completion. These developments made computer systems more widely accessible and easier to use.

Real-time systems also emerged during the 1960s. These operating systems enabled computers to be used to control systems characterized by the need for immediate response, such as weapons systems or industrial plants. For example, if an oil refinery is being controlled by a real-time system, that system must respond immediately to temperature conditions that could cause an explosion.

In the late 1960s and the early 1970s, there was a trend toward general-purpose operating systems. These tried to be all things to all users. Often called multi-mode systems, some of them simultaneously supported batch processing, time sharing, real-time processing and multiprocessing. They were large, expensive, and difficult to develop and maintain, but they helped sell a lot of computers. The prime example of this type of operating system was the one offered with the IBM 360 family of computers first introduced in 1964. To get one of these monsters to perform even the simplest task, users had to learn a complex job control language (JCL) and employ it to specify how their programs were to be run and what resources they would need.

The operating systems from the mid-1970s to the present cannot be characterized by a single, all-encompassing feature. The development of microcomputers and of simple, easy-to-use, single-user operating system has had a profound effect on the newest systems being developed for all types of computers. The features most in demand are a high degree of user-friendliness and a computing environment that is menu-driven (refers to the user of displays and prompts that aid users in selecting functions). Also, operating systems that support on-line processing, computer networking, data security, and distributed data processing are the latest word. Modern operating systems create a virtual machine, an interface that relieves the user of any need to be concerned about most of the physical details of the computer system or network being accessed. The virtual machine presented by the operating system lets users concentrate on getting done what they need, without having to be familiar with the internal functioning of the actual hardware.

History of operating system.

5.4. Different Classes of Computers

Most operating system researches and development has been done for mainframe computers. Mainframe operating systems have been developing over the last thirty years.

In the mid-1960's, minicomputers appeared which are on smaller and less expensive than mainframe system. In the 1970's personal computers appeared which are even smaller and less expensive operating systems.

In general, an examination of operating systems for mainframes, minicomputers, and personal computers(PC) shows that features which were at one time available only on mainframes have been adopted by minicomputers. Those on minicomputers have been introduced on PC. The same concepts and techniques are appropriate for all the various different classes of computers.

Operating System

A good example (Fig.1.13) of this migration can be seen by considering the evolution of the Unix from Multics operating system. Multics was developed from 1965 to 1970 at MIT as a computing utility. It ran on a very large and complex mainframe computer. Many of the ideas which were developed for Multics were subsequently used at Bell Labs in the design of Unix, which has become one of the most popular minicomputer systems around 1970 was offered on many PCs around 1980. Thus the features developed for a large mainframe system can be seen to have moved to PC over time.

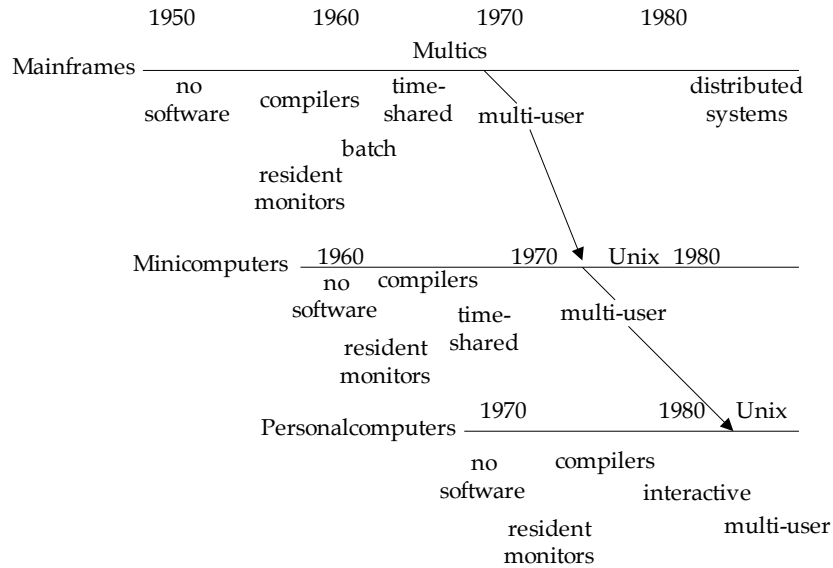


Fig. 1.13 : Migration of operating system concepts and features.

Introduction to Operating System

5.5. Exercises

5.5.1. Multiple choice questions

1. Which of the following is not the function of an operating system?
 - i) Memory management
 - ii) CPU management
 - iii) I/O and file management
 - iv) Debugging programs.

2. How many tasks of information management are there in this lesson?
 - i) 2
 - ii) 3
 - iii) 4
 - iv) None of the above.

5.5.2. Questions for short answers

- a) List some of the functions of a modern operating system.
- b) What is major advantage of multiprocessing?
- c) What are the tasks of information management?
- d) What do you know about micro-kernel?

5.5.3. Analytical questions

- a) What are the services provided by an OS? Briefly describe.
- b) Write an essay on the history of operating systems.