Unit 6 Operating System Concepts

Structure:

6.1 Introduction
Objectives
6.2 Operating System Concepts
6.3 Functions of Operating System
6.4 Development of Operating System
Early Systems
Simple Batch Systems
Multi Programmed Batch Systems
Time Sharing Systems
Distributed Systems
Real-time Operating System
6.5 Operating System Components
6.6 Operating System Services
6.7 Operating Systems for Different Computers
6.8 Summary
6.9 Terminal Questions
6.10 Answers

6.1 Introduction

As we discussed in the previous unit, Computer software has become a driving force. It is the engine that drives business decision making. It serves as the basis for modern scientific investigation and engineering problem solving. In this unit, we will study the operating system concepts like, functions of an Operating System, Development of Operating System, Early Systems, Simple Batch Systems, Multi Programmed Batch Systems, Time Sharing Systems, Distributed Systems, Real-time Operating System, Operating System Components, Operating System Services and Operating Systems for Different Computers.

Objectives:

After studying this unit, you should be able to:

- describe Operating System
- deliberate the various functions of an Operating System
- explain the evolution of Operating Systems

- define the Operating System Components
- explain the various Operating System Services

6.2 Operating System Concepts

An operating system (OS) is a platform that controls the implementation of an application program and acts as an interface between the user and computer hardware. The purpose of an OS is to provide an environment in which a user can execute programs in a convenient and efficient manner. The operating system must provide definite services to programs and to the users of those programs in order to make the programming task easier, these services will differ from one OS to another. It shields the user of the machine from the low-level details of the machine's operation and provides frequently needed facilities. There is no universal definition of what an operating system consists of. You can think of it as being the software which is already installed on a machine, before you add anything of your own.

Generally the operating system has a number of key features: (i) Technical layer of software for driving the hardware of the computer, like disk drives, the keyboard and the screen; (ii) File system which provides a way of organizing files logically, and (iii) Simple command language which enables users to run their own programs and to manipulate their files in a simple way. Some operating systems also provide text editors, compilers, debuggers and a variety of other tools. Since the operating system (OS) is in charge of a computer, all requests to use its resources and devices need to go through the OS. An OS therefore provides (IV) legal entry points into its code for performing basic operations like writing to devices.

Procedure of instructions telling the computer what to do is called a program. The user normally uses a text editor to write their program in a high level language, such as Pascal, C, Java, etc. Alternatively, they may write it in assembly language. Assembly language is a computer language whose statements have an almost one to one correspondence to the instructions understood by the CPU of the computer. It provides a way of specifying in precise detail what machine code the assembler should create.

A compiler is used to translate a high level language program into assembly language or machine code, and an assembler is used to translate an assembly language program into machine code. A linker is used to combine

reloadable object files (code files corresponding to incomplete portions of a program) into executable code files (complete code files, for which the addresses have been resolved for all global functions and variables).

A thread is an example of execution (the entity that executes). All the threads that make up a process share access to the same user program, virtual memory, open files, and other operating system resources. Each thread has its own program counter, general purpose registers, and user and kernel stack. The program counter and general purpose registers for a thread are stored in the CPU when the thread is executing, and saved away in memory when it is not executing.

Multi-threaded processes are becoming very important, because computers with multiple processors are becoming commonplace, as are distributed systems, and servers. It is important that you learn how to program in this manner. Multi-threaded programming, particularly dealing with synchronisation issues, is not trivial, and a good conceptual understanding of synchronisation is essential.

6.3 Functions of Operating System

An operating system is a software component that acts as the core of a computer system. It performs various functions and is essentially the interface that connects your computer and its supported components. In this section, we will discuss the basic functions of the operating system. Operating systems generally accomplish these goals by running processes in low privilege and providing service calls that invoke the operating system kernel in high-privilege state.

The main functions of an operating System are:

Resource Management:

The resource management function of an operating system allocates computer resources such as CPU time, main memory, secondary storage, and input and output devices for use. One can view Operating Systems from two points of views: Resource manager and extended machines. Form Resource manager point of view Operating Systems manage the different parts of the system efficiently and from extended machines point of view Operating Systems provide a virtual machine to users that is more convenient to use. The structurally Operating Systems can be design as a

monolithic system, a hierarchy of layers, a virtual machine system, a micro-kernel, or using the client-server model. The basic concepts of Operating Systems are processes, memory management, I/O management, the file systems, and security.

Data Management:
Data management keeps track of the data on disk and other storage devices. The application program deals with data by file name and a particular location within the file. The operating system's file system knows where the data are physically stored and interaction between the application and operating system is through the programming interface (API). Whenever an application needs to read or write data, it makes a call to the operating system.

Job management:
Job management controls the order and time in which applications are run and is more sophisticated in the mainframe environment where scheduling the daily work has always been routine. IBM's job control language (JCL) was developed decades ago for that purpose. In a desktop environment, batch files can be written to perform a sequence of operations that can be scheduled to start at a given time.

Task Management:
Task management, multitasking, which is the ability to simultaneously execute multiple programs, is available in all operating systems today. Critical in the mainframe and server environment, applications can be prioritized to run faster or slower depending on their purpose. In the desktop world, multitasking is necessary for keeping several applications open at the same time so users can bounce back and forth among them.

Device Management:
Device management controls peripheral devices by sending them commands in their proprietary command language. The software routine that deals with each device is called a "driver," and the operating system requires drivers for the peripherals attached to the computer. When a new peripheral is added, that device's driver is installed into the operating system.

User interface – The user interacts with the operating systems through the user interface and usually interested in the look and feel of the operating

system. The most important components of the user interface are the command interpreter, the file system, on-line help, and application integration. The recent trend has been toward increasingly integrated graphical user interfaces that encompass the activities of multiple processes on networks of computers.

Daily uses of an Operating System

– Performing application programs
– Formatting floppy diskettes
– Setting up directories to organize the files
– Displaying a list of files stored on a particular disk
– Verifying that there is enough room on a disk to save a file.
– Protecting and backing up your files by copying them to other disks for safekeeping.

Self Assessment Questions

1. _____ is software that hides lower level details and provides a set of higher-level functions.
2. _____ transforms the computer hardware into multiple virtual computers, each belonging to a different program.

6.4 Development of Operating System

Operating system and computer architecture have had a great deal of impact on each other. To simplify the use of the hardware, OS's were developed. As operating systems were designed and used, it became obvious that changes in the design of the hardware could simplify them.

6.4.1 Early Systems

In the initial days of electronic digital computing, the whole lot was done on the bare hardware. Very few computers existed and those that did exist were experimental in nature. The researchers who were making the first computers were also the programmers and the users. They worked directly on the “bare hardware”. There was no operating system. The experimenters wrote their programs in assembly language and a running program had complete control of the entire computer. Debugging consisted of a combination of fixing both the software and hardware, rewriting the object code and changing the actual computer itself. The nonexistence of any operating system meant that only one person could use a computer at a

time. Even in the research lab, there were many researchers competing for limited computing time. The first solution was a reservation system, with researchers signing up for specific time slots.

The answer to this problematic was to have programmers make their work off-line on some input medium (often on punched cards, paper tape, or magnetic tape) and then hand the work to a computer operator. The computer operator would load up jobs in the order received (with priority overrides based on politics and other factors). Each job still ran one at a time with complete control of the computer, but as soon as a job finished, the operator would transfer the results to some output medium (punched tape, paper tape, magnetic tape, or printed paper) and deliver the results to the suitable programmer. If the program ran to completion, the result would be some end data. If the program crashed, memory would be transferred to some output medium for the programmer to study (because some of the early business computing systems used magnetic core memory, these became known as “core dumps”). After the first successes with digital computer experiments, computers moved out of the lab and into practical use. The first practical application of these experimental digital computers was the generation of artillery tables for the British and American armies. Much of the early research in computers was paid for by the British and American militaries. Business and scientific applications followed. As computer use increased, programmers noticed that they were duplicating the same efforts.

Every single programmer was writing his or her own routines for I/O, such as reading input from a magnetic tape or writing output to a line printer. It made sense to write a common device driver for each input or output device and then have every programmer share the same device drivers rather than each programmer writing his or her own. Some programmers resisted the use of common device drivers in the belief that they could write “more competent” or faster or "“better” device drivers of their own.

6.4.2 Simple Batch Systems

While punched cards were used for user jobs, processing of a job involved physical actions by the system operator, e.g., loading a deck of cards into the card reader, pressing switches on the computer's console to initiate a job, etc. These actions wasted a lot of central processing unit (CPU) time.

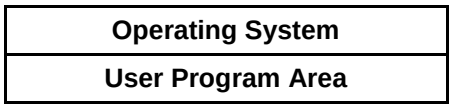

Fig. 6.1: Simple Batch System

To speed up processing, jobs with similar needs were *batched* together and were run as a group. Batch processing (BP) was implemented by locating a component of the BP system, called the batch monitor or supervisor, permanently in one part of computer's memory. The remaining memory was used to process a user job - the current job in the batch as shown in the figure 6.1 above.

The interval between job submission and accomplishment was considerable in batch processed system as a number of programs were put in a batch and the entire batch had to be processed before the results were printed. Further card reading and printing were slow as they used slower mechanical units compared to CPU which was electronic. The speed mismatch was of the order of 1000. To alleviate this problem programs were spooled. Spool is an acronym for simultaneous peripheral operation on-line. In essence the idea was to use a cheaper processor known as peripheral processing unit (PPU) to read programs and data from cards store them on a disk. The faster CPU read programs/data from the disk processed them and wrote the results back on the disk. The cheaper processor then read the results from the disk and printed them.

6.4.3 Multi Programmed Batch Systems

Even though disks are faster than card reader/ printer they are still two orders of magnitude slower than CPU. It is thus useful to have several programs ready to run waiting in the main memory of CPU. When one program needs input/output (I/O) from disk it is suspended and another program whose data is already in main memory (as shown in the figure 6.2 bellow) is taken up for execution. This is called multiprogramming.

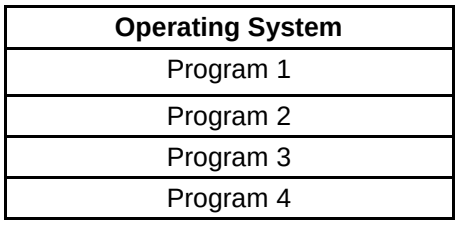

Fig. 6.2: Multi Programmed Batch Systems

Multiprogramming (MP) increases CPU utilization by organizing jobs such that the CPU always has a job to execute. Multiprogramming is the first instance where the operating system must make decisions for the user.

The MP arrangement ensures concurrent operation of the CPU and the I/O subsystem. It ensures that the CPU is allocated to a program only when it is not performing an I/O operation.

6.4.4 Time Sharing Systems

Multiprogramming features were overlaid on batch processing to ensure good utilization of CPU but from the point of view of a user the service was poor as the *response time,* i.e., the time elapsed between submitting a job and getting the results was unacceptably high. Development of interactive terminals changed the scenario. Computation became an *on-line* activity. A user could provide inputs to a computation from a terminal and could also examine the output of the computation on the same terminal. Hence, the response time needed to be drastically reduced. This was completed by storing programs of several users in memory and providing each user a slice of time on CPU to process his/her program.

6.4.5 Distributed Systems

Distributed operating system is an operating system which manages a number of computers and hardware devices which make up a Distributed System. Such an operating system has a number of functions: it manages the communication between entities on the system, it imposes a security policy on the users of the system, it manages a Distributed File System, it monitors problems with hardware and software, it manages the connections between application programs and itself, and it allocates resources such as file storage to the individual users of the system. A good distributed operating system should give the user the impression that they are interacting with a single computer.

A recent trend in computer system is to distribute computation among several processors. In the loosely *coupled systems* the processors do not share memory or a clock. Instead, each processor has its own local memory. The processors communicate with one another using communication network.

The processors in a distributed system may vary in size and function, and referred by a number of different names, such as sites, nodes, computers and so on depending on the context. The major reasons for building distributed systems are:

Resource sharing: If a number of different sites are connected to one another, then a user at one site may be able to use the resources available at the other.

Computation speed up: If a particular computation can be partitioned into a number of sub computations that can run concurrently, then a distributed system may allow a user to distribute computation among the various sites to run them concurrently.

Reliability: If one site fails in a distributed system, the remaining sites can potentially continue operations.

Communication: There are many instances in which programs need to exchange data with one another. Distributed data base system is an example of this.

6.4.6 Real-time Operating System

The advent of timesharing provided good response times to computer users. However, timesharing could not satisfy the requirements of some applications. Real-time (RT) operating systems were developed to meet the response requirements of such applications.

There are two types of real-time systems. A hard real-time system guarantees that critical tasks complete at a specified time. A less restrictive type of real time system is soft real-time system, where a critical real-time task gets priority over other tasks, and retains that priority until it completes. The several areas in which this type is useful are multimedia, virtual reality, and advance scientific projects such as undersea exploration and planetary rovers. Because of the expanded uses for soft real-time functionality, it is finding its way into most current operating systems, including major versions of UNIX and Windows NT O.S.

A real-time operating system is one, which helps to fulfill the worst-case response time requirements of an application. An RT OS provides the following facilities for this purpose:

1. Multitasking within an application.
2. Ability to define the priorities of tasks.

3. Priority driven or deadline oriented scheduling.
4. Programmer defined interrupts.

Self Assessment Questions

3. Multiprogramming (MP) increases CPU utilization by organizing jobs such that the CPU always has a job to execute. (True/False)
4. Real-time (RT) operating systems were developed to meet the response requirements of such applications. (True/False)

6.5 Operating System Components

Even though, not all systems have the same structure many modern operating systems share the same goal of supporting the following types of system components.

1. Process Management

The operating system manages many kinds of activities ranging from user programs to system programs like printer spooler, name servers, file server etc. Each of these activities is encapsulated in a process. A process includes the complete execution context (code, data, PC, registers, OS resources in use etc.).

It is important to note that a process is not a program. A process is only ONE instant of a program in execution. There are many processes can be running the same program. The five major activities of an operating system in regard to process management are

1. Creation and deletion of user and system processes
2. Suspension and resumption of processes
3. A mechanism for process synchronization
4. A mechanism for process communication
5. A mechanism for deadlock handling

2. Main-Memory Management

Primary-Memory or Main-Memory is a large array of words or bytes. Each word or byte has its own address. Main-memory provides storage that can be access directly by the CPU. That is to say for a program to be executed, it must in the main memory.

The major activities of an operating in regard to memory-management are:

1. Keep track of which part of memory is currently being used and by whom.
2. Decide which processes are loaded into memory when memory space becomes available.
3. Allocate and de-allocate memory space as needed.

3. File Management

A file is a collection of related information defined by its creator. Computer can store files on the disk (secondary storage), which provides long term storage. Some examples of storage media are magnetic tape, magnetic disk and optical disk. Each of these media has its own properties like speed, capacity, and data transfer rate and access methods. A file system normally organized into directories to ease their use. These directories may contain files and other directions.

The five main major activities of an operating system in regard to file management are

1. The creation and deletion of files.
2. The creation and deletion of directions.
3. The support of primitives for manipulating files and directions.
4. The mapping of files onto secondary storage.
5. The backup of files on stable storage media.

4. I/O System Management

I/O subsystem hides the peculiarities of specific hardware devices from the user. Only the device driver knows the peculiarities of the specific device to which it is assigned.

5. Secondary-Storage Management

Normally systems have numerous levels of storage, including primary storage, secondary storage and cache storage. Instructions and data must be placed in primary storage or cache to be referenced by a running program. Because main memory is too small to accommodate all data and programs, and its data are lost when power is lost, the computer system must provide secondary storage to back up main memory. Secondary storage consists of tapes, disks, and other media designed to hold information that will eventually be accessed in primary storage (primary, secondary, cache) is ordinarily divided into bytes or words consisting of a

fixed number of bytes. Each location in storage has an address; the set of all addresses available to a program is called an address space.
The three major activities of an operating system in regard to secondary storage management are:

1. Scheduling the requests for memory access.
2. Managing the free space available on the secondary-storage device.
3. Allocation of storage space when new files have to be written.

6. Networking

A distributed system is a group of processors that do not share memory, peripheral devices, or a clock. The processors communicate with one another through communication lines called network. The communication-network design must consider routing and connection strategies, and the problems of contention and security.

7. Protection System

If a computer system has multiple users and allows the concurrent execution of multiple processes, then various processes must be protected from one another's activities. Protection refers to mechanism for controlling the access of programs, processes, or users to the resources defined by a computer system.

8. Command Interpreter System

A command interpreter is an interface of the operating system with the user. The user gives commands with are executed by operating system (usually by turning them into system calls). The main function of a command interpreter is to get and execute the next user specified command. Command-Interpreter is usually not part of the kernel, since multiple command interpreters (shell, in UNIX terminology) may be supported by an operating system, and they do not really need to run in kernel mode. There are two main advantages of separating the command interpreter from the kernel.

1. If we want to change the way the command interpreter looks, i.e., I want to change the interface of command interpreter, I am able to do that if the command interpreter is separate from the kernel. I cannot change the code of the kernel so I cannot modify the interface.
2. If the command interpreter is a part of the kernel, it is possible for a malicious process to gain access to certain part of the kernel that it

should not have. To avoid this scenario it is advantageous to have the command interpreter separate from kernel.

Self Assessment Questions

5. _____ is a collection of related information defined by its creator.
6. _____ System is a collection of processors that do not share memory, peripheral devices, or a clock.
7. The main function of a command interpreter is to get and execute the next user specified command. (True/False)

6.6 Operating System Services

Operating systems are responsible for providing essential services within a computer system:

- Loading of programs and transfer of programs between secondary storage and main memory
- Supervision of the input/output devices
- File management
- Protection facilities

Following are the five services provided by operating systems for the convenience of the users.

1. Execution of Program

The persistence of a computer system is to allow the user to execute programs. So the operating system provides an environment where the user can conveniently run programs. The user does not have to worry about the memory allocation or multitasking or anything. These things are taken care of by the operating systems. Running a program involves the allocating and de-allocating memory, CPU scheduling in case of multi-process. These functions cannot be given to the user-level programs. So user-level programs cannot help the user to run programs independently without the help from operating systems.

2. I/O Operations

Every program needs an input and produces output. This involves the use of I/O. The operating systems hide from the user the details of underlying hardware for the I/O. All the users see that the I/O has been performed without any details. So the operating system, by providing I/O, makes it convenient for the users to run programs. For efficiently and protection

users cannot control I/O so this service cannot be provided by user-level programs.

3. File System Manipulation

The output of a program may need to be written into new files or input taken from some files. The operating system provides this service. The user does not have to worry about secondary storage management. User gives a command for reading or writing to a file and sees his/her task accomplished. Thus operating system makes it easier for user programs to achieve their task.

This facility involves secondary storage management. The speed of I/O that depends on secondary storage management is critical to the speed of many programs and hence I think it is best relegated to the operating systems to manage it than giving individual users the control of it. It is not difficult for the user-level programs to provide these services but for above mentioned reasons it is best if this service is left with operating system.

4. Communications Process

There are instances where processes need to communicate with each other to interchange information. It may be between processes running on the same computer or running on the different computers. By providing this service the operating system relieves the user from the worry of passing messages between processes. In case where the messages need to be passed to processes on the other computers through a network, it can be done by the user programs. The user program may be customized to the specifications of the hardware through which the message transits and provides the service interface to the operating system.

5. Error Finding

An error in one part of the system may cause malfunctioning of the complete system. To avoid such a situation the operating system constantly monitors the system for detecting the errors. This relieves the user from the worry of errors propagating to various part of the system and causing malfunctioning. This service cannot be allowed to be handled by user programs because it involves monitoring and in cases altering area of memory or de-allocation of memory for a faulty process, or may be relinquishing the CPU of a process that goes into an infinite loop. These tasks are too critical to be handed over

to the user programs. A user program if given these privileges can interfere with the correct (normal) operation of the operating systems.

Self Assessment Questions

8. An error in one part of the system may cause malfunctioning of the complete system. (True/False)
9. _____involves the allocating and de-allocating memory, CPU scheduling in case of multi-process.

6.7 Operating Systems for Different Computers

Operating systems can be gathered according to functionality: operating systems for Supercomputers, Computer Clusters, Mainframes, Servers, Workstations, Desktops, Handheld Devices, Real Time Systems, or Embedded Systems.

1. OS for Supercomputers:

Supercomputers are the fastest computers, very expensive and are employed for specialized applications that require immense amounts of mathematical calculations, for example, weather forecasting, animated graphics, fluid dynamic calculations, nuclear energy research, and petroleum exploration. Out of many operating systems used for supercomputing UNIX and Linux are the most dominant ones.

2. Computer Clusters Operating Systems:

A computer cluster is a group of computers that work together closely so that in many respects they can be viewed as though they are a single computer. The components of a cluster are commonly, connected to each other through fast local area networks. Besides many open source operating systems, and two versions of Windows 2003 Server, Linux is popularly used for Computer clusters.

3. Mainframe Operating Systems:

Mainframes used to be the primary procedure of computer. Mainframes are large centralized computers and at one time they provided the bulk of business computing through time sharing. Mainframes are still useful for some large scale tasks, such as centralized billing systems, inventory systems, database operations, etc.

4. Servers Operating Systems:

Servers are computers or groups of computers that provides services to other computers, connected via network. Based on the requirements, there are various versions of server operating systems from different vendors, starting with Microsoft's Servers from Windows NT to Windows 2003, OS/2 servers, UNIX servers, Mac OS servers, and various flavors of Linux.

5. Workstation Operating Systems:

Workstations are more powerful versions of personal computers. Like desktop computers, often only one person uses a particular workstation, and run a more powerful version of a desktop operating system. Most of the times workstations are used as clients in a network environment. The popular workstation operating systems are Windows NT Workstation, Windows 2000 Professional, OS/2 Clients, Mac OS, UNIX, Linux, etc.

6. Desktop Operating Systems:

A personal computer (PC) is a microcomputer whose price, size, and capabilities make it useful for individuals, also known as Desktop computers or home computers. Desktop operating systems are used for personal computers**,** for example DOS, Windows 9x, Windows XP, Macintosh OS, Linux, etc.

7. Embedded Operating Systems:

Embedded systems are combinations of processors and special software that are inside of another device, such as the electronic ignition system on cars. Examples of embedded operating systems are Embedded Linux, Windows CE, Windows XP Embedded, Free DOS, Free RTOS, etc.

8. Operating Systems for Handheld Computers:

Handheld operating systems are much smaller and less capable than desktop operating systems, so that they can fit into the limited memory of handheld devices. The operating systems include Palm OS, Windows CE, EPOC, and many Linux versions such as Qt Palmtop, and Pocket Linux, etc.

Self Assessment Questions

10. _____ are computers or groups of computers that provides services to other computers, connected via network.
11. _____are more powerful versions of personal computers.

12. _____systems are combinations of processors and special software that are inside of another device, such as the electronic ignition system on cars.

6.8 Summary

Let's recapitulate the important concepts discussed in this unit:

- A compiler is used to translate a high level language program into assembly language or machine code, and an assembler is used to translate an assembly language program into machine code.
- Resources Management – An operating system as resource manager controls how processes (the active agents) may access resources (passive entities).
- Multiprogramming (MP) increases CPU utilization by organizing jobs such that the CPU always has a job to execute.
- Multiprogramming features were overlaid on BP to ensure good utilization of CPU but from the point of view of a user the service was poor as the *response time,* i.e., the time elapsed between submitting a job and getting the results was unacceptably high.
- A real-time operating system is one, which helps to fulfill the worst-case response time requirements of an application.
- A task is a sub-computation in an application program, which can be executed concurrently with other sub-computations in the program, except at specific places in its execution called synchronization points.
- Primary – Memory or Main-Memory is a large array of words or bytes.
- A file is a collection of related information defined by its creator.
- A distributed system is a group of processors that do not share memory, peripheral devices, or a clock.
- A command interpreter is an interface of the operating system with the user.
- A computer cluster is a group of computers that work together closely so that in many respects they can be viewed as though they are a single computer.

6.9 Terminal Questions

1. What is Operating system? How it is useful in the computer?
2. Briefly explain functions of operating system.
3. What is resource management?
4. Define multi programming batch systems.
5. What is distributed system?
6. What is the role of real time operating system?

6.10 Answers

Self Assessment Questions

1. Abstraction
2. operating system
3. True
4. True
5. File
6. Distributed
7. True
8. True
9. Running a program
10. Servers
11. Workstations
12. Embedded

Terminal Questions

1. An operating system (OS) is a program that controls the execution of an application program and acts as an interface between the user and computer hardware. (Refer section 6.2)
2. Modern Operating systems generally have following three major goals. (Refer section 6.3)
3. An operating system as resource manager controls how processes (the active agents) may access resources (passive entities). (Refer section 6.3)
4. Multiprogramming (MP) increases CPU utilization by organizing jobs such that the CPU always has a job to execute. (Refer section 6.4.3)
5. A recent trend in computer system is to distribute computation among several processors. (Refer section 6.4.5)

6. Real-time (RT) operating systems were developed to meet the response requirements of such applications. (Refer section 6.4.6)

Book References:

- Modern Operating Systems (3rd Edition) by Andrew S. Tanenbaum
- Operating System Concepts Eight Edition by Avi Silberschatz, Peter Baer Galvin & Greg Gagne
- Operating Systems Internals and Design Principles by William Stallings

E-References

- www.computerhope.com
- www.howstuffworks.com
- www.personal.kent.edu
- www.searchcio-midmarket.techtarget.com