Biyani Institute of Science & management

I Internal Examination Sep. 2019
Class: MCA-I
Subject- Operating System (MCA-103)

MM: 20 Set: B Time: 1½Hrs

[I] Answer the following questions in one line only (2 ×1=2)

Q.1 Define timesharing and parallel Operating System?

Time-sharing operating systems

Time-sharing is a technique which enables many people, located at various terminals, to use a particular computer system at the same time. Time-sharing or multitasking is a logical extension of multiprogramming. Processor's time which is shared among multiple users simultaneously is termed as time-sharing.

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

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

The operating system uses CPU scheduling and multiprogramming to provide each user with a small portion of a time. Computer systems that were designed primarily as batch systems have been modified to time-sharing systems.

Advantages of Timesharing operating systems are as follows −

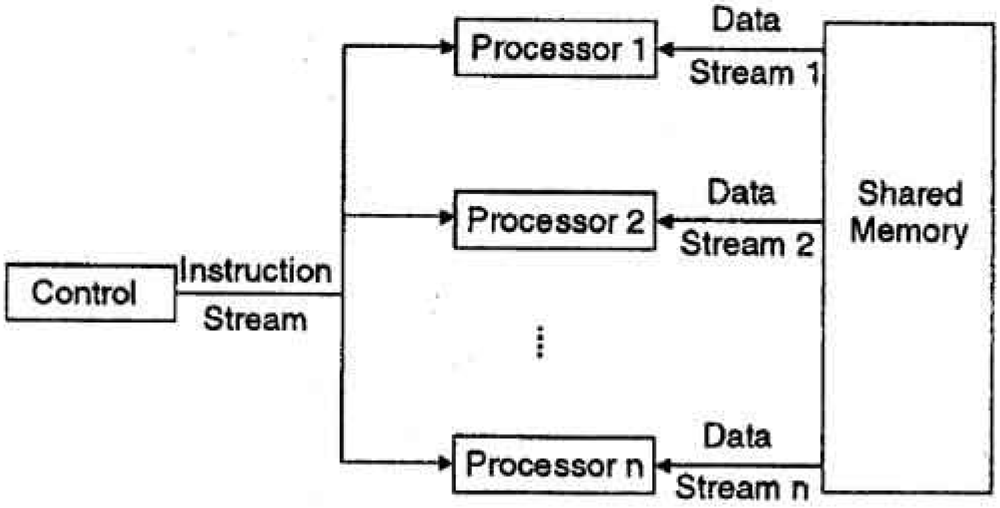
- Provides the advantage of quick response.
- Avoids duplication of software.
- Reduces CPU idle time.

Disadvantages of Time-sharing operating systems are as follows −

- Problem of reliability.
- Question of security and integrity of user programs and data.
- Problem of data communication.
- **Parallel Processing Systems** are designed to speed up the execution of programs by dividing the program into multiple fragments and processing these fragments simultaneously. Such systems are multiprocessor systems also known as tightly coupled systems. Parallel systems deal with the simultaneous use of multiple computer resources that can include a single computer with multiple processors, a number of computers connected by a network to form a parallel processing cluster or a combination of both.
- • Parallel computing is an evolution of serial computing where the jobs are broken into discrete parts that can be executed concurrently. Each part is further broken down to

a series of instructions. Instructions from each part execute simultaneously on different CPUs.

- • Parallel systems are more difficult to program than computers with a single processor because the architecture of parallel computers varies accordingly and the processes of multiple CPUs must be coordinated and synchronized. Several models for connecting processors and memory modules exist, and each topology requires a different programming model. The three models that are most commonly used in building parallel computers include synchronous processors each with its own memory, asynchronous processors each with its own memory and asynchronous processors with a common, shared memory. Flynn has classified the computer systems based on parallelism in the instructions and in the data streams. These are:
- 1. Single instruction stream, single data stream (SISD).
- 2. Single instruction stream, multiple data stream (SIMD).
- 3. Multiple instruction streams, single data stream (MISD).
- 4. Multiple instruction stream, multiple data stream (MIMD).
- The above classification of parallel computing system is focused in terms of two independent factors: the number of data streams that can be simultaneously processed, and the number of instruction streams that can be simultaneously processed. Here 'instruction stream' we mean an algorithm that instructs the computer what to do whereas 'data stream' (i.e. input to an algorithm) we mean the data that are being operated upon.
- • Even though Flynn has classified the computer 'systems into four types based on parallelism but only two of them are relevant to parallel computers. These are SIMD and MIMD computers.
- • SIMD computers are consisting of ‘n' processing units receiving a single stream of instruction from a central control unit and each processing unit operates on a different piece of data. Most SIMD computers operate synchronously using a single global dock. The block diagram of SIMD computer is shown below:



Q.2 Write Operating system components**?**

Components of Operating System

- Kernel.
- Process Execution.
- Interrupt.
- Memory Management.
- Multitasking.
- Networking.
- Security.

- User Interface.

[II] Short Answer Questions (Max 80 words). (2 × 5=10)

Q.1What do you mean by System calls, system programs?

The interface between a process and an operating system is provided by system calls. In general, system calls are available as assembly language instructions. They are also included in the manuals used by the assembly level programmers. System calls are usually made when a process in user mode requires access to a resource. Then it requests the kernel to provide the resource via a system call.

A figure representing the execution of the system call is given as follows:

As can be seen from this diagram, the processes execute normally in the user mode until a system call interrupts this. Then the system call is executed on a priority basis in the kernel mode. After the execution of the system call, the control returns to the user mode and execution of user processes can be resumed.

In general, system calls are required in the following situations:

- If a file system requires the creation or deletion of files. Reading and writing from files also require a system call.
- Creation and management of new processes.
- Network connections also require system calls. This includes sending and receiving packets.
- Access to a hardware devices such as a printer, scanner etc. requires a system call.

Q.2 Explain the Operating system structure?

An operating system is a construct that allows the user application programs to interact with the system hardware. Since the operating system is such a complex structure, it should be created with utmost care so it can be used and modified easily. An easy way to do this is to create the operating system in parts. Each of these parts should be well defined with clear inputs, outputs and functions.

Simple Structure

There are many operating systems that have a rather simple structure. These started as small systems and rapidly expanded much further than their scope. A common example of this is MS-DOS. It was designed simply for a niche amount for people. There was no indication that it would become so popular.

An image to illustrate the structure of MS-DOS is as follows:

It is better that operating systems have a modular structure, unlike MS-DOS. That would lead to greater control over the computer system and its various applications. The modular structure would also allow the programmers to hide information as required and implement internal routines as they see fit without changing the outer specifications.

Layered Structure

One way to achieve modularity in the operating system is the layered approach. In this, the bottom layer is the hardware and the topmost layer is the user interface.

An image demonstrating the layered approach is as follows:

As seen from the image, each upper layer is built on the bottom layer. All the layers hide some structures, operations etc from their upper layers.

One problem with the layered structure is that each layer needs to be carefully defined. This is necessary because the upper layers can only use the functionalities of the layers below them

[III]Answer the following questions in 150 words. (2 × 6=12)

Q.1 Explain the CPU scheduling criteria?

Another component involved in the CPU scheduling function is the Dispatcher. The dispatcher is the module that gives control of the CPU to the process selected by the short-term scheduler. This function involves:

- Switching context
- Switching to user mode
- Jumping to the proper location in the user program to restart that program from where it left last time.

The dispatcher should be as fast as possible, given that it is invoked during every process switch. The time taken by the dispatcher to stop one process and start another process is known as the Dispatch Latency. Dispatch Latency can be explained using the below figure:

Types of CPU Scheduling

CPU scheduling decisions may take place under the following four circumstances:

1. When a process switches from the running state to the waiting state(for I/O request or invocation of wait for the termination of one of the child processes).
2. When a process switches from the running state to the ready state (for example, when an interrupt occurs).
3. When a process switches from the waiting state to the ready state(for example, completion of I/O).
4. When a process terminates.

CPU Scheduling: Scheduling Criteria

There are many different criterias to check when considering the "best" scheduling algorithm, they are:

CPU Utilization

To make out the best use of CPU and not to waste any CPU cycle, CPU would be working most of the time(Ideally 100% of the time). Considering a real system, CPU usage should range from 40% (lightly loaded) to 90% (heavily loaded.)

Throughput

It is the total number of processes completed per unit time or rather say total amount of work done in a unit of time. This may range from 10/second to 1/hour depending on the specific processes.

Turnaround Time

It is the amount of time taken to execute a particular process, i.e. The interval from time of submission of the process to the time of completion of the process(Wall clock time).

Waiting Time

The sum of the periods spent waiting in the ready queue amount of time a process has been waiting in the ready queue to acquire get control on the CPU.

Load Average

It is the average number of processes residing in the ready queue waiting for their turn to get into the CPU.

Response Time

Amount of time it takes from when a request was submitted until the first response is produced. Remember, it is the time till the first response and not the completion of process execution(final response).

In general CPU utilization and Throughput are maximized and other factors are reduced for proper optimization.

Q.2 Explain the Following terms:

(a) Process Management

Process Management Introduction

A Program does nothing unless its instructions are executed by a CPU. A program in execution is called a process. In order to accomplish its task, process needs the computer resources.

There may exist more than one process in the system which may require the same resource at the same time. Therefore, the operating system has to manage all the processes and the resources in a convenient and efficient way.

Some resources may need to be executed by one process at one time to maintain the consistency otherwise the system can become inconsistent and deadlock may occur.

The operating system is responsible for the following activities in connection with Process Management

1. Scheduling processes and threads on the CPUs.
2. Creating and deleting both user and system processes.

3. Suspending and resuming processes.
4. Providing mechanisms for process synchronization.
5. Providing mechanisms for process communication.

(b) Multiple-processor scheduling

Multiprocessor Operating system

A multiprocessor system consists of several processors which share memory. In the multiprocessor, there is more than one processor in the system. The reason we use multiprocessor is that sometimes load on the processor is very high but input output on other function is not required. This type of operating system is more reliable as even if on processor goes down the other can still continues to work. This system is relatively cheap because we are only having the copies of processor but other devices like input-output and Memory are shared. In the multiprocessor system all the processor operate under the single operating system. Multiplicity of the processor and how the processors work together are transparent to the other.

In this, the user does not know in which processor their process work. A process is divided into several small processes and they work independently on the different processor. A system can be both multi-programmed by having multiple programs running at the same time and multiprocessing by having more than one physical and the processor.

In this diagram, there are more than 1 CPU and they shared a common memory.

Multiprocessing scheduling

In the multiprocessor scheduling, there are multiple CPU's which share the load so that various process run simultaneously. In general, the multiprocessor scheduling is complex as compared to single processor scheduling. In the multiprocessor scheduling, there are many processors and they are identical and we can run any process at any time.

The multiple CPU's in the system are in the close communication which shares a common bus, memory and other peripheral devices. So we can say that the system is a tightly coupled system. These systems are used when we want to process a bulk amount of data. These systems are mainly used in satellite, weather forecasting etc.

Multiprocessing system work on the concept of symmetric multiprocessing model. In this system, each processor work on the identical copy of the operating system and these copies communicate with each other. We the help of this system we can save money because of other devices like peripherals. Power supplies and other devices are shared. The most important thing is that we can do more work in a short period of time. If one system fails in the multiprocessor system the whole system will not halt only the speed of the processor will be slow down. The whole performance of the multiprocessing system is managed by the operating system . operating system assigns different task to the different processor in the system. In the multiprocessing system, the process is broken into the thread which they can be run independently. These type of system allow the threads to run on more than one processor simultaneously. In these systems the various process in the parallel so this is called parallel processor. Parallel processing is the ability of the CPU to run various process simultaneously. In the multiprocessing system, there is dynamically sharing of resources among the various processors.

Multiprocessor operating system is a kind of regular OS which handles many systems calls at the same time, do memory management, provide file management also the input-output devices.

There are some extra features which multiprocessor perform:

- Process synchronization
- Resource management
- Scheduling

There are various organizations of multiprocessor operating system:

1. Each CPU has its own OS

In this types of the organization then there are much Central processing units in the system and each CPU has its own private operating system and memory is shared among all the processors and input-output system are also shared. All the system is connected by the single bus.

2. Master slave multiprocessor

In this type of multiprocessor model, there is a single data structure which keeps track of the ready processes. In this model, one central processing unit works as master and other central processing unit work as a slave. In this, all the processors are handled by the single processor which is called master server. The master server runs the operating system process and the slave server run the user processes. The memory and input-output devices are shared among all the processors and all the processor are connected to a common bus. This system is simple and reduces the data sharing so this system is called Asymmetric multiprocessing.

3. Symmetric multiprocessor

Symmetric Multiprocessors (SMP) is the third model. In this model, there is one copy of the OS in memory, but any central processing unit can run it. Now, when a system call is made, then the central processing unit on which the system call was made traps to the kernel and then processes that system call. This model balances processes and memory dynamical. This approach uses Symmetric Multiprocessing where each processor is self-scheduling. The scheduling proceeds further by having the scheduler for each processor examine the ready queue and select a process to execute. In this system, this is possible that all the process may be in common ready queue or each processor may have its own private queue for the ready process.

There are mainly three sources of contention that can be found in a multiprocessor operating system.

- Locking system

As we know that the resources are shared in the multiprocessor system so there is a need to protect these resources for safe access among the multiple processors. The main purpose of locking scheme is to serialize access of the resources by the multiple processors.

- Shared data

When the multiple processor access the same data at the same time then there may be a chance of inconsistency of data so to protect this we have to use some protocols or locking scheme.

- Cache coherence

It is the shared resource data which is stored in the multiple local caches. Suppose there are two clients have a cached copy of memory and one client change the memory block and the other client could be left with invalid cache without notification of the change so this kind of conflict can be resolved by maintaining a coherence view of the data.