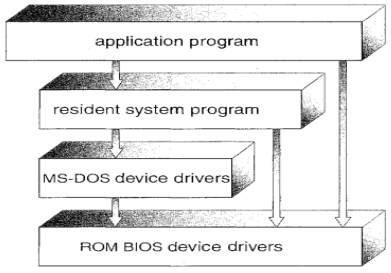
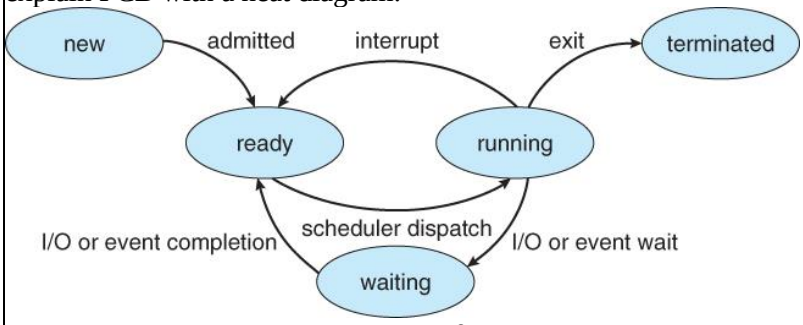
USN | | | | | | | | | | |

Internal Assessment Test 1 – July 2023

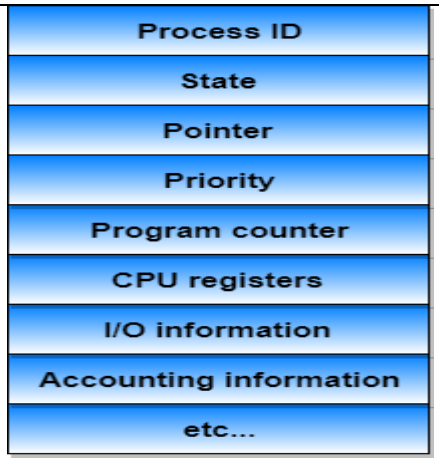
| Sub: | **Operating Systems** | | | | | Sub Code: | 21CS44 | Branch: | ISE | |
|---|---|---|---|---|---|---|---|---|---|---|
| Date: | **07/07/2023** | Duration: | 90 min's | Max Marks: | 50 | Sem/Sec: | IV A, B & C | | OBE | |

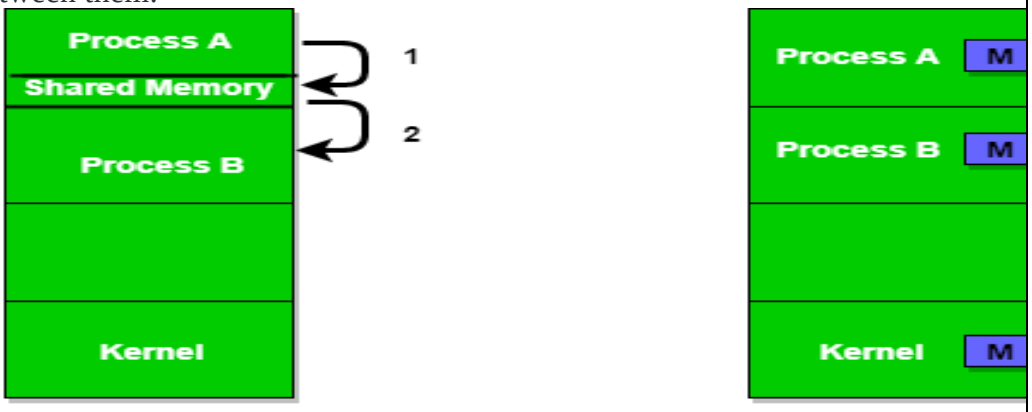
**Answer any FIVE FULL Questions**

| | | MARKS | CO | RBT |
|---|---|---|---|---|
| 1a) | What is operating system? What are functions of operating system?<br>An **operating system** is a piece of software that manages all the resources of a computer system, both hardware and software, and provides an environment in which the user can execute his/her programs in a convenient and efficient manner. Operating systems exist because they offer a reasonable way to solve the problem of creating a usable computer system.<br>An operating system functions<br>• manages the computer hardware<br>• facilitates execution of application programs<br>• acts as an intermediary between the user and the computer hardware<br>• designed to be convenient and efficient<br>• Process management<br>creating and deleting user and system processes<br>suspending and resuming processes<br>interprocess communication<br>• process synchronization<br>• deadlock handling<br>• Memory management<br>Keeping track of which part of memory is being used by which job<br>Allocating and deallocating memory space<br>• Storage management<br>• file system management<br>creating, deleting and manipulating files and directories<br>mass storage management<br>free space management<br>storage allocation<br>disk scheduling<br>Caching<br>• Input-output management | 5 | CO1 | L1 |
| 1b) | Explain MS-DOS layer structure with a neat diagram.<br>  <br><br>**Figure 2.12** MS-DOS layer structure.<br>• It was written to provide the most functionality in the least space, so it was not divided into modules carefully.<br>• In MS-DOS, the interfaces and levels of functionality are not well separated. For instance, application programs are able to access the basic I/O routines to write directly to the display and disk drives.<br>• Such freedom leaves MS-DOS vulnerable to errant (or malicious) programs, causing entire system crashes when user programs fail. Of course, MS-DOS was also limited by the hardware of its era. | 5 | CO1 | L2 |

| | | | | |
|---|---|---|---|---|
| | • Another exampleof limited structuring is the original UNIX operating system. UNIX is another system that initially was limited by hardware functionality.<br>• It consists of two separable parts: the kernel and the system programs. The kernel is further separated into a series of interfaces and device drivers, which have been added and expanded over the years as UNIX has evolved. | | | |
| 2 | Differentiate between a process and a program. Describe the life cycle of a process. Also explain PCB with a neat diagram.<br>  <br><br>There is a Process Control Block for each process, enclosing all the information about the process. It is also known as the task control block. It is a data structure, which contains the following:<br>• **Process State**: It can be running, waiting, etc.<br>• **Process ID** and the **parent process ID**.<br>• CPU registers and Program Counter. **Program Counter** holds the address of the next instruction to be executed for that process.<br>• **CPU Scheduling** information: Such as priority information and pointers to scheduling queues.<br>• **Memory Management information**: For example, page tables or segment tables.<br>• **Accounting information**: The User and kernel CPU time consumed, account numbers, limits, etc.<br>• **I/O Status information**: Devices allocated, open file tables, etc. | 10 | CO1 | L2 |

| Process | Program |
|---|---|
| The process is basically an instance of the computer program that is being executed. | A Program is basically a collection of instructio specific task when executed by the computer. |
| A process has a **shorter lifetime**. | A Program has a **longer lifetime**. |
| A Process requires resources such as memory, CPU, Input-Output devices. | A Program is stored by hard-disk and does not r |
| A process has a dynamic instance of code and data | A Program has static code and static data. |
| Basically, a process is the **running instance** of the code. | On the other hand, the program is the **executabl** |




| | | | | |
|---|---|---|---|---|
| 3a) | What is interprocess communication? Explain the concept of Message passing and shared memory.<br>A process can be of two types:<br>• Independent process.<br>• Co-operating process.<br>An independent process is not affected by the execution of other processes while a co-operating process can be affected by other executing processes. The communication between these processes can be seen as a method of co-operation between them.<br>  <br><br>**Figure 1 -** Shared Memory and Message Passing | 10 | CO1 | L2 |
| 4a) | What do you mean by System Calls? Explain the different modes of OS?<br>A system call is a way for a user program to interface with the operating system. The program requests several services, and the OS responds by invoking a series of system calls to satisfy the request. A system call can be written in assembly language or a high-level language like **C** or **Pascal**. System calls are predefined | 5 | CO1 | L1 |

functions that the operating system may directly invoke if a high-level

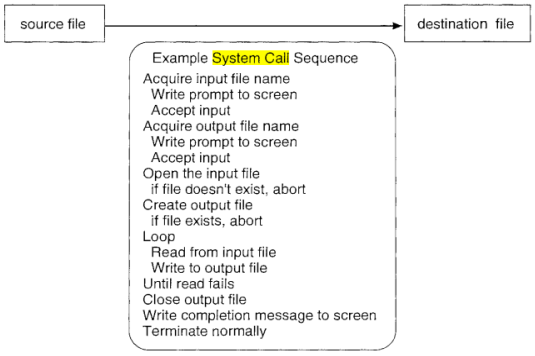



**Figure 2.4** Example of how system calls are used.

language is used.
Kernel Mode
User Mode

| | | | | |
|---|---|---|---|---|
| 4b) | Distinguish between: i) I/O bound and CPU bound ii) Multiprogramming and multitasking . | 5 | CO2 | L2 |

| Parameter | Multiprogramming | Multitasking |
|---|---|---|
| Definition | It lets multiple programs use the CPU at the same time. | Multitasking refers to the simultaneous execution of numerous programmes, processes, and threads with the certain timestamp. |
| Objective | It's useful for cutting down on CPU idle time and boosting throughput as much as possible. | It may be used to execute numerous tasks at once, significantly improving CPU and system throughput. |
| Mechanism | The context switching method is used. | Based on a time-sharing mechanism. |
| Time | Multiprogramming demands comparatively more time to complete tasks. | Multitasking allows you to complete tasks in less time. |
| Execution | In a multi-programmed system, when one job or process completes its execution or changes to an I/O task, the system temporarily suspends that process. It chooses a new process to execute from the process scheduling pool. | Multiple processes can run concurrently in this system by assigning the CPU for a fixed duration of time. |
| CPU Switching | The CPU shifts between processes swiftly in this environment. | The CPU shifts between the processes of several programs in a single-user environment. |
| CPU required | Only one CPU is needed in Multiprogramming to run the tasks. | Multiple CPUs are required for the task allocation. |

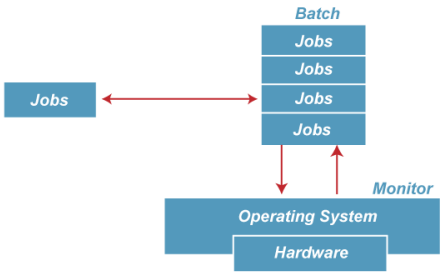
Multiprogramming vs. Multitasking

| | | | | |
|---|---|---|---|---|
| 5b) | Consider the set of given process with the Burst time Calculate the Average Waiting Time and Turn Around Time for Shortest Job First (SJF) and First Come First Serve(FCFS).<br><br>**First Come First Serve(FCFS) Avg Waiting Time 13.33**<br>**Avg Turn Around Time 26.23**<br>**Shortest Job First(SJF) Avg Waiting Time 3.33**<br>**Avg Turn Around Time 16.67** | 10 | CO2 | L3 |

| Process | Burst Time |
|---|---|
| P1 | 3 |
| P2 | 33 |
| P3 | 4 |

| | | | | |
|---|---|---|---|---|
| 6 | Explain any three types of OS.<br><br>**Batch Processing System**<br><br>In the 1970s, Batch processing was very popular. In this technique, similar types of jobs were batched together and executed in time. People were used to having a single computer which was called a mainframe.<br><br>• In Batch operating system, access is given to more than one person; they submit their respective jobs to the system for the execution. | 10 | CO1 | L2 |

- The system put all of the jobs in a queue on the basis of first come first serve and then executes the jobs one by one. The users collect their respective output when all the jobs get executed.




- The purpose of this operating system was mainly to transfer control from one job to another as soon as the job was completed.
- It contained a small set of programs called the resident monitor that always resided in one part of the main memory. The remaining part is used for servicing jobs.

**Multiprogramming Operating System**

- Operating system that executes more than one program using a single processor machine.
- More than one task or program or jobs are present inside the main memory at one point of time.
- Buffering and spooling can overlap I/O and CPU tasks to improve the system performance but it has some limitations that a single user cannot always keep CPU or I/O busy all the time.
- The OS could pick and start the execution of one of the jobs in memory, whenever the jobs does not need CPU that means the job is working with I/O at that time the CPU is idle at that time the OS switches to another job in memory and CPU executes a portion of it till the job issues a request for I/O and so on.

**Faculty Signature** **CCI Signature** **HOD Signature**

**Faculty Signature** **CCI Signature** **HOD Signature**