

Programme Name: \_\_\_\_\_\_\_\_\_\_\_\_\_BCS\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Course Code: \_\_\_CSC 1016\_\_\_\_\_\_\_

Course Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Operating System\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Assignment / Lab Sheet / Project / Case Study No. \_\_1\_\_

Date of Submission: \_\_\_\_\_\_\_\_\_\_\_5/06/2020\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Submitted By: Submitted To:**

Student Name: Dipesh Tha Shrestha Faculty Name: Kushal Regmi

IUKL ID: Department: LMS

Semester: Second Semester

Intake: September 2019

1. **What is Operating System? Explain different functions of an operating system. [10 marks]**

Answer= An Operating System (OS) is an interface between a computer user and computer hardware. An operating system is a software which performs all the basic tasks like file management, memory management, process management, handling input and output, and controlling peripheral devices such as disk drives and printers.

Some popular Operating Systems include Linux Operating System, Windows Operating System, VMS, OS/400, AIX, z/OS, etc.

The different functions of an operating system are:

* Memory Management
* Processor Management
* Device Management
* File Management
* Security
* Control over system performance
* Job accounting
* Error detecting aids
* Coordination between other software and users
* **Memory Management**

Memory management refers to management of Primary Memory or Main Memory. Main memory is a large array of words or bytes where each word or byte has its own address.

Main memory provides a fast storage that can be accessed directly by the CPU. For a program to be executed, it must in the main memory. An Operating System does the following activities for memory management −

* Keeps tracks of primary memory, i.e., what part of it are in use by whom, what part are not in use.
* In multiprogramming, the OS decides which process will get memory when and how much.
* Allocates the memory when a process requests it to do so.
* De-allocates the memory when a process no longer needs it or has been terminated.
* **Processor Management**

In multiprogramming environment, the OS decides which process gets the processor when and for how much time. This function is called processscheduling. An Operating System does the following activities for processor management −

* Keeps tracks of processor and status of process. The program responsible for this task is known as traffic controller.
* Allocates the processor (CPU) to a process.
* De-allocates processor when a process is no longer required.
* **Device Management**

An Operating System manages device communication via their respective drivers. It does the following activities for device management −

* Keeps tracks of all devices. Program responsible for this task is known as the I/O controller.
* Decides which process gets the device when and for how much time.
* Allocates the device in the efficient way.
* De-allocates devices.
* **File Management**

A file system is normally organized into directories for easy navigation and usage. These directories may contain files and other directions.

An Operating System does the following activities for file management −

* Keeps track of information, location, uses, status etc. The collective facilities are often known as file system.
* Decides who gets the resources.
* Allocates the resources.
* De-allocates the resources.
* Other Important Activities

Following are some of the important activities that an Operating System performs −

* **Security** − By means of password and similar other techniques, it prevents unauthorized access to programs and data.
* **Control over system performance** − Recording delays between request for a service and response from the system.
* **Job accounting** − Keeping track of time and resources used by various jobs and users.
* **Error detecting aids** − Production of dumps, traces, error messages, and other debugging and error detecting aids.
* **Coordination between other softwares and users** − Coordination and assignment of compilers, interpreters, assemblers and other software to the various users of the computer systems.

1. **Describe the different scheduling criteria. Differentiate the Pre-emptive and Non-preemptive scheduling approaches**

Answer= 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**

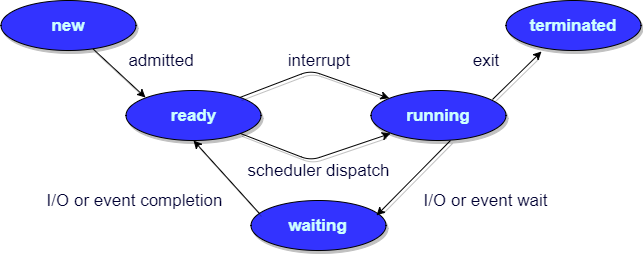
Amount of time it takes from when a request was submitted until the first response is produced. Remember, it is the time till the first response and not the completion of process execution(final response).

Differentiate the Pre-emptive and Non-preemptive scheduling approaches is given below:

| **PARAMENTER** | **PREEMPTIVE SCHEDULING** | **NON-PREEMPTIVE SCHEDULING** |
| --- | --- | --- |
| Basic | In this resources(CPU Cycle) are allocated to a process for a limited time. | Once resources(CPU Cycle) are allocated to a process, the process holds it till it completes its burst time or switches to waiting state. |
| Interrupt | Process can be interrupted in between. | Process can not be interrupted untill it terminates itself or its time is up. |
| Starvation | If a process having high priority frequently arrives in the ready queue, low priority process may starve. | If a process with long burst time is running CPU, then later coming process with less CPU burst time may starve. |
| Overhead | It has overheads of scheduling the processes. | It does not have overheads. |
| Flexibility | flexible | rigid |
| Cost | cost associated | no cost associated |
| CPU Utilization | In preemptive scheduling, CPU utilization is high. | It is low in non preemptive scheduling. |
| Examples | Examples of preemptive scheduling are Round Robin and Shortest Remaining Time First. | Examples of non-preemptive scheduling are First Come First Serve and Shortest Job First. |

1. **Using appropriate diagram, explain States-of-a- Process visits during the course of its execution**

Answer:



The different states of a process visits during the course of its execution are:

* **New (Create) –** In this step, the process is about to be created but not yet created, it is the program which is present in secondary memory that will be picked up by OS to create the process.
* **Ready –** New -> Ready to run. After the creation of a process, the process enters the ready state i.e. the process is loaded into the main memory. The process here is ready to run and is waiting to get the CPU time for its execution. Processes that are ready for execution by the CPU are maintained in a queue for ready processes.
* **Run –** The process is chosen by CPU for execution and the instructions within the process are executed by any one of the available CPU cores.
* **Blocked or wait –** Whenever the process requests access to I/O or needs input from the user or needs access to a critical region(the lock for which is already acquired) it enters the blocked or wait state. The process continues to wait in the main memory and does not require CPU. Once the I/O operation is completed the process goes to the ready state.
* **Terminated or completed –** Process is killed as well as PCB is deleted.
* **Suspend ready –** Process that was initially in the ready state but were swapped out of main memory(refer Virtual Memory topic) and placed onto external storage by scheduler are said to be in suspend ready state. The process will transition back to ready state whenever the process is again brought onto the main memory.
* **Suspend wait or suspend blocked –** Similar to suspend ready but uses the process which was performing I/O operation and lack of main memory caused them to move to secondary memory.  
  When work is finished it may go to suspend ready.