**Ques1. What is the main Function of an operating System?**

**Ans .** An Operating System has following three Functions:

(1) Manage the computer's resources, such as the central processing unit, memory, disk drives, and printers

(2) establish a user interface

(3) execute and provide services for applications software.

**Ques2. What does the CPU do when it has no program to execute?**

**Ans .** The time when CPU has No Program to execute it simply shits idle.

**Ques3. Int main()**

**{**

**int i, int j;**

**scanf(“%d”,&i);**

**for(j=0;j<i;j++)**

**{**

**sum=j+i;**

**}**

**printf(“%d”,sum);**

**exit(0);**

**}**

**In the above problem, differentiate each and every line of code as CPU execution or I/O execution.**

**Ans.** In the above code **line1:** Int main (), **line3:** int I, int j, **line 5:** For loop, **line 7:** sum,

and **line 10** exit () are CPU executable code i.e. CPU will be allotted to the process.

**Line 4:** scanf ( ), **line 9:** printf ( ) are I/O execution. The I/O like printer and scanner will be allotted to process.

**Ques4. Difference between Multi-Programing, Multi-Tasking, and Multi-Processing?**

**Ans.**

|  |  |  |
| --- | --- | --- |
| **Multi-Programing** | **Multi-tasking** | **Multi-processing** |
| CPU will be allotted to process one after other once.  Less throughput, and reliable than other two. | CPU will be multiplexing without completing any job.  More Speed and reliable than Multi-Programing but less than Multi-processing. | Contain many CPU and all CPU can execute Process simultaneously.  Highest throughput, faster and more realiable. |

**Ques5. Difference between a program and a process?**

**Ans.**

|  |  |
| --- | --- |
| **Program** | **Process** |
| A program is a set of instruction or code that are designed to perform a specific task. | A process is an operation that perform the functions or manipulation as per the code for execution. |

**Ques6. Categorize the following state:**

1. **New:** A Process present in a secondary memory and is about to execute or if

Executed is called new process.

1. **Ready:** In the main memory there are many processes present. It is up to you

which you are going to run.

1. **Run:** The process has assigned the memory or given CPU for execution.
2. **Block or Wait:** Sometime a processgoes for input output, at that time process simply

transferred to block state. Once it is finished the process is again

assigned to process.

1. **Termination:** Once the process is finished its job successfully it is simply killed

or terminated. Even the context(PCB) is also deleted.

1. **Suspend-Ready:** Sometime 100 processes are present in ready state in main memory

And you want to use the memory for some other purpose. In that process we can remove the process for sometime.

1. **Suspend-block:**  It is good to suspend the process that are already blocked, for

time being put the block processes in the hard disk as they are doing input output.

**Ques7. Define the term Context Switch with an example?**

**Ans.** A context switch occurs when a computer's [CPU](https://www.computerhope.com/jargon/c/cpu.htm) switches from one process or thread to a different process or thread.

Context switching allows for one CPU to handle numerous processes or threads without the need for additional processors. Any operating system that allows for multitasking relies heavily on the use of context switching to allow different processes to run at the same time. Typically, there are three situations that a context switch is necessary, as shown below.

Multitasking - When the CPU needs to switch processes in and out of memory, so that more than one process can be running.

Kernel/User Switch - When switching between user mode to kernel mode, it may be used (but isn't always necessary).

Interrupts - When the CPU is interrupted to return data from a disk read.

**Ques8. Consider a system with ‘n’ CPU processors and ‘m’ processes, then answer the following queries regarding minimum and maximum number of processes:**

|  |  |  |
| --- | --- | --- |
|  | **Minimum** | **Maximum** |
| **Ready** | **0** | **n** |
| **Running** | **0** | **m** |
| **Block** | **0** | **m** |

**Ques9. Define the responsibilities of:**

1. **Short term scheduler:** Short term scheduling concerns with the allocation of CPU time to processes in order to meet some pre-defined system performance objectives. The definition of these objectives (scheduling policy) is an overall system design issue, and determines the ``character'' of the operating system from the user's (i.e. the buyer's) point of view, giving rise to the traditional distinctions among ``multi-purpose, time shared'', ``batch production'', ``real-time'' systems, and so on.
2. **Long term Scheduler:** Long term scheduling obviously controls the degree of multiprogramming in multitasking systems, following certain policies to decide whether the system can honour a new job submission or, if more than one job is submitted, which of them should be selected. The need for some form of compromise between degree of multiprogramming and throughput seems evident, especially when one considers interactive systems. The higher the number of processes, in fact, the smaller the time each of them may control CPU for, if a fair share of responsiveness is to be given to all processes.
3. **Mid-term Scheduler:** The use of medium term scheduler is to improve multiprogramming by allowing multiple processes to reside in main memory by swapping out processes that are waiting (need I/O) or low priority processes and swapping in other processes that were in ready queue. So you can see that we requied medium term scheduler when we have limited memory. This swapping in and out operation does not take place when we are running a single small program and have large memory.

**Ques10. State some principles of giving better user interactivity**

## Ans. 1) Clarity is job.

## 2) Interfaces exist to enable interaction

## 3) Conserve attention at all costs

## 4) Keep users in control

## 5) Direct manipulation is best

## 6) One primary action per screen

## Ques11. Why can’t a virus just take hold of a system and keeps running? Think and

## try to answer.

## Ans. The virus is mostly present in executable files like .com and .exe files. The don’t

## Run randomly and not in a same fashion again and again, as it will be very easy

## to track them. They increases the size of the file and make processor busy

## results in slowing the system ready.

## Ques12. What is the responsibility of dispatcher?

## Ans. Dispatcher obey the command Short term scheduler and sends the new

## processes from queue to the memory.

## Ques13. What are the applications of the real time operating System?

## Ans. In real time operating System you will be given jobs and all the jobs will have

## some deadline. You are supposing the jobs in the given deadline.

## Ques14. What do you understand by the term System Call?

## Ans. System call provides an interface to the services by an operating System.

## These Calls are generally available as routine written in C and C++ languages.

## Ques15. What is the use of fork and exec System call?

## Ans. Fork System call is used to create new child process from any parent process.

## Exec System Call will not execute same binary child process instead will execute

## different process.