1. **What are the functions of an operating system?**

Memory Management

Processor Management

Device Management

File Management

Security

Control over system performance

Job accounting

Error detecting aids

Coordination between other software and users

1. **Define Operating System.**

An **Operating system (OS)** is a software which acts as an interface between the end user and computer hardware. Every computer must have at least one OS to run other programs. An application like Chrome, MS Word, Games, etc needs some environment in which it will run and perform its task. The OS helps you to communicate with the computer without knowing how to speak the computer's language. It is **not** possible for the user to use any computer or mobile device without having an operating system.

1. **List services provided by OS.**

User Interface.

Program Execution.

File system manipulation.

Input / Output Operations.

Communication.

Resource Allocation.

Error Detection.

Accounting

1. **List out system components.**

Motherboard

Processor

Memory (RAM)

Floppy drive

Hard disk

CD-ROM, CD-RW, or DVD-ROM drive

Keyboard

Mouse

Video card

Monitor (display)

Sound card

Speakers

Modem

1. **Define buffering.**

The buffer stores transmitted data temporarily as it is going between devices or between a device and an app. A buffer in a computer environment means that a set amount of data is going to be stored in order to preload the required data right before it gets used by the CPU.

1. **What is multiprogramming?**

In a multiprogramming system there are one or more programs loaded in main memory which are ready to execute. Only one program at a time is able to get the CPU for executing its instructions. The main idea of multiprogramming is to maximize the use of CPU time. Indeed, suppose the currently running process is performing an I/O task then, the OS may interrupt that process and give the control to one of the other in-main-memory programs.

1. **What are the five major activities of an operating system with regard to file management?**

The creation and deletion of **files**.

The creation and deletion of directions.

The support of primitives for manipulating **files** and directions.

The mapping of **files** onto secondary storage.

The back up of **files** on stable storage media.

1. **What is Thread?**

Thread is an execution unit that is part of a process. A process can have multiple threads, all executing at the same time. It is a unit of execution in concurrent programming. A thread is lightweight and can be managed independently by a scheduler. It helps you to improve the application performance using parallelism.

1. **What is Micro-Kernel? What is the need to move from Kernels to Micro-Kernel?**

Microkernel is one of the classifications of the kernel. Being a kernel it manages all system resources. But in a microkernel, the **user services** and **kernel services** are implemented in different address space. The user services are kept in **user address space**, and kernel services are kept under **kernel address space**, thus also reduces the size of kernel and size of operating system as well.

1. **Differentiate user level and kernel level threads**

User level thread

1. Implemented by users
2. Context switch is less
3. **Inexpensive**
4. If one user process blocks then entire process will be blocked. Ex.Java thread

Kernel level thread

1. Implemented by OS
2. Context switch time is more
3. **More expensive**
4. If one thread blocks another thread will execute Ex. Windows, Solaris
5. **What is the purpose of system call?**

It provides an interface between a process and operating system to allow user-level processes to request services of the operating system. System calls are the only entry points into the kernel system. All programs needing resources must use system calls

1. **What is device driver?**

A device driver is a software program that controls a particular type of hardware device that is attached to a computer. When buying an operating system, many device drivers are built into the product.

1. **Define Process? Components**

A process is more than a program code. A process is an ‘active’ entity as oppose to program which considered being a ‘passive’ entity. As we all know that a program is an algorithm expressed in some programming language. Being a passive, a program is only a part of process. Process, on the other hand, includes:

* Current value of Program Counter (PC)
* Contents of the processors register
* Value of the variables
* The process stack, which typically contains temporary data such as subroutine parameter, return address, and temporary variables.
* A data section that contains global variables.
* A process is the unit of work in a system

1. **Differentiate process and thread?**

| **BASIS FOR COMPARISON** | **PROCESS** | **THREAD** |
| --- | --- | --- |
| Basic | Program in execution. | Lightweight process or part of it. |
| Memory sharing | Completely isolated and do not share memory. | Shares memory with each other. |
| Resource consumption | More | Less |
| Efficiency | Less efficient as compared to the process in the context of communication. | Enhances efficiency in the context of communication. |
| Time required for creation | More | Less |
| Context switching time | Takes more time. | Consumes less time. |
| Time required for termination | More | Less |

1. **What is process state and write all its state**

READY - **The process** is waiting to be assigned to a processor. RUNNING - Instructions are being executed. WAITING - **The process** is waiting for some event to occur(such as an I/O completion or reception **of** a signal). TERMINATED - **The process** has finished execution.

1. **Define PCB. What its contains?**

Process Control Block is a data structure that contains information of the process related to it. It also defines the current state of the operating system.

**Pointer –** It is a stack pointer which is required to be saved when the process is switched from one state to another to retain the current position of the process.

**Process state –** It stores the respective state of the process.

**Process number –** Every process is assigned with a unique id known as process ID or PID which stores the process identifier.

**Program counter –** It stores the counter which contains the address of the next instruction that is to be executed for the process.

**Register –** These are the CPU registers which includes: accumulator, base, registers and general purpose registers.

**Memory limits –** This field contains the information about memory management system used by operating system. This may include the page tables, segment tables etc.

**Open files list –** This information includes the list of files opened for a process.

1. **Differentiate preemptive and nonpreemptive scheduling policies**

| **BASIS FOR COMPARISON** | **PREEMPTIVE**  **SCHEDULING** | **NON PREEMPTIVE**  **SCHEDULING** |
| --- | --- | --- |
| Basic | The resources are allocated to a process for a limited time. | Once resources 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 till it terminates or switches to waiting state. |
| Starvation | If a high priority process frequently arrives in the ready queue, low priority process may starve. | If a process with long burst time is running CPU, then another process with less CPU burst time may starve. |
| Flexibility | Preemptive scheduling is flexible. | Non-preemptive scheduling is rigid. |
| Cost | Preemptive scheduling is cost associated. | Non-preemptive scheduling is not cost associative. |

1. **Define Throughput, Turnaround time.**

**Throughput:** Throughput is the amount of work completed in a unit of time. In other words throughput is the processes executed to number of jobs completed in a unit of time.

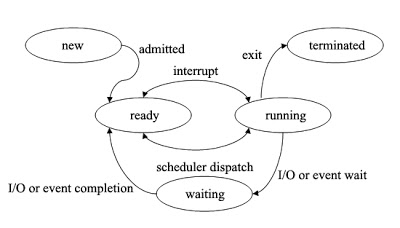
**Arrival Time:**Time at which the process arrives in the ready queue.  
**Completion Time:** Time at which process completes its execution.  
**Burst Time:**Time required by a process for CPU execution.  
**Turn Around Time:**Time Difference between completion time and arrival time.  
Turn Around Time = Completion Time – Arrival Time

**Waiting Time(W.T):** Time Difference between turn around time and burst time.  
Waiting Time = Turn Around Time – Burst Time

1. **What do you mean by a cooperating process?**

In the computer system, there are many processes which may be either independent processes or **cooperating processes** that run in the operating system. A process is said to be independent when it cannot affect or be affected by any other processes that are running the system. It is clear that any process which does not share any data (temporary or persistent) with any another process then the process independent. On the other hand, **a cooperating process is one** which can affect or affected by any another process that is running on the computer. the cooperating process is one which shares data with another process.

1. **Sketch the state diagram of a process.**



1. **Define resource?**

A system resource is any usable part of a computer that can be controlled and assigned by the [operating system](https://www.lifewire.com/operating-systems-2625912) so all of the [hardware](https://www.lifewire.com/computer-hardware-2625895) and software on the computer can work together as designed.

System resources are often talked about in relation to [system memory](https://www.lifewire.com/what-is-random-access-memory-ram-2618159) (your computer's RAM) but resources might also come from the [CPU](https://www.lifewire.com/what-is-a-cpu-2618150), the [motherboard](https://www.lifewire.com/motherboards-system-boards-and-mainboards-2618154), or even other hardware

1. **Define context switch?**

The process of saving the context of one process and loading the context of another process is known as Context Switching. In simple terms, it is like loading and unloading the process from running state to ready state.

1. **Distinguish between CPU-bounded and I/O bounded processes?**

If the process is intensive in terms of CPU operations then it is called CPU bound process. Similarly, If the process is intensive in terms of I/O operations then it is called IO bound process.

**IO Bound processes:** processes that perform lots of IO operations. Each IO operation is followed by a short CPU burst to process the IO, then more IO happens.

**CPU bound processes:** processes that perform lots of computation and do little IO. Tend to have a few long CPU bursts

**Pre-emption –** Process is forcefully removed from CPU. Pre-emption is also called as time sharing or multitasking.

**Non pre-emption –** Processes are not removed until they complete the execution.

1. **What are the scheduling policies adopted for a process?**

**Throughput:** Throughput is the amount of work completed in a unit of time. In other words throughput is the processes executed to number of jobs completed in a unit of time.

**Arrival Time:**Time at which the process arrives in the ready queue.  
**Completion Time:** Time at which process completes its execution.  
**Burst Time:**Time required by a process for CPU execution.  
**Turn Around Time:**Time Difference between completion time and arrival time.  
Turn Around Time = Completion Time – Arrival Time

**Waiting Time(W.T):** Time Difference between turn around time and burst time.  
Waiting Time = Turn Around Time – Burst Time

1. **List various CPU Scheduling algorithms?**

* First-Come First-Serve Scheduling, FCFS. ...
* Shortest-Job-First Scheduling, SJF. ...
* Priority Scheduling. ...
* Round Robin Scheduling. ...
* Multilevel Queue Scheduling. ...
* Multilevel Feedback-Queue Scheduling.

**PART-B**

1. Explain the services provided by the Operating System
2. What is purpose of system calls? Briefly write on the types of system calls provided by the operating system.
3. Explain various threading models?
4. Explain various system components?
5. Discuss in detail about various process states with a neat diagram and explain the contents of Process Control Block (PCB).
6. Assume you have the following jobs to execute with one processor:

**Process**  BurstTime(pi)

P0 80

P1 20

P2 10

P3 20

P4 50

1. Create a Gantt chart illustrating FCFS and SJF scheduling of executing these processes.
2. What is the turnaround time for p3?
3. What is the average wait time for the processes?
4. Write a brief note on the following process scheduling algorithms with example
   1. Round-Robin scheduling.
   2. Priority Scheduling.
5. Explain multilevel queue and feedback scheduling algorithm?
6. Define the term process and differentiate between heavyweight and lightweight processes. Assume that following jobs have arrived in the order 1,2,3,4 and 5 :

Job Arrival Time Burst Time Priority

1 0 15 2

2 2 03 1

3 5 05 5

4 6 08 4

5 7 12 3

Give Gantt chart and calculate Avg. Turn-around Time and Waiting Time for :

i) FCFS ii) SJF scheduling algorithm.

1. Describe the difference among short-term, medium-term and long-term scheduling with suitable example.
2. Consider the following processes with CPU burst time given in milliseconds.

**Time Slice = 3 ms**

|  |  |  |  |
| --- | --- | --- | --- |
| Process | Arrival time | Burst time | Priority |
| P1 | 0 | 3 | 2 |
| P2 | 3 | 2 | 1 |
| P3 | 7 | 2 | 1 |
| P4 | 10 | 7 | 3 |

Calculate average waiting time and average turn around time for the following algorithms.

(i) FCFS (ii) Preemptive SJF

(iii) Non Preemptive Priority (iv) Round Robin