Introduction to Process

Process:

The process is an example of a computer program used. Contains the program code and its current function.

Program vs Process

Process	Program
1.Program contains a set of instructions designed to complete a task.	1.Process is an instance of an executing program.
2.Lifespan of process is less than process	2.Lifespan of program is longer
3.Process exists for a limited span of time as it gets terminated after the completion of a task.	3.Program exists at a single place and continues to exist until it is deleted.
4.Process is a dynamic entity.	4. Program is a static entity.
5.The process has a high resource requirement, it requires resources such as CPU, memory address, O / O during its lifetime.	5. Program has no resource requirement, it only requires memory space to store commands.

Each process has a set of states that it keeps on changing during its life cycle.

Process States:

- **New State:** This is the state when the process is just created. It is the first state of a process.
- **Ready State**: After the creation of the process, when the process is ready for its execution then it goes into the ready state. In a ready state, the process is ready for its execution by the CPU but it is waiting for its turn to come. There can be more than one process in the ready state.

- **Ready Suspended State:** There can be more than one process in the ready state but due to memory constraint, if the memory is full then some process from the ready state gets placed in the ready suspended state.
- **Running State:** Amongst the process present in the ready state, the CPU chooses one process amongst them by using some CPU scheduling algorithm. The process will now be executed by the CPU and it is in the running state.
- **Waiting or Blocked State:** During the execution of the process, the process might require some I/O operation like writing on file or some more priority process might come. In these situations, the running process will have to go into the waiting or blocked state and the other process will come for its execution. So, the process is waiting for something in the waiting state.
- **Waiting Suspended State:** When the waiting queue of the system becomes full then some of the processes will be sent to the waiting suspended state.
- **Terminated State:** After the complete execution of the process, the process comes into the terminated state and the information related to this process is deleted.

The information about a particular process is stored in a per process data structure called Process Control Block.

Process Control Block(PCB)

Each process is represented in the operating system by a process control block (PCB) also called a task control block. It contains many pieces of information associated with a specific process, including these:

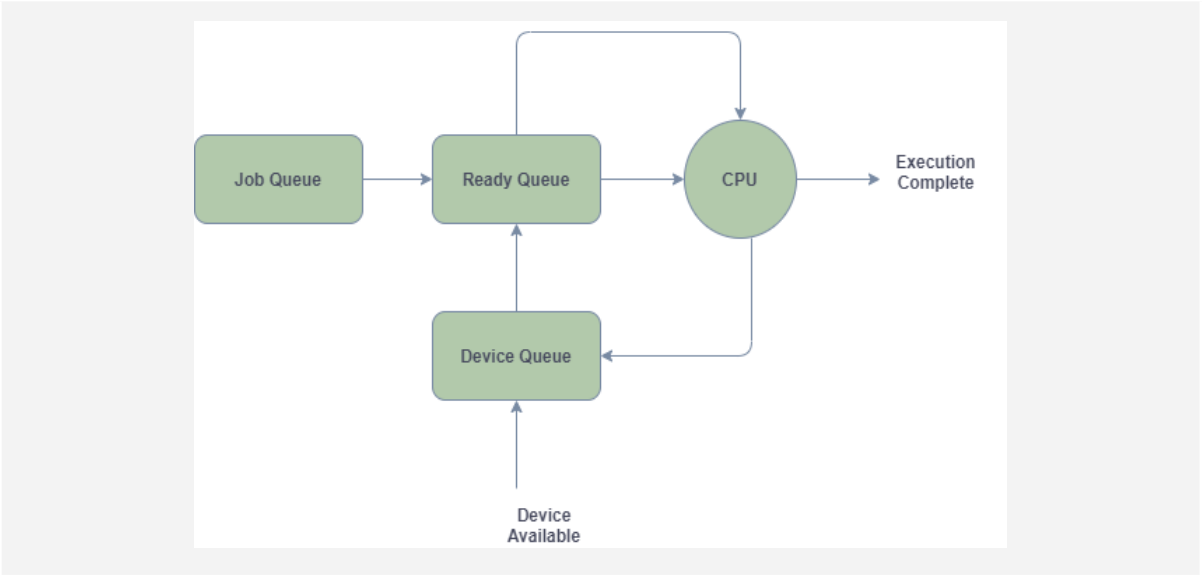
- **Process :** The state may be new, ready, running and so on
- **Program counter:** It indicates the address of the next instruction to be executed for this program.
- **CPU registers:** These vary in number and type based on architecture. They include accumulators, stack pointers, general purpose registers etc.
- **CPU scheduling :** This includes process priority, pointers to scheduling queues and any scheduling parameters.
- **Memory-management:** This includes the value of base and limit registers (protection) and page tables, segment tables depending on memory.
- **Accounting:** It includes amount of CPU and real time used, account numbers, process numbers etc
- **I/O status information:** It includes list of I/O devices allocated to this process, a list of open files etc

During the lifecycle of a process, it undergoes multiple state changes. These changes are facilitated by Process Schedulers. Process Schedulers use following scheduling queues.

Scheduling Queue

Scheduling queues refers to queues of processes or devices. When the process enters into the system, then this process is put into a job queue. This queue consists of all processes in the system. The operating system also maintains other queues such as device queue. Device queue is a queue for which multiple processes are waiting for a particular I/O device. Each device has its own device queue.

Queues are of three types:
Job Queue: As a process enters the system, it is put in a job queue that contains all the processes in the system.
Ready queue: The processes that are residing in the memory and are ready for execution are kept in the ready queue.The ready queue is implemented as a linked list of PCBs with a header containing pointers to the first and the last PCBs.
Waiting Queue or Device queue: The list of the processes waiting for a particular i/o device is called a device queue.



Context Switching

- Context switching is done to switch between processes.
- Switching the CPU to another process requires saving the state of the current process and reloading the state of another process. States are saved into and reloaded from PCBs.
- Context-switch time is a pure overhead as the system does not do any useful work during a control switch.
- Context-switch time depends highly on the hardware.
- Context switching is faster on RISC processors with overlapped register windows.

Apart from the process and its lifecycle, we learnt about various operations such as process creation and process termination. We also learnt about two types of processes: Orphan Processes and Zombie Processes

Orphan Process: The process whose parent gets terminated, while it is still running. In the case of Unix systems, this process is immediately adopted by the init process. Remember, the init process is the first process that runs on the Unix systems and has the PID 1.

Zombie Process: Process which has completed its execution with the help of exit() system call, but still has a entry in process table.