Performance Criteria

The performance criteria include the following:

1. **CPU Utilization**
   We want to keep the CPU as busy as possible. Conceptually, CPU utilization can range from 0% to 100%. In a real system, it should range from 40% (for a lightly loaded system) to 90% (for a heavily used system).
2. **Throughput**
   Throughput is the number of processes that are completed per time unit. For long processes, this rate may be one process per hour; for short transactions, it may be ten processes per second.
   $$Throughput = \frac{Number\ of\ processes\ completed}{Time\ Unit}$$
3. **Turnaround Time**
   The interval from the time of submission of a process to the time of completion is the turnaround time. Turnaround time is the sum of the periods spent waiting to get into memory, waiting in the ready queue, executing on the CPU, & doing I/O.
   $$Turnaround\ Time = Process\ End\ Time - Process\ Arrival\ Time$$
   OR
   $$Turnaround\ Time = Burst\ Time + Waiting\ Time$$
4. **Waiting Time**
   Waiting time is the sum of the periods spent waiting in the ready queue. The CPU scheduling algorithm affects only the amount of time that a process spends waiting in the ready queue.
   $$Waiting\ Time = Turanaround\ Time - Burst\ Time$$
   OR

5. **Response Time**
   Response time is the time from the submission of a request until the first response is produced. Response time is the time it takes to start responding, not the time it takes to output the response.
   $$Response\ Time = First\ Response\ Time - Process\ Arrival\ Time$$

We normally want to maximize the CPU utilization and Throughput, and Minimize the Turnaround Time, Waiting Time, and Response Time.

Process State

As a process executes, it changes state. The state of a process is defined in part by the current activity of that process. Each process may be in one of the following states:

- **New**: The process is being created.
- **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).
- **Ready**: The process is waiting to be assigned to a processor.
- **Terminated**: The process has finished execution.

These state names are arbitrary, and they vary across operating systems.

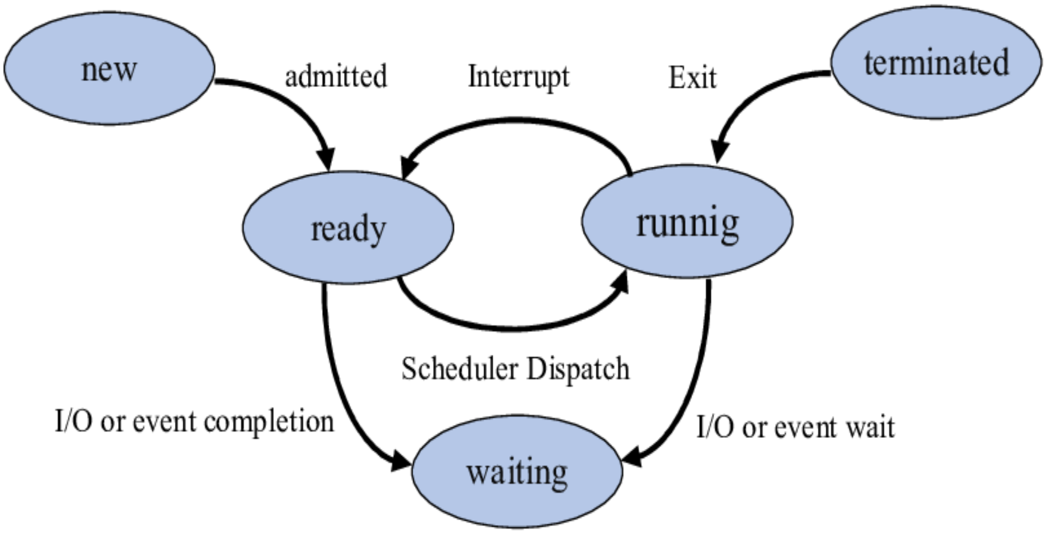

Figure: Process State Transition Diagram

New State: A process that just has been created but has not yet been admitted to the pool of execution processes by the operating system. Every new operation which is requested to the system is known as the new born process.

Ready State: When the process is ready to execute but he is waiting for the CPU to execute then this is called as the ready state. After completion of the input and output the process will be on ready state means the process will wait for the processor to execute.

Running State: The process that is currently being executed. When the process is running under the CPU, or when the program is executed by the CPU, then this is called as the running process and when a process is running then this will also provide us some outputs on the screen.

Waiting / Blocked State: A process that cannot execute until some event occurs or an I/O completion. When a process is waiting for some input and output operations then this is called as the waiting state and in this process is not under the execution instead the process is stored out of memory and when the user will provide the input and then this will again be on ready state.

Terminated State: After the completion of the process, the process will be automatically terminated by the CPU. So this is also called as the terminated state of the process. After executing the complete process the processor will also deallocate the memory which is allocated to the process. So this is called as the terminated process.

Note: Only one process can be running on any processor at any instant, although many processes may be ready and waiting.

Multi Programming

There are two types of multiprogramming:

- **Preemptive:** Process is forcefully removed from CPU.
- **Non Preemptive:** Processes are not removed until they complete the execution.

Degree of Multiprogramming

The number of processes that can reside in the ready state at maximum decides the degree of multiprogramming, e.g., if the degree of programming = 100, this means 100 processes can reside in the ready state at maximum.

Process Scheduling Queues

The OS maintains all PCBs in Process Scheduling Queues. The OS maintains a separate queue for each of the process states and PCBs of all processes in the same execution state are placed in the same queue. When the state of a process is changed, its PCB is unlinked from its current queue and moved to its new state queue.

The Operating System maintains the following important process scheduling queues −

- **Job Queue**: This queue keeps all the processes in the system.
- **Ready Queue:** This queue keeps a set of all processes residing in main memory, ready and waiting to execute. A new process is always put in this queue.
- **Device Queues**: The processes which are blocked due to unavailability of an I/O device constitute this queue.

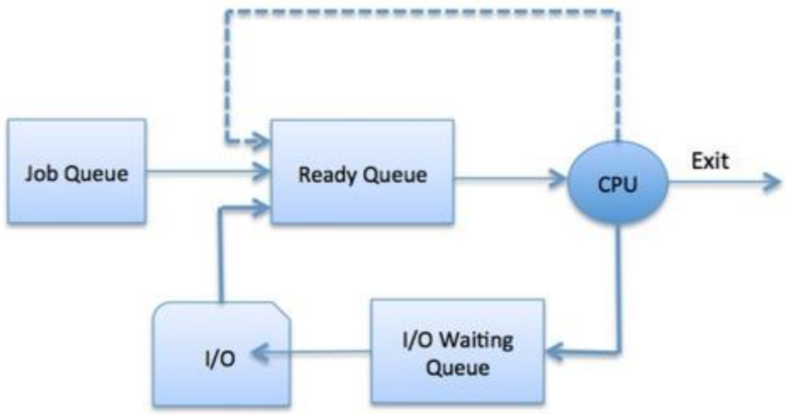

Figure: Process Scheduling Queues

Schedulers

A process migrates between the various scheduling queues through its life time. The operating system must select processes from these queues in some fashion. The selection process is carried out by the appropriate scheduler:

1. **The Long-Term Scheduler or Job Scheduler**
    - It selects which processes should be brought into the ready queue.
    - It controls the degree of multiprogramming.
    - Long term scheduler invoked very infrequently (in seconds, minutes) so it may be slow.
2. **The Short-Term Scheduler or CPU Scheduler**
    - It selects which process should be executed next and allocates CPU.
    - It must select a new process for the CPU frequently.
    - Short term scheduler invoked very frequently (in milliseconds), so it must be fast.
    - It also calls dispatcher (A dispatcher is software that moves process from ready state to running state and vice versa. In other words, it is context switching).
3. **The Medium-Term Scheduler**
    - It is responsible for Suspension decision.
    - It is used for swapping, that is moving the process from main memory to secondary and vice versa.

Comparison of Schedulers

Long-Term Scheduler	Short-Term Scheduler	Medium-Term Scheduler
It is a job scheduler	It is a CPU scheduler	It is a process swapping scheduler.
Speed is lesser than short term scheduler	Speed is fastest among other two	Speed is in between both short and long term scheduler.
It controls the degree of multiprogramming	It provides lesser control over degree of multiprogramming	It reduces the degree of multiprogramming.
It selects processes from pool and loads them into memory for execution	It selects those processes which are ready to execute	It can re-introduce the process into memory and execution can be continued.

Process Control Block

Each process is represented in the operating system by a **Process Control Block (PCB).** It is also called a **Task Control Block (TCB)**. A PCB is shown in Figure below. The PCB is identified by an integer process ID (PID). It contains many pieces of information associated with a specific process, including these:

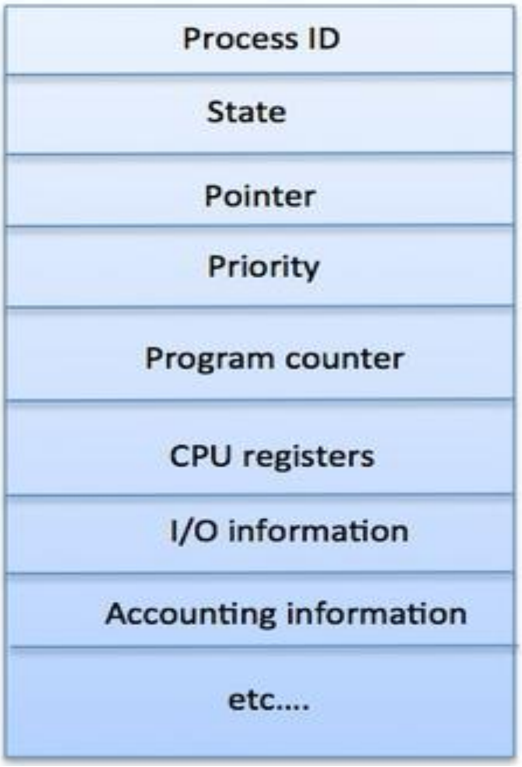

Figure: Process Control Block (PCB)

- **Process ID:** Unique identification for each of the process in the operating system.
- **Process State:** The current state of the process i.e., whether it is new, ready, running, waiting.
- **Pointer:** A pointer to parent process.
- **CPU Scheduling Information:** Process priority and other scheduling information which is required to schedule the process.
- **Program Counter:** Program Counter is a pointer to the address of the next instruction to be executed for this process.
- **CPU Registers:** Various CPU registers where process need to be stored for execution for running state.
- **IO Status Information:** This includes a list of I/O devices allocated to the process.
- **Accounting Information:** This includes the amount of CPU used for process execution, time limits, execution ID etc.
- **Process Privileges:** This is required to allow/disallow access to system resources.
- **Memory Management Information:** This includes the information of page table, memory limits, Segment table depending on memory used by the operating system.

The architecture of a PCB is completely dependent on Operating System and may contain different information in different operating systems. The PCB is maintained for a process throughout its lifetime, and is deleted once the process terminates.

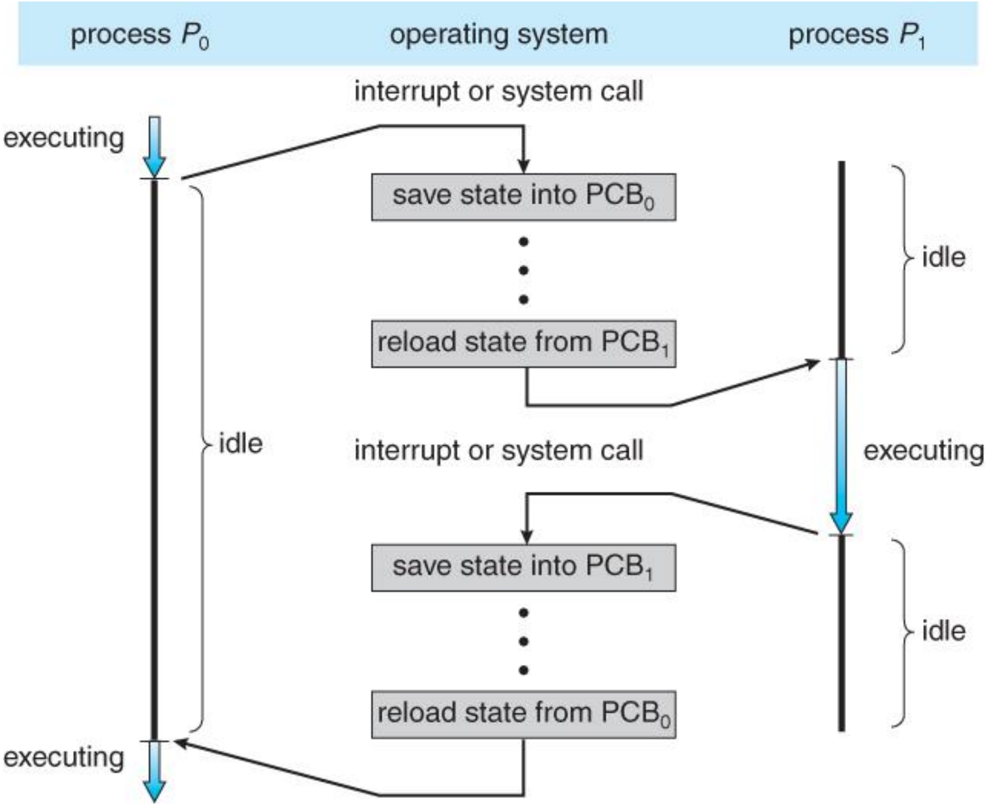

Figure: Diagram Showing CPU Switch From Process to Process.