# Process Scheduling

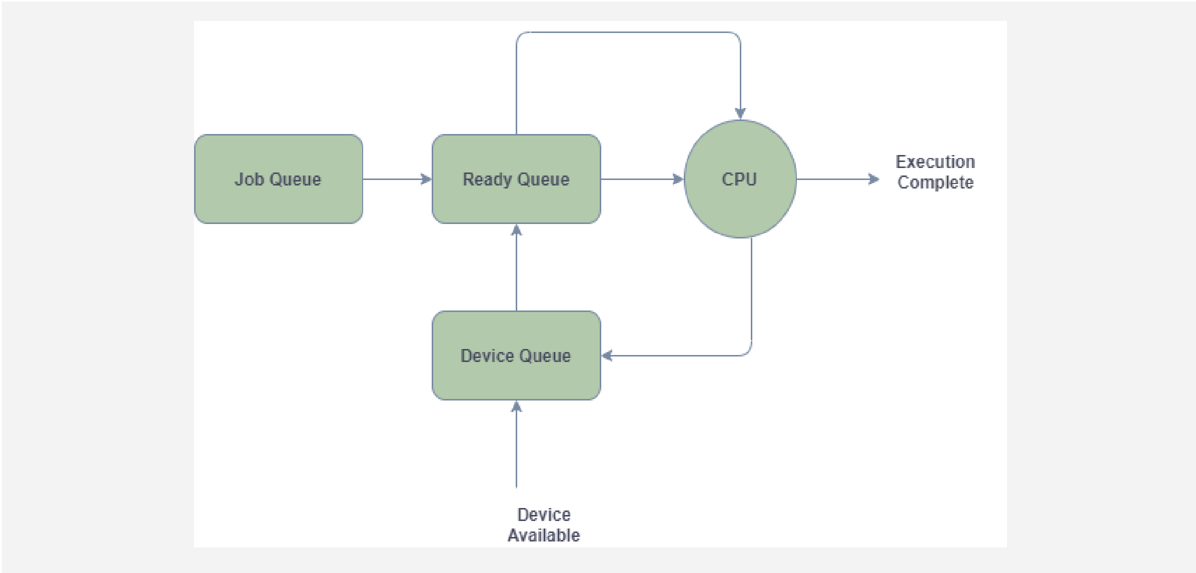
## Process Scheduling

Process scheduling is the job of the process manager that handles the removal of the currently running process from the CPU and the selection of another process on the basis of a particular approach. It is an essential part of a multiprogramming operating system. Such operating systems allow more than one process to be loaded into the executable memory at a time and the loaded process shares the CPU using time multiplexing. In multiprogramming systems, one process can use CPU while another is waiting for I/O. This is possible only with process scheduling.

## Scheduling Queue

Scheduling queues refer to queues of processes or devices. When the process enters 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 the device queue. A 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.
**Device queue:** The list of the processes waiting for a particular i/o device is called a device queue.



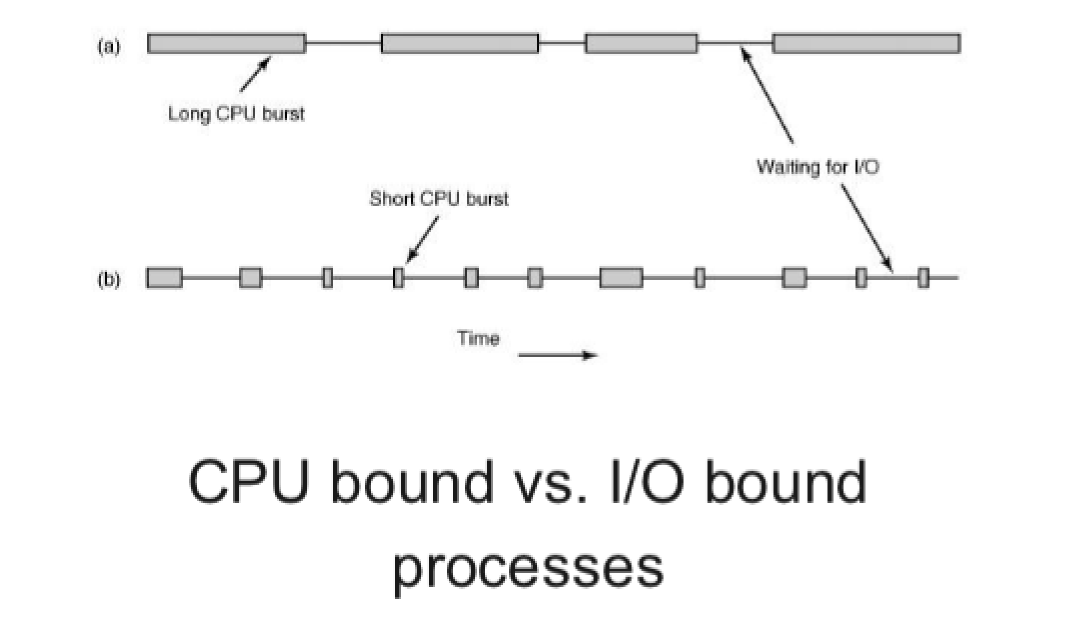

## CPU Bound Process

- CPU Bound processes are those that use algorithms with a large number of statistics.
- They can be expected to hold the CPU as long as the editor allows it.
- Programs such as simulations may be tied to the CPU for most of the process life.
- Users do not expect an immediate response from a computer when using CPU-bound programs.
- They should be given the most important scheduled schedule.

## I/O Bound Process

- The most awaiting processes for the completion of input or output (I / O) are I / O Bound.
- Interactive processes, such as office applications especially I / O are binding throughout the life of the process. Some processes may be bound by I / O in just a few short periods.
- The expected running time of bound I / O processes means that the process will not last very long.

- This should be provided very early by the editor.



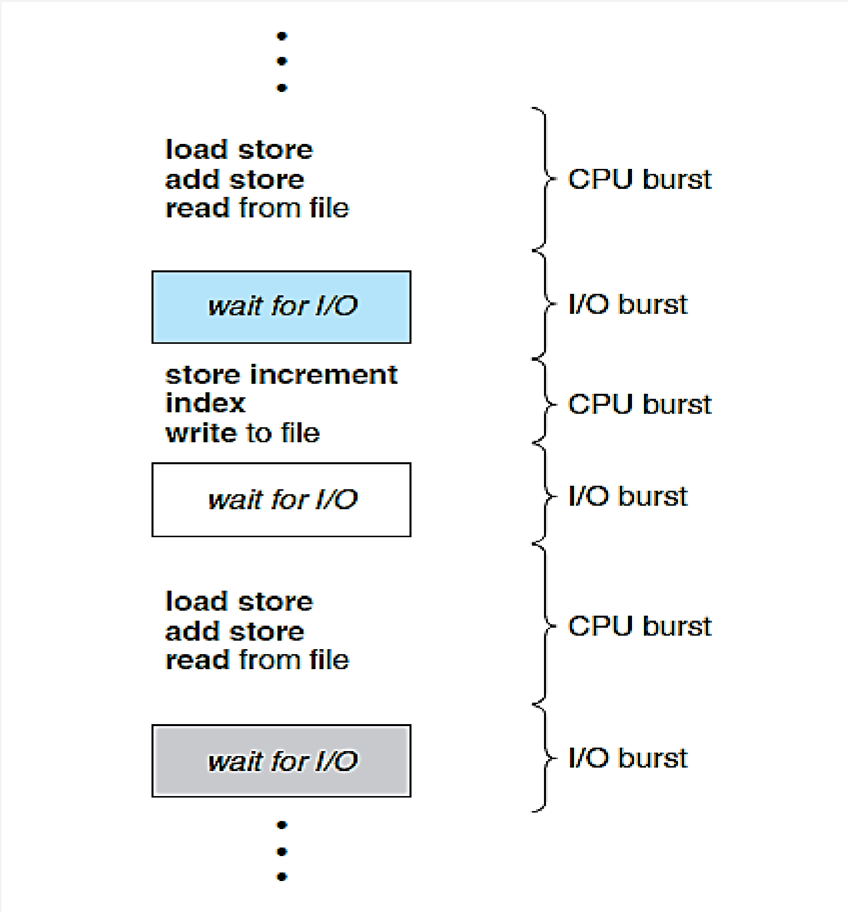

## Context Switching

- 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.

## CPU-I/O Burst Cycle

- Process execution consists of a cycle of CPU execution and I/O wait. Processes alternate between these two states.
- Process execution begins with a CPU burst. That is followed by an I/O burst, then another CPU burst, then another I/O burst, and so on.

- Eventually, the last CPU burst will end with a system request to terminate execution, rather than with another I/O burst.




**Alternate Sequence of CPU and I/O Burst**

## CPU Scheduler

When the CPU turns out to be idle then the operating system must select one of the essential processes in the ready queue to be executed. Short term scheduler is used for the selection process. There are several algorithms that are used for ready queue such as FIFO, LIFO priority queue, and so on. Generally, PCB is used to keep the record of queues.
CPU scheduling decisions may take place when a process:

1. Switches from running to waiting state
2. Switches from running to ready state
3. Switches from waiting to ready
4. Terminates

There are two types of scheduling:

**Preemptive scheduling**
In preemptive scheduling, a process can be forced to leave the CPU and switch to the ready queue. Example – Unix, Linux, Windows 95, and higher.

**Non-preemptive scheduling**
In non-preemptive scheduling or cooperating scheduling, a process keeps the CPU until it terminates or switches to the waiting state. Some machines support non-preemptive scheduling only. Example – Window 3.1x.

## Dispatcher

A dispatcher is the module of the operating system that gives control of the CPU to the process selected by the CPU scheduler.
Steps:

- switching context
- switching to user mode
- jumping to the proper location in the user program

**Dispatch latency:** time it takes for the dispatcher to stop one process and start another running.

## Scheduling Criteria

**CPU utilization:** keep the CPU as busy as possible
**Throughput:** No. of processes that complete their execution per time unit
**Turnaround time:** the amount of time to execute a particular process
**Waiting time:** the amount of time a process has been waiting in the ready queue

**Response time:** the amount of time it takes from when a request was submitted until the first response is produced, not output (for the time-sharing environment)