We'll discuss four major scheduling algorithms here which are following

* First Come First Serve (FCFS) Scheduling
* Shortest-Job-First (SJF) Scheduling
* Priority Scheduling
* Round Robin(RR) Scheduling

**Burst Time/ Execution Time/ Service Time**

Burst Time is actually time that is required to complete execution of particular task or process.

**Turnaround time**

Total time between submission of a process and its completion.

**Waiting Time**

The time the process remains in the ready queue.

**First Come First Serve (FCFS)**

* Jobs are executed on first come, first serve basis.
* Easy to understand and implement.
* Poor in performance as average wait time is high.
* By far the simplest CPU-scheduling algorithm is the first-come, first-served (FCFS) scheduling algorithm. With this algorithm, processes are assigned the CPU in the order they request it.
* Basically, there is a single queue of ready processes. Relative importance of jobs measured only by arrival time (poor choice).
* The implementation of the FCFS policy is easily managed with a FIFO queue. When a process enters the ready queue, its PCB is linked onto the tail of the queue.
* The average waiting time under the FCFS policy, however, is often quite long. Consider the following set of processes that arrive at time 0, with the length of the CPU burst given in milliseconds:

## Shortest Job First (SJF) (by default pre-emptive)

* Best approach to minimize waiting time.
* Impossible to implement
* Processer should know in advance how much time process will take.
* A different approach to CPU scheduling is the **shortest-job-first (SJF)** scheduling algorithm. This algorithm associates with each process the length of the process's next CPU burst.
* When the CPU is available, it is assigned to the process that has the smallest next CPU burst. If the next CPU bursts of two processes are the same, FCFS scheduling is used.
* The SJF algorithm can be either **pre-emptive** or **nonpreemptive**. The choice arises when a new process arrives at the ready queue while a previous process is still executing.
* The next CPU burst of the newly arrived process may be shorter than what is left of the currently executing process. A pre-emptive SJF algorithm will preempt the currently executing process, whereas a nonpreemptive SJF algorithm will allow the currently running process to finish its CPU burst.
* Pre-emptive SJF scheduling is sometimes called shortest-remaining-time-first scheduling.

## Priority Based Scheduling

* Each process is assigned a priority. Process with highest priority is to be executed first and so on.
* Processes with same priority are executed on first come first serve basis.
* Priority can be decided based on memory requirements, time requirements or any other resource requirement.

## Round Robin Scheduling

* **Round-robin** (RR) is one of the algorithms employed by [process](http://en.wikipedia.org/wiki/Process_scheduler) and [network schedulers](http://en.wikipedia.org/wiki/Network_scheduler) in [computing](http://en.wikipedia.org/wiki/Computing). As the term is generally used, [time slices](http://en.wikipedia.org/wiki/Preemption_(computing)#Time_slice) are assigned to each process in equal portions and in circular order, handling all processes without [priority](http://en.wiktionary.org/wiki/priority) (also known as [cyclic executive](http://en.wikipedia.org/wiki/Cyclic_executive)). Round-robin scheduling is simple, easy to implement, and [starvation](http://en.wikipedia.org/wiki/Resource_starvation)-free. Round-robin scheduling can also be applied to other scheduling problems, such as data packet scheduling in computer networks. It is an Operating System concept.
* Each process is provided a fix time to execute called quantum.
* Once a process is executed for given time period. Process is preempted and other process executes for given time period.
* Context switching is used to save states of preempted processes.