Introduction of Operating System – Set 1

Last Updated: 28-08-2019

An operating system acts as an intermediary between the user of a computer and computer hardware. The purpose of an operating system is to provide an environment in which a user can execute programs in a convenient and efficient manner.

An operating system is a software that manages the computer hardware. The hardware must provide appropriate mechanisms to ensure the correct operation of the computer system and to prevent user programs from interfering with the proper operation of the system.

**Operating System –** Definition:

* An operating system is a program that controls the execution of application programs and acts as an interface between the user of a computer and the computer hardware.
* A more common definition is that the operating system is the one program running at all times on the computer (usually called the kernel), with all else being application programs.
* An operating system is concerned with the allocation of resources and services, such as memory, processors, devices, and information. The operating system correspondingly includes programs to manage these resources, such as a traffic controller, a scheduler, memory management module, I/O programs, and a file system.

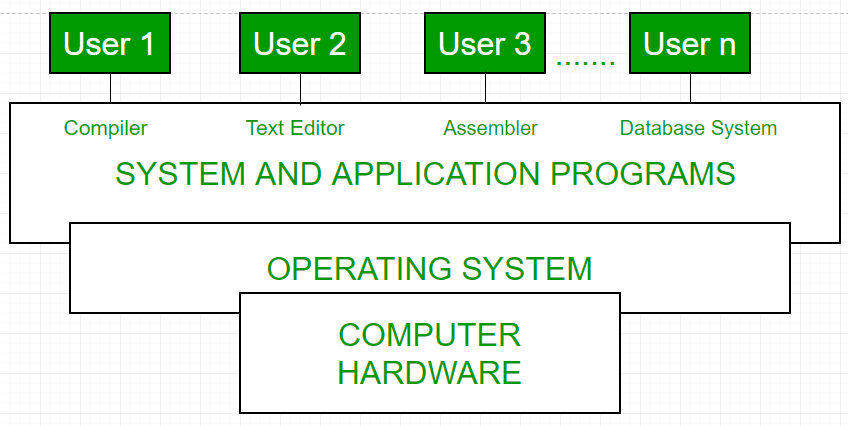
**Functions of Operating system –** Operating system performs three functions:

1. **Convenience:** An OS makes a computer more convenient to use.
2. **Efficiency:** An OS allows the computer system resources to be used in an efficient manner.
3. **Ability to Evolve:** An OS should be constructed in such a way as to permit the effective development, testing and introduction of new system functions at the same time without interfering with service.

Operating system as User Interface –

1. User
2. System and application programs
3. Operating system
4. Hardware

Every general-purpose computer consists of the hardware, operating system, system programs, and application programs. The hardware consists of memory, CPU, ALU, and I/O devices, peripheral device, and storage device. System program consists of compilers, loaders, editors, OS, etc. The application program consists of business programs, database programs.

  
Fig1: Conceptual view of a computer system

Every computer must have an operating system to run other programs. The operating system coordinates the use of the hardware among the various system programs and application programs for various users. It simply provides an environment within which other programs can do useful work.

The operating system is a set of special programs that run on a computer system that allows it to work properly. It performs basic tasks such as recognizing input from the keyboard, keeping track of files and directories on the disk, sending output to the display screen and controlling peripheral devices.  
OS is designed to serve two basic purposes:

1. It controls the allocation and use of the computing System’s resources among the various user and tasks.
2. It provides an interface between the computer hardware and the programmer that simplifies and makes feasible for coding, creation, debugging of application programs.

[**Types of Operating System**](https://www.geeksforgeeks.org/operating-system-types-operating-systems-awaiting-author/)**–**

* Batch Operating System- Sequence of jobs in a program on a computer without manual interventions.
* Time sharing operating System- allows many users to share the computer resources.(Max utilization of the resources).
* Distributed operating System- Manages a group of different computers and make appear to be a single computer.
* Network operating system- computers running in different operating system can participate in common network (It is used for security purpose).
* Real time operating system – meant applications to fix the deadlines.

# Difference between Multiprogramming, multitasking, multithreading and multiprocessing

Last Updated: 16-09-2019

1. **Multiprogramming –** A computer running more than one program at a time (like running Excel and Firefox simultaneously).
2. **Multiprocessing –** A computer using more than one CPU at a time.
3. **Multitasking –** Tasks sharing a common resource (like 1 CPU).
4. **Multithreading** is an extension of multitasking.

### ****1. Multi programming –****

In a modern computing system, there are usually several concurrent application processes which want to execute. Now it is the responsibility of the Operating System to manage all the processes effectively and efficiently.  
One of the most important aspects of an Operating System is to multi program.  
In a computer system, there are multiple processes waiting to be executed, i.e. they are waiting when the CPU will be allocated to them and they begin their execution. These processes are also known as jobs. Now the main memory is too small to accommodate all of these processes or jobs into it. Thus, these processes are initially kept in an area called job pool. This job pool consists of all those processes awaiting allocation of main memory and CPU.  
CPU selects one job out of all these waiting jobs, brings it from the job pool to main memory and starts executing it. The processor executes one job until it is interrupted by some external factor or it goes for an I/O task.

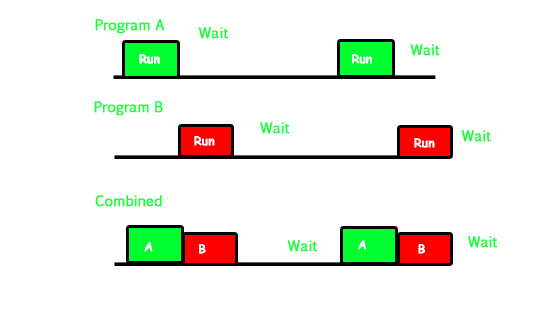
**Non-multi programmed system’s working –**

* In a non multi programmed system, As soon as one job leaves the CPU and goes for some other task (say I/O ), the CPU becomes idle. The CPU keeps waiting and waiting until this job (which was executing earlier) comes back and resumes its execution with the CPU. So CPU remains free for all this while.
* Now it has a drawback that the CPU remains idle for a very long period of time. Also, other jobs which are waiting to be executed might not get a chance to execute because the CPU is still allocated to the earlier job.  
  This poses a very serious problem that even though other jobs are ready to execute, CPU is not allocated to them as the CPU is allocated to a job which is not even utilizing it (as it is busy in I/O tasks).
* It cannot happen that one job is using the CPU for say 1 hour while the others have been waiting in the queue for 5 hours. To avoid situations like this and come up with efficient utilization of CPU, the concept of multi programming came up.

The main idea of multi programming is to maximize the CPU time.  
**Multi programmed system’s working –**

* In a multi-programmed system, as soon as one job goes for an I/O task, the Operating System interrupts that job, chooses another job from the job pool (waiting queue), gives CPU to this new job and starts its execution. The previous job keeps doing its I/O operation while this new job does CPU bound tasks. Now say the second job also goes for an I/O task, the CPU chooses a third job and starts executing it. As soon as a job completes its I/O operation and comes back for CPU tasks, the CPU is allocated to it.
* In this way, no CPU time is wasted by the system waiting for the I/O task to be completed.  
  Therefore, the ultimate goal of multi programming is to keep the CPU busy as long as there are processes ready to execute. This way, multiple programs can be executed on a single processor by executing a part of a program at one time, a part of another program after this, then a part of another program and so on, hence executing multiple programs. Hence, the CPU never remains idle.

In the image below, program A runs for some time and then goes to waiting state. In the mean time program B begins its execution. So the CPU does not waste its resources and gives program B an opportunity to run.



### ****2. Multiprocessing –****

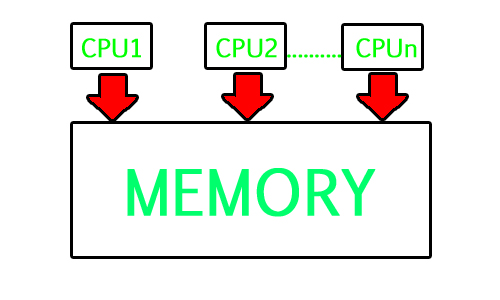
In a uni-processor system, only one process executes at a time.  
Multiprocessing is the use of two or more CPUs (processors) within a single Computer system. The term also refers to the ability of a system to support more than one processor within a single computer system. Now since there are multiple processors available, multiple processes can be executed at a time. These multi processors share the computer bus, sometimes the clock, memory and peripheral devices also.

**Multi processing system’s working –**

* With the help of multiprocessing, many processes can be executed simultaneously. Say processes P1, P2, P3 and P4 are waiting for execution. Now in a single processor system, firstly one process will execute, then the other, then the other and so on.
* But with multiprocessing, each process can be assigned to a different processor for its execution. If its a dual-core processor (2 processors), two processes can be executed simultaneously and thus will be two times faster, similarly a quad core processor will be four times as fast as a single processor.

**Why use multi processing –**

* The main advantage of multiprocessor system is to get more work done in a shorter period of time. These types of systems are used when very high speed is required to process a large volume of data. Multi processing systems can save money in comparison to single processor systems because the processors can share peripherals and power supplies.
* It also provides increased reliability in the sense that if one processor fails, the work does not halt, it only slows down. e.g. if we have 10 processors and 1 fails, then the work does not halt, rather the remaining 9 processors can share the work of the 10th processor. Thus the whole system runs only 10 percent slower, rather than failing altogether.



Multiprocessing refers to the hardware (i.e., the CPU units) rather than the software (i.e., running processes). If the underlying hardware provides more than one processor then that is multiprocessing. It is the ability of the system to leverage multiple processors’ computing power.

**Difference between Multi programming and Multi processing –**

* A System can be both multi programmed by having multiple programs running at the same time and multiprocessing by having more than one physical processor. The difference between multiprocessing and multi programming is that Multiprocessing is basically executing multiple processes at the same time on multiple processors, whereas multi programming is keeping several programs in main memory and executing them concurrently using a single CPU only.
* Multiprocessing occurs by means of parallel processing whereas Multi programming occurs by switching from one process to other (phenomenon called as context switching).

### ****3. Multitasking –****

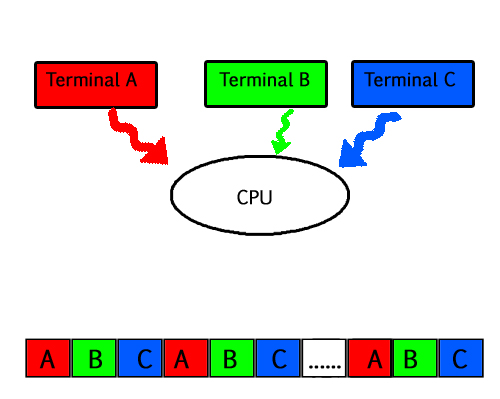
As the name itself suggests, multi tasking refers to execution of multiple tasks (say processes, programs, threads etc.) at a time. In the modern operating systems, we are able to play MP3 music, edit documents in Microsoft Word, surf the Google Chrome all simultaneously, this is accomplished by means of multi tasking.

Multitasking is a logical extension of multi programming. The major way in which multitasking differs from multi programming is that multi programming works solely on the concept of context switching whereas multitasking is based on time sharing alongside the concept of context switching.

**Multi tasking system’s working –**

* In a time sharing system, each process is assigned some specific quantum of time for which a process is meant to execute. Say there are 4 processes P1, P2, P3, P4 ready to execute. So each of them are assigned some time quantum for which they will execute e.g time quantum of 5 nanoseconds (5 ns). As one process begins execution (say P2), it executes for that quantum of time (5 ns). After 5 ns the CPU starts the execution of the other process (say P3) for the specified quantum of time.
* Thus the CPU makes the processes to share time slices between them and execute accordingly. As soon as time quantum of one process expires, another process begins its execution.
* Here also basically a context switch is occurring but it is occurring so fast that the user is able to interact with each program separately while it is running. This way, the user is given the illusion that multiple processes/ tasks are executing simultaneously. But actually only one process/ task is executing at a particular instant of time. In multitasking, time sharing is best manifested because each running process takes only a fair quantum of the CPU time.

In a more general sense, multitasking refers to having multiple programs, processes, tasks, threads running at the same time. This term is used in modern operating systems when multiple tasks share a common processing resource (e.g., CPU and Memory).



* As depicted in the above image, At any time the CPU is executing only one task while other tasks are waiting for their turn. The illusion of parallelism is achieved when the CPU is reassigned to another task. i.e all the three tasks A, B and C are appearing to occur simultaneously because of time sharing.
* So for multitasking to take place, firstly there should be multiprogramming i.e. presence of multiple programs ready for execution. And secondly the concept of time sharing.

### ****4. Multi threading –****

A thread is a basic unit of CPU utilization. Multi threading is an execution model that allows a single process to have multiple code segments (i.e., threads) running concurrently within the “context” of that process.  
e.g. VLC media player, where one thread is used for opening the VLC media player, one thread for playing a particular song and another thread for adding new songs to the playlist.

Multi threading is the ability of a process to manage its use by more than one user at a time and to manage multiple requests by the same user without having to have multiple copies of the program.

**Multi threading system’s working –**

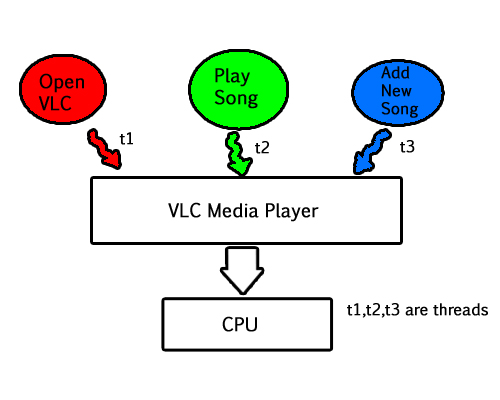
**Example 1 –**

* Say there is a web server which processes client requests. Now if it executes as a single threaded process, then it will not be able to process multiple requests at a time. Firstly one client will make its request and finish its execution and only then, the server will be able to process another client request. This is really costly, time consuming and tiring task. To avoid this, multi threading can be made use of.
* Now, whenever a new client request comes in, the web server simply creates a new thread for processing this request and resumes its execution to hear more client requests. So the web server has the task of listening to new client requests and creating threads for each individual request. Each newly created thread processes one client request, thus reducing the burden on web server.

**Example 2 –**

* We can think of threads as child processes that share the parent process resources but execute independently. Now take the case of a GUI. Say we are performing a calculation on the GUI (which is taking very long time to finish). Now we can not interact with the rest of the GUI until this command finishes its execution. To be able to interact with the rest of the GUI, this command of calculation should be assigned to a separate thread. So at this point of time, 2 threads will be executing i.e. one for calculation, and one for the rest of the GUI. Hence here in a single process, we used multiple threads for multiple functionality.

The image below completely describes the VLC player example:



**Advantages of Multi threading –**

* Benefits of Multi threading include increased responsiveness. Since there are multiple threads in a program, so if one thread is taking too long to execute or if it gets blocked, the rest of the threads keep executing without any problem. Thus the whole program remains responsive to the user by means of remaining threads.
* Another advantage of multi threading is that it is less costly. Creating brand new processes and allocating resources is a time consuming task, but since threads share resources of the parent process, creating threads and switching between them is comparatively easy. Hence multi threading is the need of modern Operating Systems.

A CPU-bound process is a process that spends most of his time executing instructions.

An I/O-bound process is one which spends most of his time waiting for I/O operations to complete.It is important for a scheduler to distinguish the two because this allows the scheduler to keep a balanced system. If the scheduler only executes processes that are I/O-bound, then the CPU is underutilized. On the contrary, if the scheduler only executes processes that are CPU-bound then a high CPU utilization is achieved, but other system resources (such as the hard drive or network interface) may be underused. A good balance of CPU-bound and I/O-bound processes keeps the overall system as busy as possible.

**I/O-bound** programs have the property of performing only a small amount of computation before performing I/O. Such programs typically do not use up their entire CPU quantum.

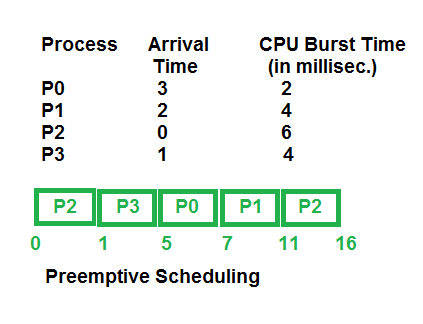
**CPU-bound programs**, on the other hand, use their entire quantum without performing any blocking I/O operations. Consequently, one could make better use of the computer’s resources by giving higher priority to I/O-bound programs and allow them to execute ahead of the CPU-bound programs.

Preemptive and Non-Preemptive Scheduling

Last Updated: 02-04-2020

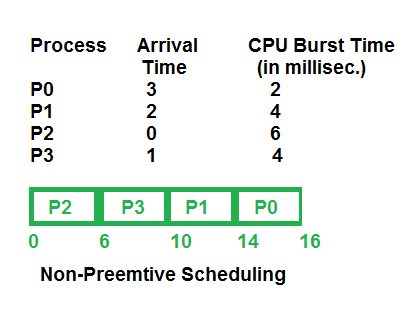
Prerequisite – [CPU Scheduling](https://www.geeksforgeeks.org/gate-notes-operating-system-process-scheduling/)  
**1. Preemptive Scheduling:**  
Preemptive scheduling is used when a process switches from running state to ready state or from waiting state to ready state. The resources (mainly CPU cycles) are allocated to the process for the limited amount of time and then is taken away, and the process is again placed back in the ready queue if that process still has CPU burst time remaining. That process stays in ready queue till it gets next chance to execute.

Algorithms based on preemptive scheduling are: [Round Robin (RR)](https://www.geeksforgeeks.org/program-round-robin-scheduling-set-1/),[Shortest Remaining Time First (SRTF)](https://www.geeksforgeeks.org/program-shortest-job-first-scheduling-set-2srtf-make-changesdoneplease-review/" \t "_blank), [Priority (preemptive version)](https://www.geeksforgeeks.org/program-for-preemptive-priority-cpu-scheduling/), etc.



**2. Non-Preemptive Scheduling:**  
Non-preemptive Scheduling is used when a process terminates, or a process switches from running to waiting state. In this scheduling, once the resources (CPU cycles) is allocated to a process, the process holds the CPU till it gets terminated or it reaches a waiting state. In case of non-preemptive scheduling does not interrupt a process running CPU in middle of the execution. Instead, it waits till the process complete its CPU burst time and then it can allocate the CPU to another process.

Algorithms based on non-preemptive scheduling are:[Shortest Job First (SJF basically non preemptive)](https://www.geeksforgeeks.org/program-shortest-job-first-sjf-scheduling-set-1-non-preemptive/) and [Priority (non preemptive version)](https://www.geeksforgeeks.org/operating-system-priority-scheduling-different-arrival-time-set-2/), etc.



**Key Differences Between Preemptive and Non-Preemptive Scheduling:**

1. In preemptive scheduling the CPU is allocated to the processes for the limited time whereas in Non-preemptive scheduling, the CPU is allocated to the process till it terminates or switches to waiting state.
2. The executing process in preemptive scheduling is interrupted in the middle of execution when higher priority one comes whereas, the executing process in non-preemptive scheduling is not interrupted in the middle of execution and wait till its execution.
3. In Preemptive Scheduling, there is the overhead of switching the process from ready state to running state, vise-verse, and maintaining the ready queue. Whereas in case of non-preemptive scheduling has no overhead of switching the process from running state to ready state.
4. In preemptive scheduling, if a high priority process frequently arrives in the ready queue then the process with low priority has to wait for a long, and it may have to starve. On the other hands, in the non-preemptive scheduling, if CPU is allocated to the process having larger burst time then the processes with small burst time may have to starve.
5. Preemptive scheduling attain flexible by allowing the critical processes to access CPU as they arrive into the ready queue, no matter what process is executing currently. Non-preemptive scheduling is called rigid as even if a critical process enters the ready queue the process running CPU is not disturbed.
6. The Preemptive Scheduling has to maintain the integrity of shared data that’s why it is cost associative as it which is not the case with Non-preemptive Scheduling.

# CPU Scheduling Criteria

Last Updated: 16-06-2020

Different [CPU scheduling algorithms](https://www.geeksforgeeks.org/cpu-scheduling-in-operating-systems/) have different properties and choice of a particular algorithm depends on the various factors. Many criteria have been suggested for comparing CPU scheduling algorithms.

The criteria include the following:

1. **CPU utilisation –**  
   The main objective of any CPU scheduling algorithm is to keep the CPU as busy as possible. Theoretically CPU utilisation can range from 0 to 100 but in a real time system it varies from 40 to 90 percent depending on the load upon the system.
2. **Throughout –**  
   A measure of the work done by CPU is the number of processes being executed and completed per unit time. This is called throughput. The throughput may vary depending upon the length or duration of processes.
3. **Turnaround time –**  
   For a particular process, an important criteria is how long it takes to execute that process. The time elapsed from the time of submission of a process to the time of completion is known as turnaround time. Turn-around time is the sum of times spent waiting to get into memory, waiting in ready queue, executing in CPU and waiting for I/O.
4. **Waiting time –**  
   A scheduling algorithm does not affect the time required to complete the process once it starts execution. It only affects the waiting time of a process i.e. time spent by a process waiting in the ready queue.
5. **Response time –**  
   In an interactive system turn-around time is not the best criteria. A process may produce some output fairly early and continue computing new results while previous results are being output to user. Thus another criteria is the time taken from submission of process of request until the first response is produced. This measure is called response time.

CPU Scheduling in Operating Systems

Last Updated: 28-06-2020

Scheduling of processes/work is done to finish the work on time.

Below are different time with respect to a process.

***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*

**Why do we need scheduling?**  
A typical process involves both I/O time and CPU time. In a uni programming system like MS-DOS, time spent waiting for I/O is wasted and CPU is free during this time. In multi programming systems, one process can use CPU while another is waiting for I/O. This is possible only with process scheduling.

**Objectives of Process Scheduling Algorithm**

*Max CPU utilization [Keep CPU as busy as possible]  
Fair allocation of CPU.  
Max throughput [Number of processes that complete their execution per time unit]  
Min turnaround time [Time taken by a process to finish execution]  
Min waiting time [Time a process waits in ready queue]  
Min response time [Time when a process produces first response]*

## What is First Come First Serve Method?

**First Come First Serve (FCFS)** is an operating system scheduling algorithm that automatically executes queued requests and processes in order of their arrival. It is the easiest and simplest CPU scheduling algorithm. In this type of algorithm, processes which requests the CPU first get the CPU allocation first. This is managed with a FIFO queue. The full form of FCFS is First Come First Serve.

As the process enters the ready queue, its PCB (Process Control Block) is linked with the tail of the queue and, when the CPU becomes free, it should be assigned to the process at the beginning of the queue.

## Characteristics of FCFS method

* It supports non-preemptive and pre-emptive scheduling algorithm.
* Jobs are always executed on a first-come, first-serve basis.
* It is easy to implement and use.
* This method is poor in performance, and the general wait time is quite high.

## Advantages of FCFS

Here, are pros/benefits of using FCFS scheduling algorithm:

* The simplest form of a CPU scheduling algorithm
* Easy to program
* First come first served

## Disadvantages of FCFS

Here, are cons/ drawbacks of using FCFS scheduling algorithm:

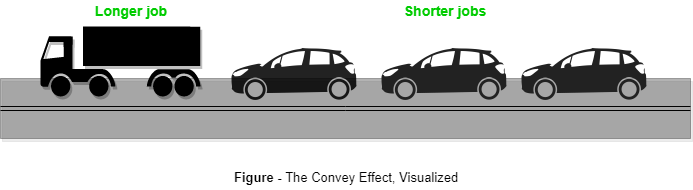
* It is a Non-Preemptive CPU scheduling algorithm, so after the process has been allocated to the CPU, it will never release the CPU until it finishes executing.
* The Average Waiting Time is high.
* Short processes that are at the back of the queue have to wait for the long process at the front to finish.
* Not an ideal technique for time-sharing systems.
* Because of its simplicity, FCFS is not very efficient.

# Convoy Effect in Operating Systems

Last Updated: 12-02-2019

**Prerequisites :** Basics of FCFS Scheduling ([Program for FCFS Scheduling | Set 1](https://www.geeksforgeeks.org/program-fcfs-scheduling-set-1/), [Program for FCFS Scheduling | Set 2](https://www.geeksforgeeks.org/program-fcfs-scheduling-set-2-processes-different-arrival-time/))

Convoy Effect is phenomenon associated with the First Come First Serve (FCFS) algorithm, in which the whole Operating System slows down due to few slow processes.



FCFS algorithm is non-preemptive in nature, that is, once CPU time has been allocated to a process, other processes can get CPU time only after the current process has finished. This property of FCFS scheduling leads to the situation called Convoy Effect.

Suppose there is one CPU intensive (large burst time) process in the ready queue, and several other processes with relatively less burst times but are Input/Output (I/O) bound (Need I/O operations frequently).