National University of Computer &

Emerging Sciences Karachi Campus

****

**Operating System**

**Project Report**

**COMPARISION OF PROCESS SCHEDULING ALGORITHMS**

**Section C**

**Group Members:**

Aahil Halani 20k-0308

Anas Ali 20k-0352

Muhammad Rafay 20k-0180

**Introduction:**

To keep all of the processes that come with a processor running in parallel, task scheduling is required. Not every algorithm is better at solving the significant problem in all circumstances. In short bursts of time, the FCFS algorithm is better than the others, whereas Round Robin is better for multiple processes all of the time. It is, however, impossible to predict, the next process. Average Waiting Time (T avg) is a standard metric for assessing the scheduling algorithm's performance. Several techniques have been used to keep the process running and restore normal CPU performance. The goal is to figure out which algorithm is best for a particular situation.

**Working Criteria:**

The working of this project would be based on comparing all the algorithms on the basis of testing and analysis. The Arrival time, Burst time, waiting time and Avg times will be compared. Each algorithm will be tested according to worst and best process scheduling possible.

**Tools and Technologies:**

Programming Language: C++ language

Platform: Dev C++

**Algorithms:**

1. First Come First Serve
2. Shortest Job First(Non-Preemptive)
3. Shortest Time Remaining First(Preemptive version of SJF)
4. Priority(Preemptive and Non-preemptive)
5. Round Robin

Below listed are the pros and cons of each algorithm with its main features

First Come First Serve (FCFS)

**Advantages:**

* [FCFS algorithm](https://www.studytonight.com/operating-system/first-come-first-serve) doesn't include any complex logic, it just puts the process requests in a queue and executes it one by one.
* Hence, FCFS is pretty simple and easy to implement.
* Eventually, every process will get a chance to run, so starvation doesn't occur.

**Disadvantages:**

* There is no option for pre-emption of a process. If a process is started, then CPU executes the process until it ends.
* Because there is no pre-emption, if a process executes for a long time, the processes in the back of the queue will have to wait for a long time before they get a chance to be executed.

Shortest Job First (SJF)

**Advantages:**

* According to the definition, short processes are executed first and then followed by longer processes.
* The throughput is increased because more processes can be executed in less amount of time.

**Disadvantages:**

* The time taken by a process must be known by the CPU beforehand, which is not possible.
* Longer processes will have more waiting time, eventually they'll suffer starvation.

**Note:** Preemptive Shortest Job First scheduling will have the same advantages and disadvantages as those for SJF.

Round Robin (RR)

**Advantages:**

Each process is served by the CPU for a fixed time quantum, so all processes are given the same priority.

* Starvation doesn't occur because for each round robin cycle, every process is given a fixed time to execute. No process is left behind.

**Disadvantages:**

* The throughput in RR largely depends on the choice of the length of the time quantum. If time quantum is longer than needed, it tends to exhibit the same behavior as FCFS.
* If time quantum is shorter than needed, the number of times that CPU switches from one process to another process, increases. This leads to decrease in CPU efficiency.

Priority based Scheduling

**Advantages**:

The priority of a process can be selected based on memory requirement, time requirement or user preference. For example, a high end game will have better graphics, that means the process which updates the screen in a game will have higher priority so as to achieve better graphics performance.

**Disadvantages:**

* A second scheduling algorithm is required to schedule the processes which have same priority.
* In preemptive priority scheduling, a higher priority process can execute ahead of an already executing lower priority process. If lower priority process keeps waiting for higher priority processes, starvation occurs.

Situations and Circumstances when a particular algorithm will be best suited for the job

Usage of Scheduling Algorithms in Different Situations

Every scheduling algorithm has a type of a situation where it is the best choice. Let's look at different such situations:

**Situation 1:**

The incoming processes are short and there is no need for the processes to execute in a specific order.

In this case, FCFS works best when compared to SJF and RR because the processes are short which means that no process will wait for a longer time. When each process is executed one by one, every process will be executed eventually.

**Situation 2:**

The processes are a mix of long and short processes and the task will only be completed if all the processes are executed successfully in a given time.

Round Robin scheduling works efficiently here because it does not cause starvation and also gives equal time quantum for each process.

**Situation 3:**

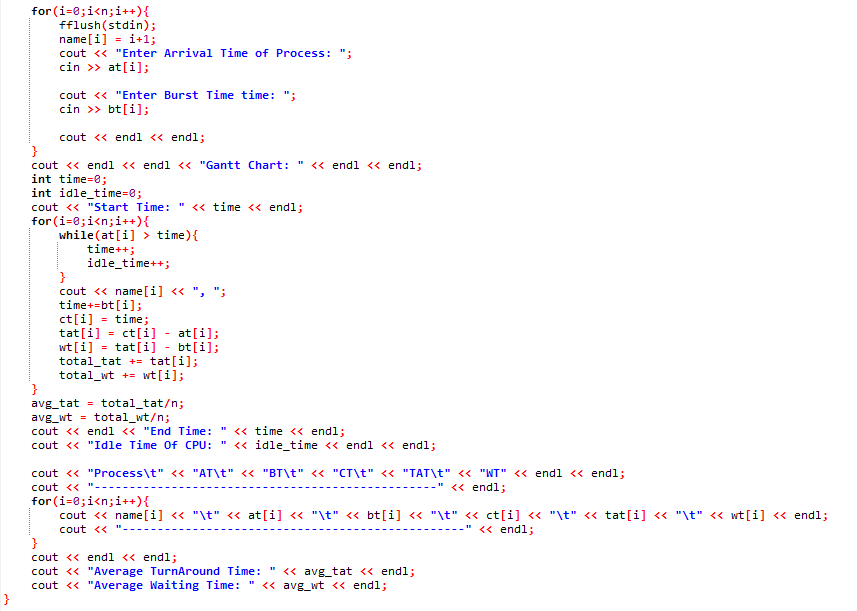
The processes are a mix of user based and kernel based processes.

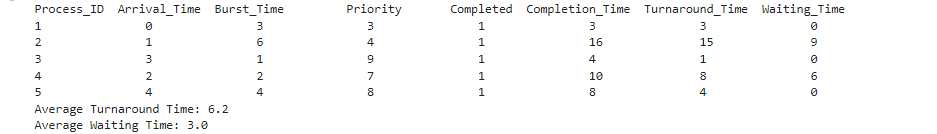
Priority based scheduling works efficiently in this case because generally kernel based processes have higher priority when compared to user based processes.

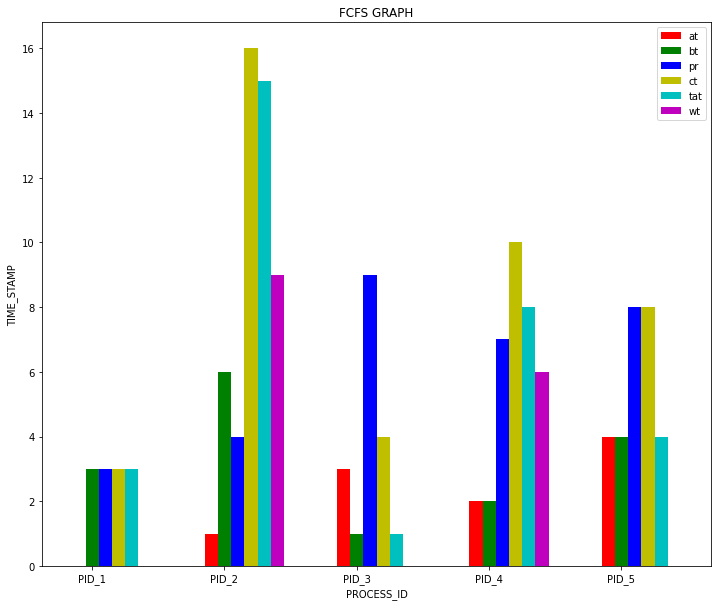
For example, the scheduler itself is a kernel based process, it should run first so that it can schedule other processes.

**Code Snippets:**

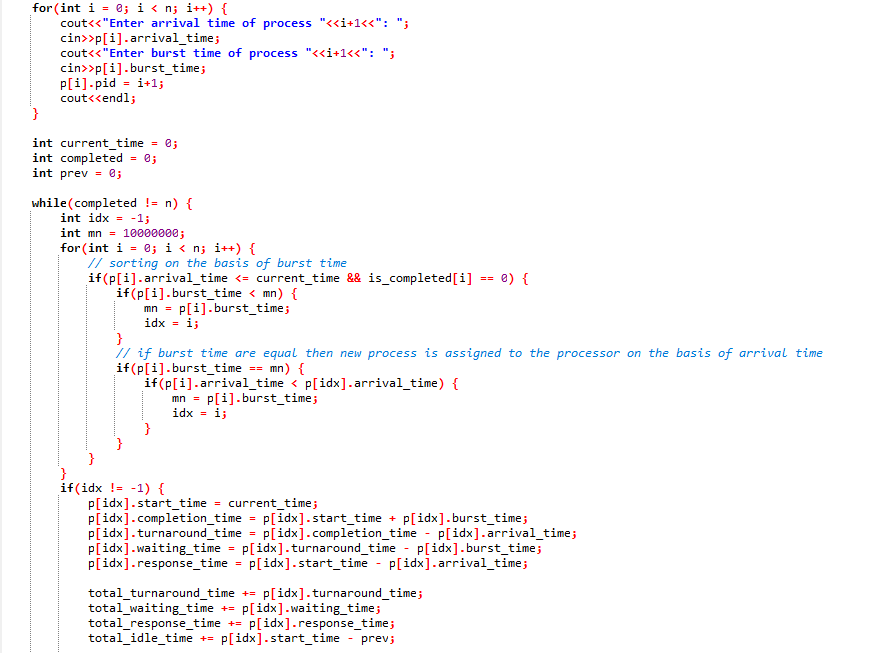
**FCFS**



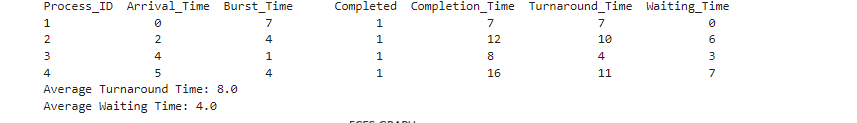


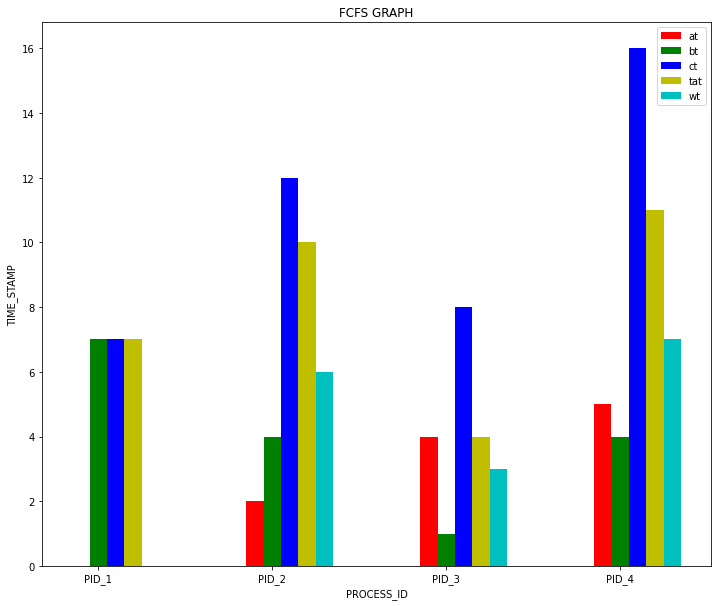


SJF

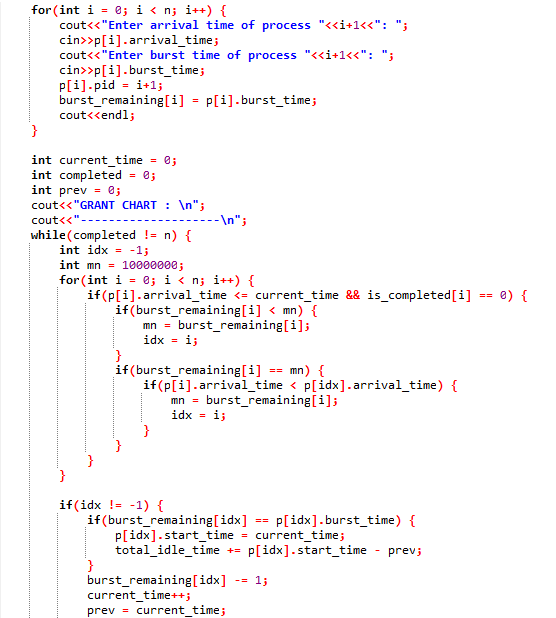




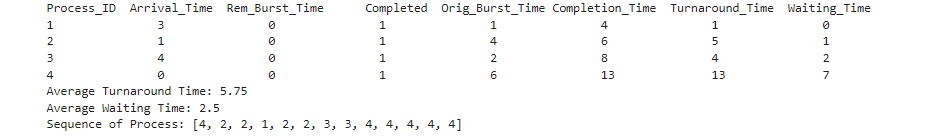


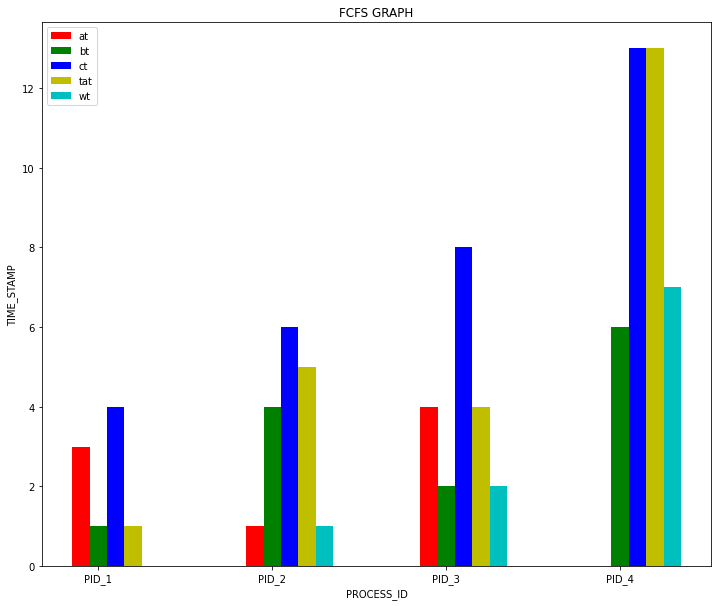


SRTF

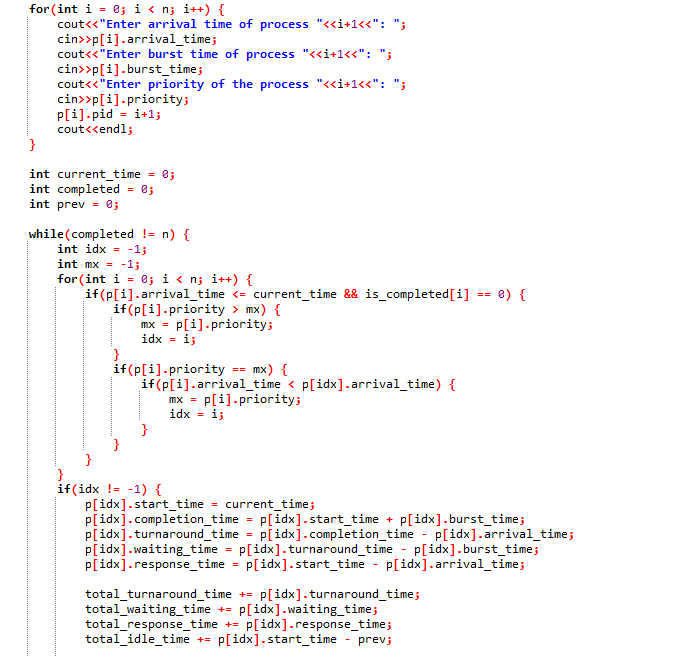




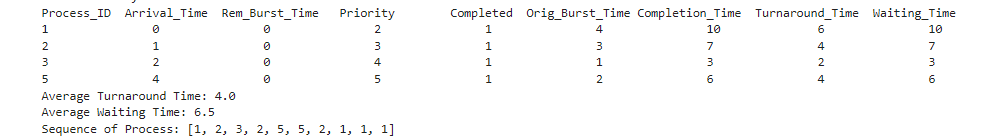


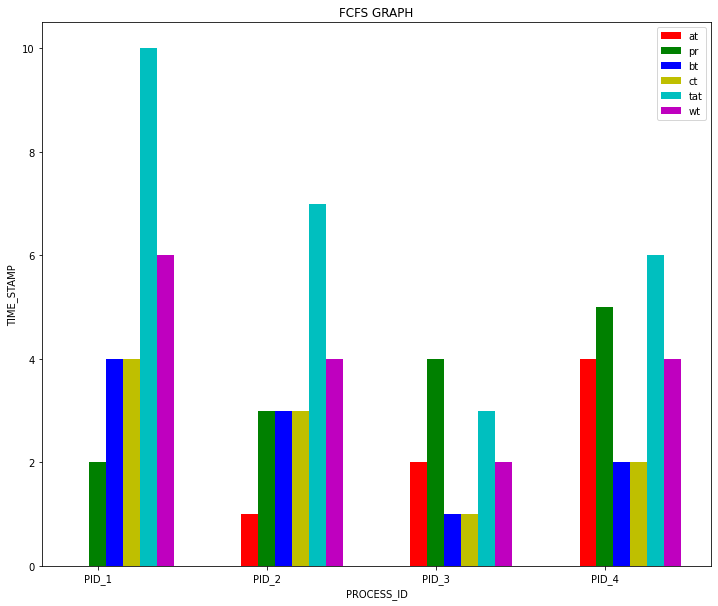


Priority

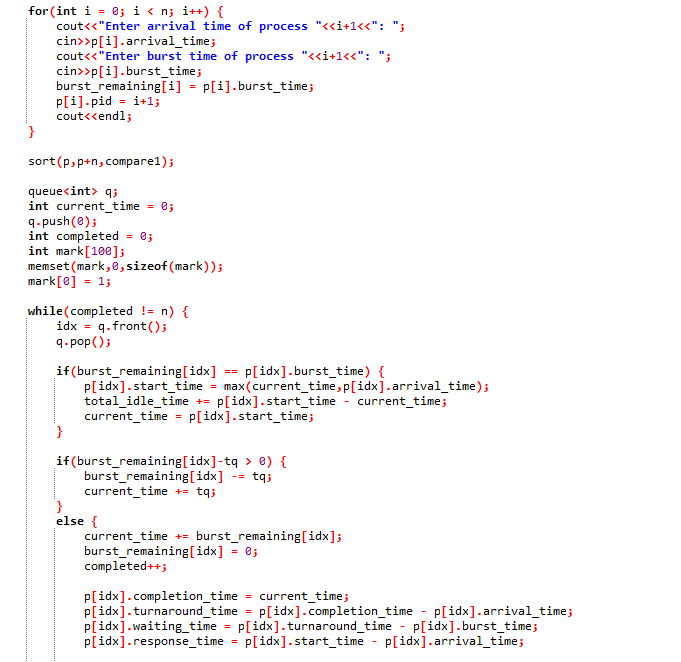


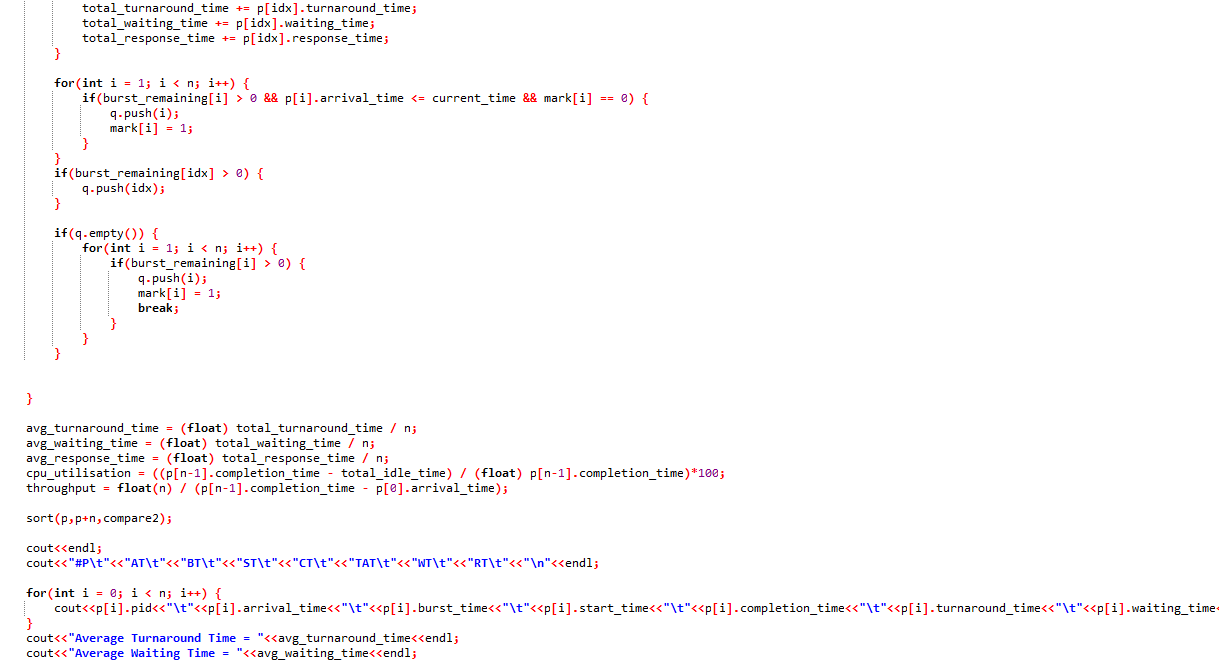


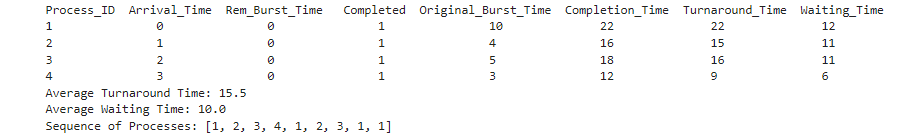


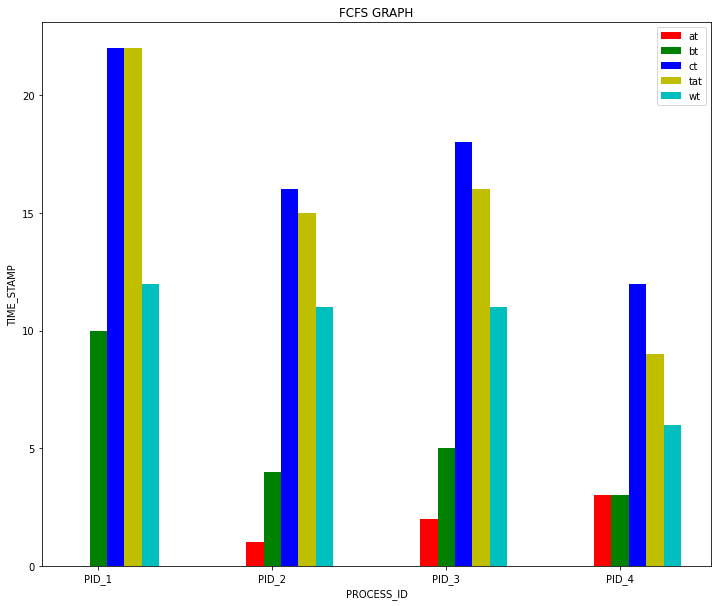


Round Robin:









**Conclusion**

It is clearly observed that turnaround time, waiting time and response time of the processes are optimum for SJF scheduling algorithm compared to all other fundamental algorithms from Figure II, Figure. III, Fig. IV, Fig. V, (Figure II to Figure V are Gantt Chart) Figure VI and Figure VII. It can also be observed that throughput and CPU utilization rate are optimum. From above analysis and discussion, we can say that the FCFS is simple to understand and suitable only for batch system where waiting time is large. SJF scheduling algorithm gives minimum average waiting time and average turnaround time. The priority scheduling algorithm is based on the priority in which the highest priority job can run first and the lowest priority job need to wait though it will create a problem of starvation.

The SJF scheduling algorithm is to serve all types of job with optimum scheduling criteria. The treatment of shortest process in SJF scheduling tends to result in increased waiting time for long processes. And the long process will never get served, though it produces minimum average waiting time and average turnaround time. The shortest job first scheduling algorithm deals with different approach, in this algorithm the major benefit is it gives the minimum average waiting time. It is recommended that any kind of simulation for any CPU scheduling algorithm has limited accuracy. The only way to evaluate a scheduling algorithm to code it and has to put it in the operating system, only then a proper working capability of the algorithm can be measured in real time systems.

**References:**

* https://en.wikipedia.org/wiki/Shortest\_job\_next
* <https://www.researchgate.net/publication/273260942_Comparative_Analysis_of_Various_Schedul>ing\_Algorithms
* Operating Systems Concepts by Abraham Silberschatz, Peter Baer Galvin, Greg Gagne
* academia.edu/Documents/in/CPU\_Scheduling