**Visualization of CPU Scheduling Algorithms**

A Project Report submitted in partial fulfillment of the

Requirements for the award of the degree of

**BACHELOR OF TECHNOLOGY**

**in**

**COMPUTER SCIENCE AND ENGINEERING**

**With IBM specialization in MAINFRAME TECHNOLOGY**

**by**

|  |  |
| --- | --- |
| **Name** | **Roll No.** |
| **Ankita Rani** | **R610218008** |
| **Aman Kumar** | **R610218005** |
| **Subham Kumar** | **R610218030** |

***Under the guidance of***

**Mr. Sumit Kumar**Assistant Professor

Department of Systemics

****

**Department of Systemics**

**School of Computer Science**

**University of Petroleum & Energy Studies**

**TABLE OF CONTENTS**

|  |  |  |  |
| --- | --- | --- | --- |
| DECLARATION | | | 4 |
| CERTIFICATE | |  | 5 |
| ACKNOWLEDGEMENT | | | 6 |
| ABSTRACT | |  | 7 |
| 1 | INTRODUCTION | | 8 |
| 2 | LITERATURE REVIEW | | 8 |
| 3 | IMPLEMENTATION | | 11 |
| 4 | CONCLUSION | | 14 |
| REFERENCES | |  | 15 |

**FIGURE INDEX**

|  |  |
| --- | --- |
| **FIG 1** | **11** |
| **FIG 2** | **13** |
| **FIG 3** | **14** |
| **FIG 4** | **14** |
| **FIG 5** | **15** |

**DECLARATION**

I hereby declare that this submission is my own and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which has been accepted for the award of any other Degree or Diploma of the University or other Institute of Higher learning, except where due acknowledgement has been made in the text.

Ankita Rani (Enroll No. R601218008)

Aman Kumar (Enroll No. R610218005)

Subham Kumar (Enroll No. R610218030)

**CERTIFICATE**

This is to certify that the project titled **Visualization of CPU Scheduling Algorithms** submitted by Aman Kumar (Enroll. No. R610218005), Ankita Rani (Enroll. No. R610218008), Subham Kumar (Enroll. No. R610218030) to the University of Petroleum & Energy Studies, for the award of the degree of BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE AND ENGINEERING is a bonafide record of project work carried out by them under my supervision and guidance. The content of the project, in full or parts have not been submitted to any other Institute or University for the award of any other degree or diploma.

Date: 24-03-2021

Mr. Sumit Kumar Dr. Neelu Jyoti Ahuja

**Name of Guide** **HOD (Dept. of Systemics)**

**ACKNOWLEDGEMENT**

We wish to express our deep gratitude to our guide **Mr. Sumit Kumar**, for all advice, encouragement and constant support he has given us throughout our project work. This work would not have been possible without his support and valuable suggestions.

We sincerely thank our Head of the Department, **Dr. Neelu Jyoti Ahuja, Professor and HOD, Dept. of Systemics** for her great support in doing our project in **Area** at **SoCS**.

We are also grateful to **Dr. Priyadarsan Patra, Professor and Dean, SoCS**. UPES for giving us the necessary facilities to carry out our project work successfully.

We would like to thank all our **friends** for their help and constructive criticism during our project work. Finally, we have no words to express our sincere gratitude to our **parents** who have shown us this world and for every support they have given us.

**ABSTRACT**

Scheduling is the technique used for controlling the order of the job which is to be performed by a CPU of a computer. The motive of scheduling is to engage the CPU at its maximum capacity and no process shall wait for longer time and to finish the entire task in minimum possible time. In this paper, we discuss various types of Scheduling algorithms and Compare their performance on terms of throughput and waiting time. First of them is First Come First Served (FCFS) which is a non-preemptive and the simplest scheduling. FCFS is good for long job. Second is Shortest Job First (SJF) scheduling which selects that job first which has least processing time i.e., the processing demanding the less CPU time is executed first. Next is Round Robin (RR) scheduling, it removes the drawbacks of FCFS by preempting running jobs periodically. But if the length of time quantum is too short then more time will waste in context switching. Last one is Priority Based scheduling where each process is assigned a priority (preference) to create the order of execution.

# **INTRODUCTION**

Scheduling is one of the fundamental algorithm functions of any of operating system. Since almost all the computer resources are schedule before use. The CPU is one of the primary computer resources thus scheduling algorithm is the heart of OS design. There are two types of scheduling commonly known as preemptive and non-preemptive scheduling.

In preemptive scheduling a process switches from the running state to the ready state (ex-when an interrupt occurs) or a process switches from the waiting state to the ready state (at completion of I/O).

In non-preemptive scheduling once the CPU has been allocated to the process the process keeps the CPU until it releases the CPU either by terminating or by switching usually assigned with priorities which means if the task has higher priority it would be executed first.

There is different scheduling algorithm which schedules the processes in their own way. Just like processes are schedule on the basis of first come first serve criteria in FCFS scheduling algorithm. In SJF scheduling algorithm, CPU is allocated to the processes on the basis of their requirement time for CPU, if a process requires short time for execution then it will be scheduled to CPU first whereas in Round robin scheduling algorithm the CPU is allocated to the process on the basis of time quantum. To check the best scheduling algorithm, we have different scheduling criteria like CPU utilization, waiting time, burst time, turnaround time, average waiting time and average turnaround time.

**Literature Review**

In this project "Visualization of CPU Scheduling Algorithms” here we will first simply create the process and submit them to the CPU for their execution. As the process are being submitted, a scheduler will schedule them to CPU in a specified manner (which is different for different scheduling algorithms). Firstly, the burst time for different processes will be generated by CPU. Using that burst time, we will calculate the waiting time and turnaround time for each process. When one process is getting executed in CPU all other process waits in the queue made by scheduler. After one process being executed, next process is elected by the scheduler (based on criteria different scheduling algorithm) and then it is assigned to CPU for its execution. Here in our project, we are dealing with three different algorithms i.e. First come first serve, shortest job first, Round robin scheduling algorithm. In FCFS scheduling algorithm, scheduler schedules the process on the basis of first come first serve policy which means process which enters first will get executed first. In SJF scheduling algorithm scheduler schedule the process on the basis of their execution time which means the process with shorter burst time will be executed first. In Round Robin scheduling algorithm, scheduler schedules the process on the basis of time quantum. Time quantum is the time for which any process is executed within a CPU, in case the process has larger burst time than time quantum then the process will get pre-empted and other process will enter the CPU for execution. Once all the process gets executed for the specified time i.e., the time quantum, then process which have their burst time remaining will again get executed in the same order in which they entered the CPU. After calculating the waiting time and turnaround time for each process we will calculate the average waiting time and average turnaround time by dividing total waiting time and total turnaround time by no. of process respectively. After this we will draw the bar chart of average waiting time and average turnaround time of different scheduling algorithms (In our case FCFS, SJF and RR scheduling algorithm). After this all we can easily compare that which scheduling algorithm is best.

**SYSTEM REQUIREMENTS**

Software: JDK, MySQL, Connector.

Hardware: Windows OS, 32/64-bit Processor, 4 GB RAM

**PROBLEM STATEMENT**

Here, Problem statement is brief description of what issue one has faced and is trying to improve through project or presentation. The main problem that across after development of various scheduling algorithm was it was quite difficult to decide that where to use which scheduling algorithm. As every scheduling algorithm is best at its own level. So therefore, according to us arise a need of visualization of C.P.U. scheduling through which we can compare each and every scheduling algorithm and can select from them according to our need, through graphs.

# **OBJECTIVES**

1.Tostudy and understand different type of CPU scheduling algorithms.

2.To implement the different types of CPU scheduling algorithm in JAVA.

3.To compare the different CPU scheduling algorithm on the basis of average waiting time and average turnaround time by drawing graph.

**SCHEDULE (PERT CHART)**

**STUDY PERIOD**

**Duration: 2 Weeks**

**Assigned to**: Aman, Ankita, Subham

**REQUIREMENT GATHERING**

**Duration: 2 Weeks**

**Assigned to**: Aman, Ankita, Subham

**DESIGN AND PSEUDOCODE**

**Duration: 2 Weeks**

**Assigned to**: Aman, Ankita, Subham

**CODING AND IMPLEMENTATION**

**Duration: 3 Weeks**

**Assigned to**: Aman, Ankita, Subham

**TESTING AND DEBUGGING**

**Duration: 2 Weeks**

**Assigned to**: Aman, Ankita, Subham

**PUBLISH REPORT**

**Duration: 1 Weeks**

**Assigned to**: Aman, Ankita, Subham

Fig 1

**Implementation**

* JDBC: Java Database Connectivity is an API which we have used to connect to MySQL database.
* Creating Connection Object

Connection con =DriverManager.*getConnection*("jdbc:mysql://localhost:3306/project","root","root");

* Creating Statement Object

Connection st=con.createStatement();

* Process id, burst time, arrival time for each process has been taken from the MySQL database into arraylist and then scheduling algorithms are applied on it to calculate average waiting time and average turn around time. Based on average waiting time and average turnaround time we have compared different scheduling algorithms.

**FIRST COME FIRST SERVE ALGORITHM**

* Input the processes along with their burst time.
* Find waiting time for all processes.
* As first process that comes need not to wait so

waiting time for process 1 will be 0 i.e. wt[0] = 0.

* Find waiting time for all other processes i.e. for

process i -> wt[i] = bt[i-1] + wt[i-1]

* Find turnaround time = wt[] + bt[] for all processes.
* Find average waiting time =

total\_waiting\_time / no\_of\_processes.

* Similarly, find average turnaround time =

total\_turn\_around\_time / no\_of\_processes.

**SHORTEST JOB FIRST ALGORITHM**

* We will sort all the processes in increasing order according to burst time.
* Then simply apply the above FCFS algorithm.

**ROUND ROBIN ALGORITHM**

* Create an array **rem\_bt[]** to keep track of remaining

burst time of processes. This array is initially a

copy of bt[].

* Create another array **wt[]** to store waiting times

of processes. Initialize this array as 0.

* Initialize time: t = 0
* Keep traversing the all processes while all processes

are not done. Do following for i'th process if it is

not done yet.

If rem\_bt[i] > quantum

t = t + quantum

bt\_rem[i] -= quantum;

else

t = t + bt\_rem[i];

wt[i] = t - bt[i]

bt\_rem[i] = 0; // This process is over

**Output Screen**

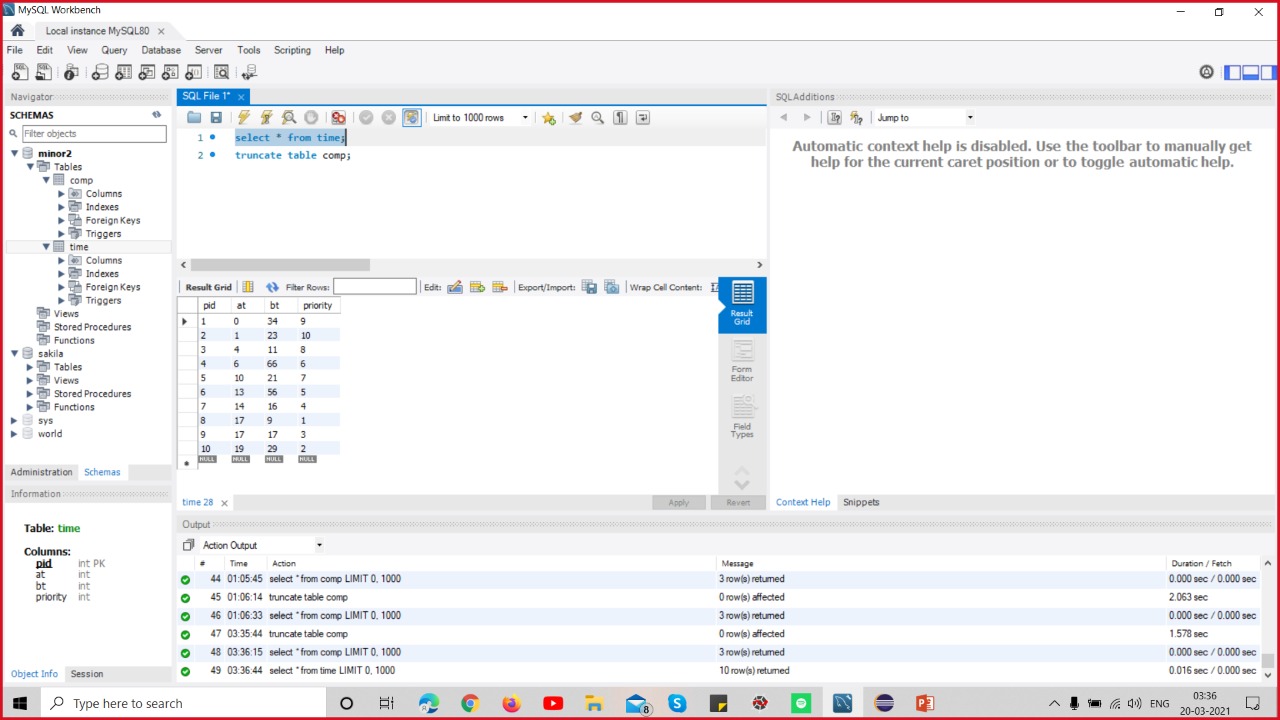


Fig 2

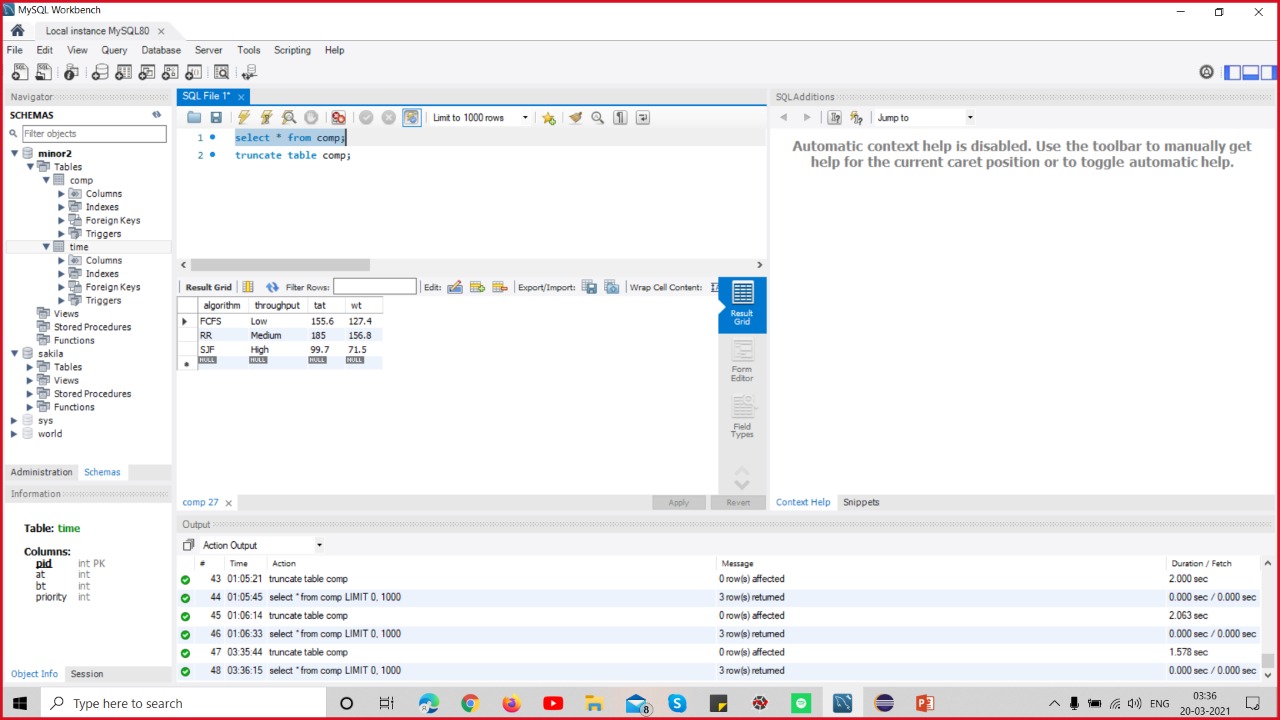


Fig 3

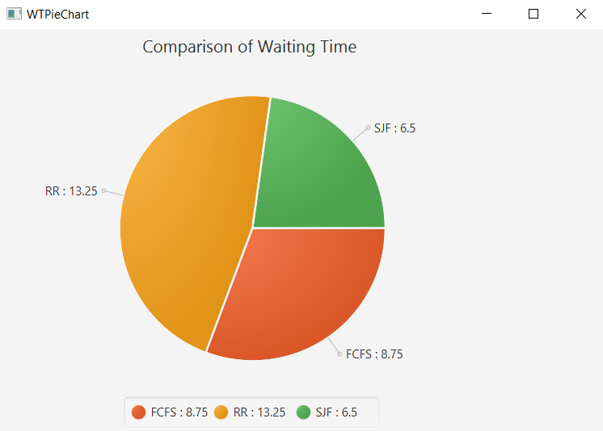


Fig. 4

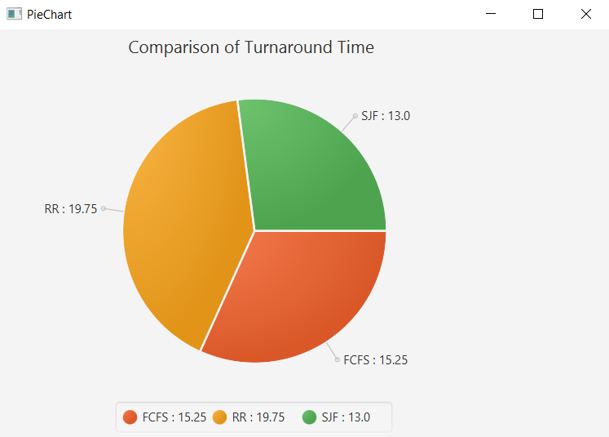
****

Fig. 5

**Limitations**

For every algorithm there is different limitations which are described below:

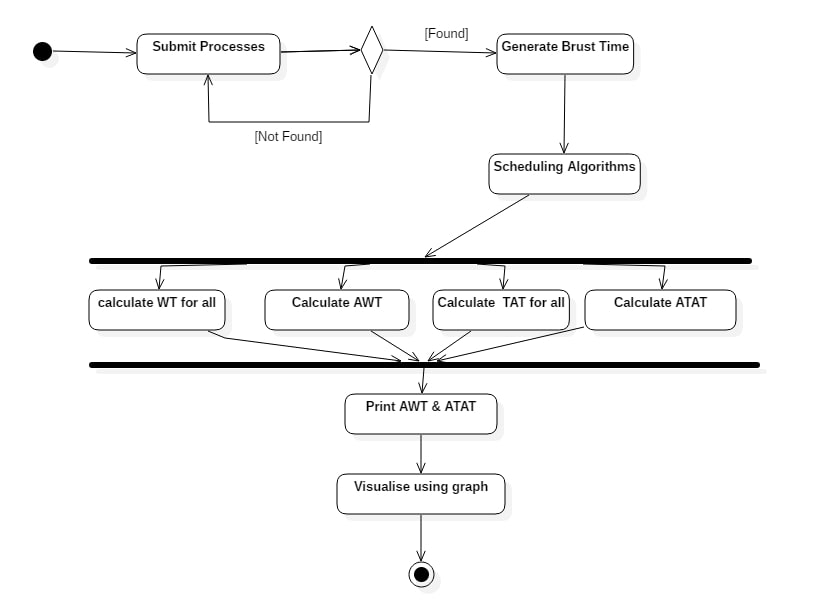
FCFS: -Since no option for pre-emption of a process then CPU executes it until it ends. It leads to starvation of short time processes if long time process starts to executes

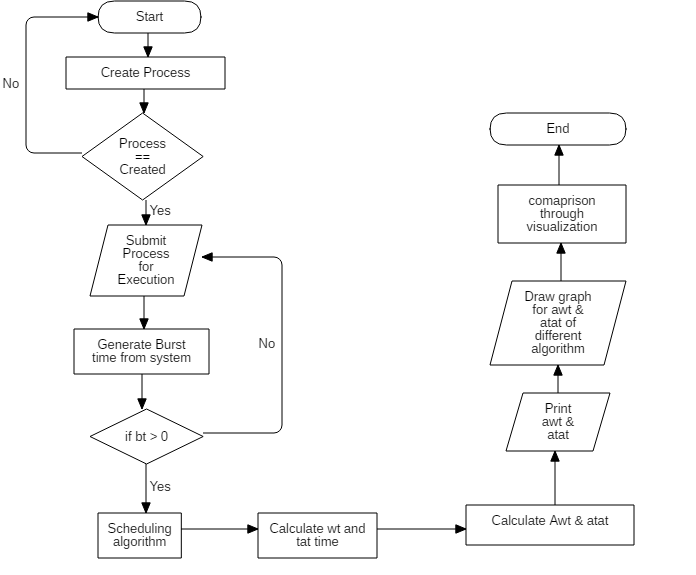
SJF: -Impossible for CPU to know time taken by a process, beforehand. Longer processes will suffer starvation.

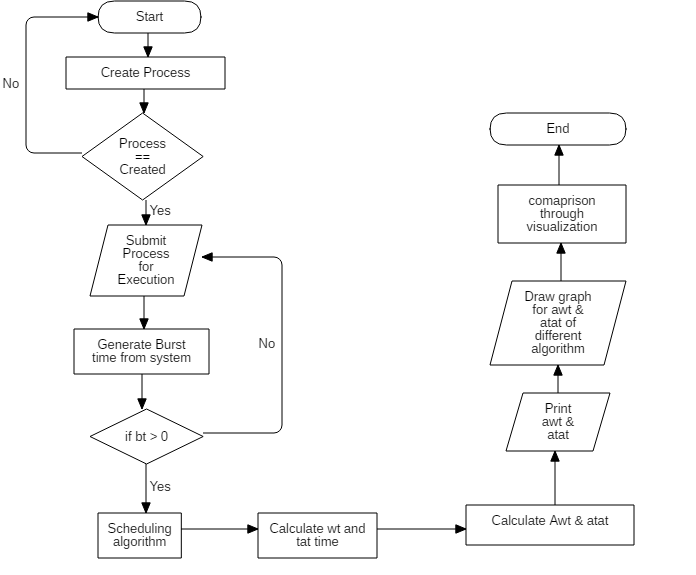
RR: -If time quantum is short it leads to decrease in efficiency of CPU and if time quantum is greater than FCFS.

**Conclusion**

Through Visualization of CPU scheduling algorithms, we can easily identify that which scheduling algorithm is best since we have a graph showing average waiting time and average turnaround time of different algorithms thereby easily comparative.







**References**

* [Optimum Time Quantum](https://www.researchgate.net/publication/264888532_Determining_the_Optimum_Time_Quantum_Value_in_Round_Robin_Process_Scheduling_Method)
* [Reading Materials for CPU Scheduling](https://www.u-aizu.ac.jp/~yliu/teaching/os/lec10r.html)
* [Analysis of Various Scheduling Algorithm](http://ijarcet.org/wp-content/uploads/IJARCET-VOL-2-ISSUE-4-1488-1491.pdf)
* [Comparison Analysis of CPU Scheduling](https://www.researchgate.net/publication/305492984_Comparison_Analysis_of_CPU_Scheduling_FCFS_SJF_and_Round_Robin)