Aryabhatta Journal of Mathematics & Informatics Vol. 6, No. 1, Jan-July, 2014 ISSN : 0975-7139
Journal Impact Factor (2013) : 0.489

# COMPARATIVE STUDY OF CPU SCHEDULING ALGORITHMS FOR REAL-TIME OPERATING SYSTEM IN UNCERTAINTY

**Er. Silky Miglani[*], Er. Sachin Sharma[**], Dr. T.P Singh[***]**
[*]Research Scholar, CSE, Yamuna Institute of Engg. & Technology, Gadholi, Yamunanagar, Haryana
[**]Research Scholar, CSE, Yamuna Institute of Engg. & Technology, Gadholi, Yamunanagar, Haryana
[***]Professor Mathematics & Computing, Yamuna Institute of Engg. & Tech., Gadholi, Yamunanagar, Haryana
Email: [1]silky.cse@gmail.com, [2]sachinsharma.one@gmail.com, [3]tpsingh78@yahoo.com

## ABSTRACT:
***The Real-Time scheduling characteristics most nearly concerned with uncertainty; the execution or burst time of tasks and the umbrella category of real time constraints under which dead line, ready time, and task period fall. The most obvious place to introduce fuzzy concepts for modelling uncertainty in scheduling is with a task execution time. With this intent, the objective of present paper is to measure the performance of different CPU scheduling policies in fuzzy environment. The simulator designed accesses the performance of Round Robin, Priority ( both pre-emptive and non pre-emptive) scheduling policies in terms of average waiting time and average turnaround time for a number of processes in uncertainty. In Round Robin and Priority CPU Scheduling algorithm, the main concern is with the uncertainty in Burst Time and the increased waiting time and turnaround time. The decision for these is usually based on parameters which are assumed to be précised. However, in many cases, the values of these parameters are vague and imprecise rather than stochastic. Hence, considered fuzzy in nature. Ready queue is maintained in FCFS queue discipline.***

**Keywords**: ***Simulator, Scheduling, Round Robin, Priority Scheduling, Fuzzy Environment, Turnaround Time.***

## INTRODUCTION:
The investigation to deal with problems in which facilities are fixed and the sequence of servicing the waiting jobs is subject to control, have given arise to an elegant theory of scheduling .In multitasking and multiprogramming environment, the way of dispatching the processes to the CPU is called process scheduling. The main goal of the scheduling is to maximize the performance of the system as well as to minimize waiting and turnaround time. Schedule may be evaluated according to some criterion of efficiency. The performance regarding the efficiency of scheduling algorithm provided to the request and the optimal utilization of processor (resources) in the system is determined by the order in which the request is serviced. Each process needs two factors when entering the operating system; first CPU and second I/O. The processes are divided into two general groups of CPU limited and I/O limited processes based on the need to these two factors. However, most processes always need the two types of sources. In a system with many processes available, there will be a competition between these processes to acquire the resources. One of the most difficult problems in designing the operating systems is the timely allocation of **CPU** to the processes and retrieving them. There is a need to keep the CPU as busy as possible.

Various researchers have done a lot of work in scheduling area. Johnson [1954] gave a heuristic approach for solving n jobs, 2stage and in some restrictive cases, 3 stage flow shop scheduling with the objective to minimize total elapsed time. Liu and Leyland [1973] for the first time studied priority driven algorithm. Singh T.P. [1985, 2006, and 2010] extended the work taking in account the various parameters in scheduling problems as job blocking, transportation time, break down interval, priority in jobs etc. which are applied in more realistic situation.

Tannenbaum et al [2003] designed two techniques used in servers; one is pre-emption of request and second is re-ordering of request in order to improve the average quality of service provided by a server. Yashuwanth et al.

[2010] developed a modified Round Robin algorithm. Scheduling decision for embedded software plays an important role on system performance. The designer should select the right scheduling algorithm at high abstraction levels in order to save him from error-prone and time consuming task or task priority assignments at the final stage of system design.

Garg and Vikram Singh [2010] developed a simulator for performance evaluation of CPU scheduling policies for real time operating systems in crisp set under stochastic environment. Mehdi, Neshat etal[2012] proposed an algorithm which optimizes the average waiting and response time for system process and compared the performance of this algorithm with classical scheduling algorithms under non-fuzzy environment. Singh T.P. et al [2013] developed a heuristic algorithm for general priority job scheduling in fuzzy environment and obtained optimal solution to the general machine scheduling problem. Recently, Sachin and Silky etal[2014] extended the work of earlier researchers by considering the parameters in triangular fuzzy environment. This work is further an extended work of Silky & Sachin in the sense that a comparative study of performance measure of CPU scheduling policies has been studied in which CPU burst time are considered in trapezoidal fuzzy nature and we have made an attempt to explore the best scheduling policy for embedded system work on real time operating system when the environment is fuzzy. Practically it has been observed that the burst time consumed by the process differs to certain degree of level while executing, it is neither deterministic nor probabilistic but vary in an interval to some degree due to register or memory transfers or certain background processes. This burst time of jobs is different in different situations and creates fuzziness. Membership function includes the fuzzy sets of "very early, "early," "medium," and "late," while criticality (task importance) it includes the fuzzy sets of "very important, "important," "average," and "unimportant." Several CPU scheduling algorithms have different features and no single one is ideal absolutely for every application.

The following are the terms related to our study:

1. **CPU Utilization**: The time for which the CPU is used by different processes.
2. **Throughput**: Number of processes completed per unit time.
3. **Waiting Time**: The amount of time that a process spends waiting in the ready queue.
4. **Response Time**: the time from the submission of a request until the first response is produced.

The study is practically more significant and relevant and has been made to achieve the following objectives:

1. Minimum context switches.
2. Maximum CPU utilization.
3. Maximum throughput.
4. Minimum turnaround time.
5. Minimum waiting time.
6. Minimum Response Time.

The paper is organized as :

Following introduction in section 1, in section 2, CPU scheduler with Round Robin and Priority based scheduling has been described. In section 3, the assumptions and notations have been discussed. In section 4, we have developed a simulator in C language under WINDOWS Operating System on an INTEL compatible machine to develop an algorithm. In section 5, the performance measure results have been presented in tabular form as well as in statistical bar charts. Finally in section 6, the concluding remarks have been made.

## 2. CPU SCHEDULAR

The simulator is designed for two types of CPU scheduling namely Round Robin (RR) and priority based.
RR uses time quantum or time slice to allocate the CPU to a process. The process has to release CPU voluntarily as soon as the quantum expires and gets inserted into ready queue. The ready queue behaves as a circular queue kept as FCFS discipline. The CPU scheduler goes around the ready queue and allocates the CPU to each process for a time interval of one time quantum where any one of thing may happen. The process may have a CPU burst of less than or equal to one time quanta in which the process after expiry of the time quantum is pre-empted, placed at the tail of ready queue. This is done through swapping process. By the use of one or more context switch, there occurs a variation in computation time. This variation may be less or more depending upon the use of context switch. Hence, computation time is considered fuzzy in nature.

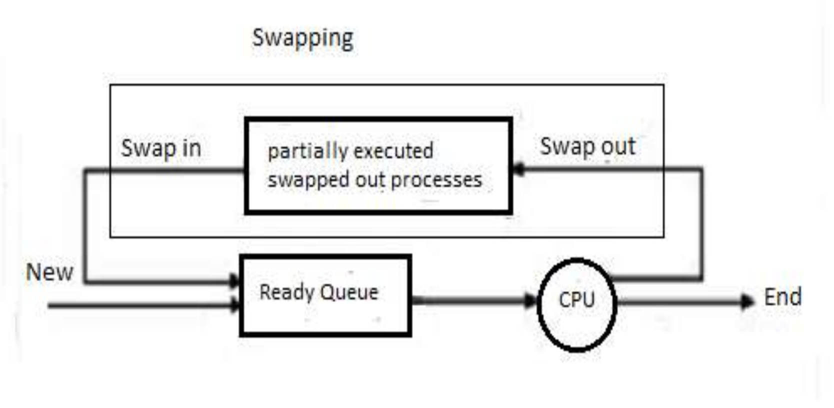


**Figure 1: Scheduler**

Priority based scheduling can be pre-emptive or non pre-emptive. The priority (Integer and user defined) is associated with each process and the CPU is allocated to the process with highest priority. Equal priority processes are scheduled in FCFS. Pre-emptive priority scheduling algorithm will release the CPU if the priority of the newly arrived process is higher than the priority of the currently running process but non pre-emptive scheduling algorithm will simply put the new process at the head of the ready queue.

## 3. ASSUMPTIONS:

i. All processes are CPU bound. No process is I/O bound.
ii. Processes with same arrival time are scheduled.

## 4. DESIGN OF SIMULATOR IN FUZZY ENVIRONMENT

In our study, the simulator has been developed in C language under windows operating system on an INTEL compatible machine to execute scheduling algorithm. Real time algorithm has been developed as a comprehensive software package which runs a simulation in real-time, generates useful data for analysis. In this module, we have considered the effect of context-switching on the performance of RR scheduling. **Turnaround time also depends on the size of the time quantum.**
For time variation, we have applied the concept of fuzzy arithmetic given by L.A. Zadeh [1965]. Fuzzy logic is used to decide the **burst** time when the information about it, is either incomplete or variates with change in environment. So, it is better to take this time in fuzzy environment rather than stochastic. Fuzzy logic offers a better way to represent the reality. In fuzzy logic, a statement is true to various degrees ranging from false, half truth to complete

truth. Fuzzy approach is better and effective way to tackle uncertainty in scheduling. Usually, fuzzy logic is used to convert numerical data in different scale like small, medium, fast etc using membership function from 0(false) to 1(true).In the modern system, the two latest technologies speed step and power gating are applied in order to reduce power consumption. Further, these technologies bring the variation in burst times causing an uncertainty. It is because of the modern processor contain more than one core to execute the jobs and in order to make power savings, these cores get active on the basis of either total load i.e. burst time of each job or number of available processes in ready queue. Burst time can be identified as a linguistic variable that may be very low, low, average or high due to change in operating frequency of CPU. In the trapezoidal fuzzy number, it is justified in this system.

The trapezoidal fuzzy numbers <a, b, c, d> represent the numbers in different environments. A fuzzy number is simply an ordinary number whose precise value is somewhat uncertain. A very convenient way to describe fuzzy numbers is to use modifying words, as nearly, closely about crudely with larger and complex uncertainties. For the defuzzification of trapezoidal fuzzy number, we apply graded mean integration. Chen and Hsieh[1999] introduced graded mean representation method based on the interior value of graded mean h-level of generalised fuzzy number for defuzzifying it.

Let 'A' be a trapezoidal fuzzy number denoted as A= (a, b, c, d) then Graded Mean Integration of 'A' is given by the formula:

$$P(A) = \frac{\int_0^1 \frac{h}{2}[(a+d)+h(b-a-d+c)]dh}{\int_0^1 h.dh} = \frac{(a+2b+2c+d)}{6} \qquad \text{(i)}$$

## 5. RESULT AND ANALYSIS

Considering the time quantum for round robin to be 4, the CPU burst time is calculated for different files in fuzzy environment using expression (i). The parameter such as waiting time and turnaround time for different jobs have been calculated using round robin and priority based scheduling as shown in table 1-4. The final results showing average waiting time and average turnaround time for different jobs using round robin and priority based scheduling are shown in tables (5,6). All the readings are taken in nanoseconds (1 ns = $10^{-9}$ seconds).

**For 10 processes:**

**Table 1: Performance Measure of Round Robin Scheduling in Fuzzy(Trapezoidal)**

| S.No. | BURST TIME IN FUZZY ENVIRONMENT | | | | DEFUZZIFIED CPU BURST TIME | WAITING TIME | TURN AROUND TIME |
|---|---|---|---|---|---|---|---|
| | <a | b | c | d> | | | |
| 1 | 11.2 | 11.42 | 12.5 | 13.9 | 12.15666667 | 99.97 | 112.12 |
| 2 | 9.8 | 10.38 | 11.5 | 12.85 | 11.06833333 | 75.51 | 86.57 |
| 3 | 16.53 | 18.43 | 19.7 | 25.38 | 19.695 | 116.13 | 135.82 |
| 4 | 10.6 | 10.9 | 11.88 | 12.42 | 11.43 | 82.57 | 94 |
| 5 | 20.7 | 22.7 | 23.7 | 23.9 | 22.9 | 130.61 | 153.51 |
| 6 | 8.57 | 9.94 | 10.3 | 10.8 | 9.975 | 90.009 | 99.97 |
| 7 | 33.11 | 33.87 | 34 | 34.68 | 33.92166667 | 144.83 | 178.75 |
| 8 | 7.11 | 7.5 | 7.68 | 8.12 | 7.598333333 | 63.99 | 71.51 |
| 9 | 17.11 | 18.66 | 19.11 | 20.1 | 18.79166667 | 127.82 | 146.61 |
| 10 | 30.66 | 30.82 | 31.61 | 32.4 | 31.32 | 145.51 | 176.83 |
| **Average-------------------->** | | | | | | **107.6949** | **125.569** |

**Table 2: Performance Measure of Priority Pre-emptive Scheduling in Fuzzy (Trapezoidal)**

| S.No. | BURST TIME IN FUZZY ENVIRONMENT | | | | DEFUZZIFIED CPU BURST TIME | PRE-EMPTIVE | | NON PRE-EMPTIVE | |
|---|---|---|---|---|---|---|---|---|---|
| | <a | b | c | d> | | WT | TAT | WT | TAT |
| 1 | 11.2 | 11.42 | 12.5 | 13.9 | 12.15666667 | 0.85 | 13 | 0 | 12.15 |
| 2 | 9.8 | 10.38 | 11.5 | 12.85 | 11.06833333 | 13.94 | 25 | 12.149 | 23.2 |
| 3 | 16.53 | 18.43 | 19.7 | 25.38 | 19.695 | 25.3 | 45 | 23.21 | 42.9 |
| 4 | 10.6 | 10.9 | 11.88 | 12.42 | 11.43 | 45.57 | 57 | 42.9 | 54.33 |
| 5 | 20.7 | 22.7 | 23.7 | 23.9 | 22.9 | 57.1 | 80 | 54.33 | 77.23 |
| 6 | 8.57 | 9.94 | 10.3 | 10.8 | 9.975 | 80.03 | 90 | 77.23 | 87.2 |
| 7 | 33.11 | 33.87 | 34 | 34.68 | 33.92166667 | 90.08 | 124 | 87.2 | 121.12 |
| 8 | 7.11 | 7.5 | 7.68 | 8.12 | 7.598333333 | 124.41 | 132 | 121.12 | 128.71 |
| 9 | 17.11 | 18.66 | 19.11 | 20.1 | 18.79166667 | 132.2 | 151 | 128.7 | 147.5 |
| 10 | 30.66 | 30.82 | 31.61 | 32.4 | 31.32 | 151.68 | 183 | 147.5 | 178.82 |
| | | | | | **Average--------------------->** | **72.116** | **90** | **69.43** | **87.31** |

**For 15 processes:**

**Table 3: Performance Measure of Round Robin Scheduling in Fuzzy(Trapezoidal)**

| S.No. | BURST TIME IN FUZZY ENVIRONMENT | | | | DEFUZZIFIED CPU BURST TIME | WT | TAT |
|---|---|---|---|---|---|---|---|
| | <a | b | c | d> | | | |
| 1 | 11.2 | 11.42 | 12.5 | 13.9 | 12.15666667 | 153.8 | 165.95 |
| 2 | 9.8 | 10.38 | 11.5 | 12.85 | 11.06833333 | 115.34 | 126.4 |
| 3 | 16.53 | 18.43 | 19.7 | 25.38 | 19.695 | 178.96 | 198.65 |
| 4 | 10.6 | 10.9 | 11.88 | 12.42 | 11.43 | 122.4 | 133.83 |
| 5 | 20.7 | 22.7 | 23.7 | 23.9 | 22.9 | 197.92 | 220.82 |
| 6 | 8.57 | 9.94 | 10.3 | 10.8 | 9.975 | 129.83 | 139.8 |
| 7 | 33.11 | 33.87 | 34 | 34.68 | 33.92166667 | 215.17 | 249.09 |
| 8 | 7.11 | 7.5 | 7.68 | 8.12 | 7.598333333 | 84 | 91.58 |
| 9 | 17.11 | 18.66 | 19.11 | 20.1 | 18.79166667 | 190.65 | 209.44 |
| 10 | 30.66 | 30.82 | 31.61 | 32.4 | 31.32 | 215.85 | 247.17 |
| 11 | 12.22 | 12.66 | 13.45 | 13.67 | 13.01833333 | 173.95 | 186.96 |
| 12 | 21.39 | 22.55 | 23.88 | 23.96 | 43.035 | 208.82 | 231.85 |
| 13 | 6.99 | 7.48 | 8.13 | 8.35 | 7.76 | 103.58 | 111.34 |
| 14 | 15 | 15.58 | 17.45 | 17.87 | 16.48833333 | 201.44 | 217.92 |
| 15 | 8.64 | 9.38 | 10.7 | 11.2 | 10 | 155.8 | 165.8 |
| | | | | | **Average--------------------->** | **163.17** | **179.773** |

**Table 4: Performance Measure of Priority Pre-emptive Scheduling in Fuzzy (Trapezoidal)**

| S.No. | BURST TIME IN FUZZY ENVIRONMENT | | | | DEFUZZIFIED CPU BURST TIME | PRE-EMPTIVE | | NON PRE-EMPTIVE | |
|---|---|---|---|---|---|---|---|---|---|
| | <a | b | c | d> | | WT | TAT | WT | TAT |
| 1 | 11.2 | 11.42 | 12.5 | 13.9 | 12.15666667 | 0.85 | 13 | 0 | 12.15 |
| 2 | 9.8 | 10.38 | 11.5 | 12.85 | 11.06833333 | 13.94 | 25 | 12.15 | 23.2 |
| 3 | 16.53 | 18.43 | 19.7 | 25.38 | 19.695 | 25.3 | 45 | 23.21 | 42.9 |
| 4 | 10.6 | 10.9 | 11.88 | 12.42 | 11.43 | 45.57 | 57 | 42.9 | 54.33 |
| 5 | 20.7 | 22.7 | 23.7 | 23.9 | 22.9 | 57.1 | 80 | 54.33 | 77.23 |
| 6 | 8.57 | 9.94 | 10.3 | 10.8 | 9.975 | 80.03 | 90 | 77.23 | 87.2 |
| 7 | 33.11 | 33.87 | 34 | 34.68 | 33.92166667 | 90.08 | 124 | 87.2 | 121.12 |
| 8 | 7.11 | 7.5 | 7.68 | 8.12 | 7.598333333 | 124.41 | 132 | 121.12 | 128.71 |
| 9 | 17.11 | 18.66 | 19.11 | 20.1 | 18.79166667 | 132.2 | 151 | 128.7 | 147.5 |
| 10 | 30.66 | 30.82 | 31.61 | 32.4 | 31.32 | 151.68 | 183 | 147.5 | 178.82 |
| 11 | 12.22 | 12.66 | 13.45 | 13.67 | 13.01833333 | 183.98 | 197 | 178.82 | 191.83 |
| 12 | 21.39 | 22.55 | 23.88 | 23.96 | 23.035 | 197.96 | 221 | 191.83 | 214.86 |
| 13 | 6.99 | 7.48 | 8.13 | 8.35 | 7.76 | 221.24 | 229 | 214.85 | 222.61 |
| 14 | 15 | 15.58 | 17.45 | 17.87 | 16.48833333 | 229.52 | 246 | 222.61 | 239.09 |
| 15 | 8.64 | 9.38 | 10.7 | 11.2 | 10 | 246 | 256 | 239.09 | 249.09 |
| **Average-------------------->** | | | | | | **119.99** | **136.6** | **116.1** | **132.72** |

**Table 5: Comparison of Waiting Time in Fuzzy**

| Total Processes | Round Robin Scheduling | Non-Preemptive Scheduling | Preemptive Scheduling |
|---|---|---|---|
| 10 | 107.69 | 69.43 | 72.116 |
| 15 | 163.17 | 116.1 | 119.99 |

The output for two different scheduling policies has been shown through bar diagram. The RR takes maximum time. Figure 2 shows different values of processes and their respective waiting time values in RR scheduling as well as non pre-emptive priority scheduling.

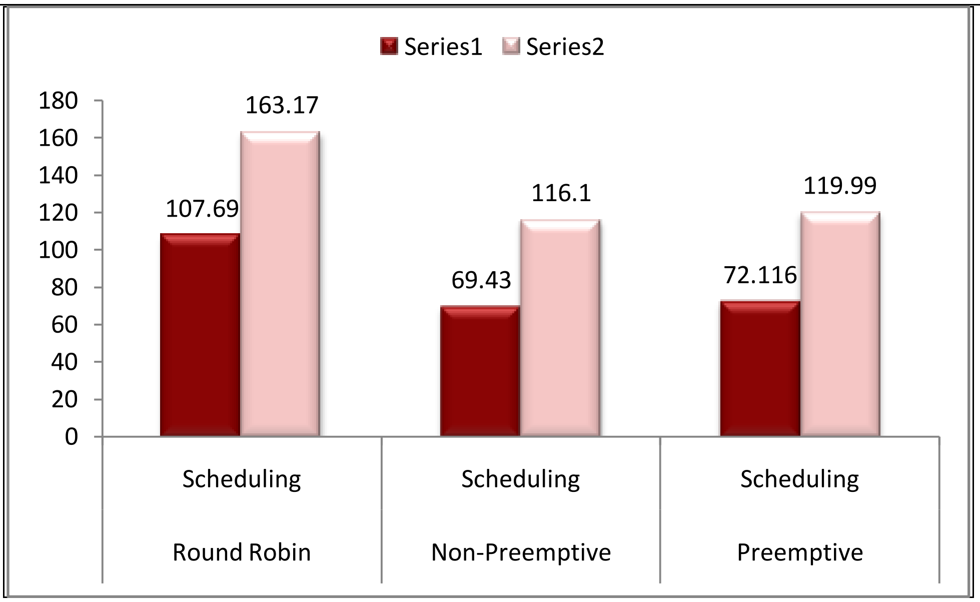


**Figure 2: Average waiting times**

**Table 6: Comparison of Turnaround Time**

| Total Processes | Round Robin Scheduling | Non-Preemptive Scheduling | Preemptive Scheduling |
|---|---|---|---|
| 10 | 125.56 | 87.31 | 90 |
| 15 | 179.77 | 132.72 | 136.6 |

Figure 3 represents the bar diagram showing the average turnaround time for 10 and 15 processes.

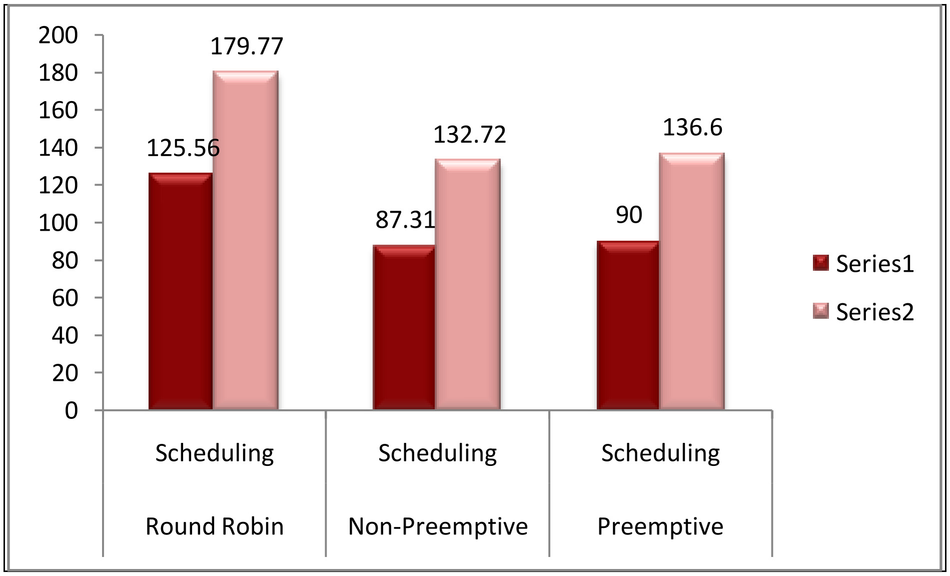


**Figure-3:** Average Turnaround times

After sorting Burst time through quick sort, the results come out to be as follows (Table 7-10) that minimises the context switches among the processes due to which the average waiting and turnaround time has been reduced further. The table shows betterment in CPU scheduling algorithms.

**Table 7: Performance Measure of RR and Priority Scheduling FOR SORTED FILES in Fuzzy(Trapezoidal)**

| S.No. | BURST TIME IN FUZZY ENVIRONMENT | | | | DEFUZZIFIED CPU BURST TIME | ROUND ROBIN | | PRE-EMPTIVE | | NON PRE-EMPTIVE | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | <a | b | c | d> | | WT | TAT | WT | TAT | WT | TAT |
| 1 | 7.11 | 7.5 | 7.68 | 8.12 | 7.598333333 | 36 | 43.59 | 0.41 | 8 | 0 | 7.59 |
| 2 | 8.57 | 9.94 | 10.3 | 10.8 | 9.975 | 71.58 | 81.55 | 8.03 | 18 | 7.59 | 15.35 |
| 3 | 9.8 | 10.38 | 11.5 | 12.85 | 11.06833333 | 73.55 | 84.61 | 18.94 | 30 | 15.34 | 25.32 |
| 4 | 10.6 | 10.9 | 11.88 | 12.42 | 11.43 | 76.61 | 88.04 | 30.57 | 42 | 25.32 | 35.32 |
| 5 | 11.2 | 11.42 | 12.5 | 13.9 | 12.15666667 | 100.04 | 112.19 | 42.85 | 55 | 35.32 | 46.38 |
| 6 | 16.53 | 18.43 | 19.7 | 25.38 | 19.695 | 116.19 | 135.88 | 55.3 | 75 | 46.38 | 57.81 |
| 7 | 17.11 | 18.66 | 19.11 | 20.1 | 18.79166667 | 119.88 | 138.67 | 75.2 | 94 | 57.8 | 69.95 |
| 8 | 20.7 | 22.7 | 23.7 | 23.9 | 22.9 | 130.67 | 153.57 | 94.1 | 117 | 69.96 | 82.97 |
| 9 | 30.66 | 30.82 | 31.61 | 32.4 | 31.32 | 141.57 | 172.89 | 117.68 | 149 | 82.96 | 99.44 |
| 10 | 33.11 | 33.87 | 34 | 34.68 | 33.92166667 | 144.89 | 178.81 | 149.08 | 183 | 99.44 | 118.73 |
| **Average--------------------->** | | | | | | **101.098** | **118.98** | **59.216** | **77.1** | **44.011** | **55.886** |

**Table 8: Performance Measure of RR and Priority Preemptive Scheduling FOR SORTED FILES in Fuzzy(Trapezoidal)**

| S.No. | BURST TIME IN FUZZY ENVIRONMENT | | | | DEFUZZIFIED CPU BURST TIME | ROUND ROBIN | | PRE-EMPTIVE | | NON PRE-MPTIVE | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | <a | b | c | d> | | WT | TAT | WT | TAT | WT | TAT |
| 1 | 7.11 | 7.5 | 7.68 | 8.12 | 7.5983333 | 56 | 63.59 | 0.41 | 8 | 0 | 7.59 |
| 2 | 6.99 | 7.48 | 8.13 | 8.35 | 7.76 | 59.58 | 67.34 | 8.24 | 16 | 7.59 | 15.35 |
| 3 | 8.57 | 9.94 | 10.3 | 10.8 | 9.975 | 111.34 | 121.32 | 16.03 | 25 | 15.34 | 25.32 |
| 4 | 8.64 | 9.38 | 10.7 | 11.2 | 10 | 113.32 | 123.32 | 26 | 36 | 25.32 | 35.32 |
| 5 | 9.8 | 10.38 | 11.5 | 12.85 | 11.068333 | 115.32 | 126.37 | 36.94 | 48 | 35.32 | 46.38 |
| 6 | 10.6 | 10.9 | 11.88 | 12.42 | 11.43 | 118.37 | 129.8 | 48.57 | 60 | 46.38 | 57.81 |
| 7 | 11.2 | 11.42 | 12.5 | 13.9 | 12.156667 | 153.8 | 165.95 | 60.85 | 73 | 57.8 | 69.95 |
| 8 | 12.22 | 12.66 | 13.45 | 13.67 | 13.018333 | 153.95 | 166.96 | 73.99 | 87 | 69.96 | 82.97 |
| 9 | 15 | 15.58 | 17.45 | 17.87 | 16.488333 | 178.96 | 195.44 | 87.52 | 104 | 82.96 | 99.44 |
| 10 | 17.11 | 18.66 | 19.11 | 20.1 | 18.791667 | 179.44 | 198.23 | 104.2 | 123 | 99.44 | 118.73 |
| 11 | 16.53 | 18.43 | 19.7 | 25.38 | 19.695 | 182.23 | 201.92 | 123.3 | 143 | 118.2 | 137.92 |
| 12 | 20.7 | 22.7 | 23.7 | 23.9 | 22.9 | 197.92 | 220.82 | 143.1 | 166 | 137.9 | 160.82 |
| 13 | 21.39 | 22.55 | 23.88 | 23.96 | 23.035 | 200.82 | 223.85 | 167 | 190 | 160.8 | 183.85 |
| 14 | 30.66 | 30.82 | 31.61 | 32.4 | 31.32 | 211.85 | 243.17 | 190.7 | 222 | 183.9 | 215.17 |
| 15 | 33.11 | 33.87 | 34 | 34.68 | 33.921667 | 215.17 | 249.09 | 222.1 | 256 | 215.2 | 249.09 |
| **Average--------------------->** | | | | | | **149.87** | **166.48** | **87.26** | **103.8** | **83.74** | **100.35** |

**Table 9: Comparison of Waiting Time**

| Total Processes | Round Robin Scheduling | Preemptive Scheduling | Non-Preemptive Scheduling |
|---|---|---|---|
| 10 | 101.098 | 59.216 | 44.011 |

| | | | |
|---|---|---|---|
| 15 | 149.87 | 87.26 | 83.74 |

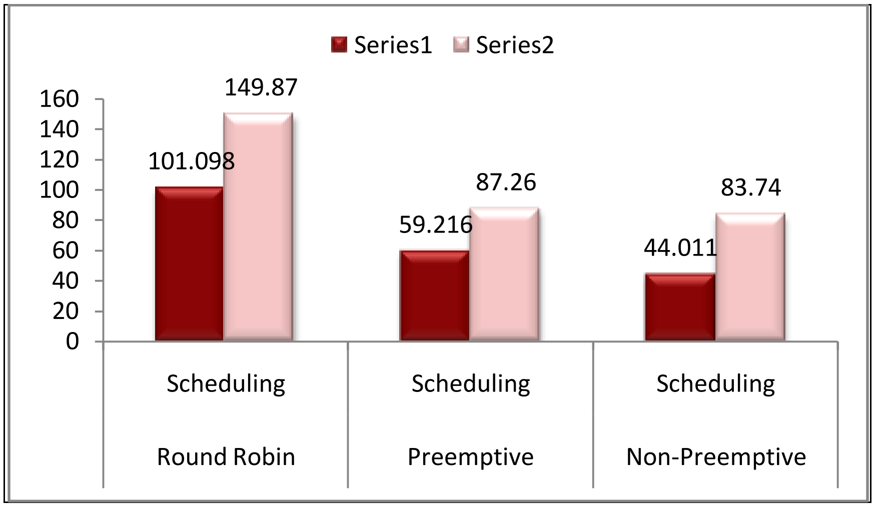


**Figure 4:** Average waiting times after sorting

**Table 10: Comparison of Turn around Time**

| Total Processes | Round Robin Scheduling | Preemptive Scheduling | Non-Preemptive Scheduling |
|---|---|---|---|
| 10 | 118.98 | 77.1 | 55.88 |
| 15 | 166.48 | 103.8 | 100.35 |

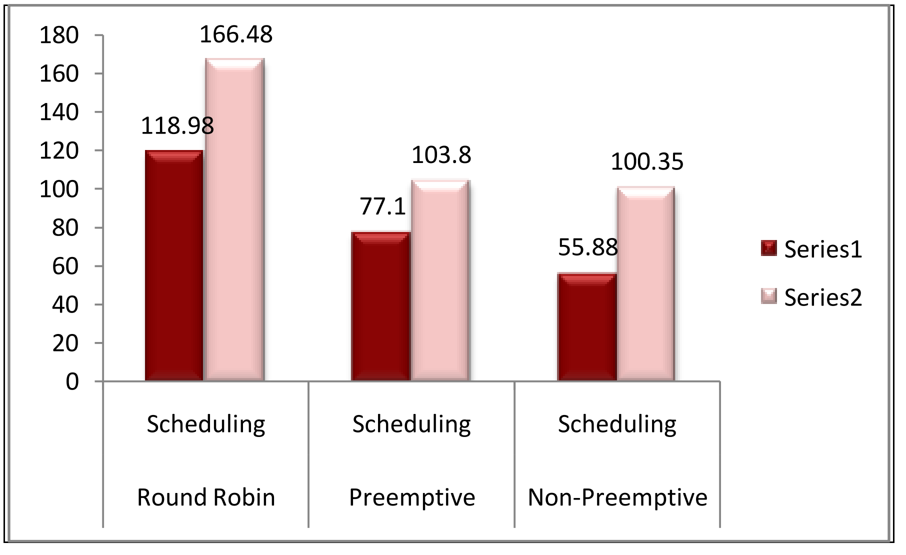


**Figure 5:** Average turnaround time after sorting

We have shown the output of two different scheduling polices through bar diagram. The RR takes more execution time. Table and graphs shows different values of waiting time of processes in RR scheduling as well as Priority based scheduling. Also average turnaround times are shown.

## 6. CONCLUDING REMARKS & FUTURE SCOPE

We have developed a simulator in C by taking multi-processor system. The processes are selected from the head of the ready queue all executed on RR and non pre-emptive and pre-emptive basis. The data from priority scheduling has been compared with RR. The ultimate result indicates that both pre-emptive and non pre-emptive are better than RR. Since the average waiting time under RR policy takes a quite longer time. If we take only single processor and triangular fuzzy number is considered, the data tally with the earlier work done by SACHIN and SILKY.

We have developed a simulator in C .The processes are selected from the head of ready queue and are executed on round robin and non pre-emptive priority basis. The data from Priority Scheduling are compared with RR. Our results indicate that non pre-emptive scheduling is better than round robin and priority (pre-emptive) scheduling. The average waiting time under RR policy takes a quite longer time.

The work can be extended by taking multiple processors by dividing the ready queue into parts .Further research can be done by taking time quanta under fuzzy environment.

## REFERENCES


1. Zadeh, L.A,(1965), "*Fuzzy sets and systems*", System Theory, Brooklyn N.Y., pp 29-37
2. C.L. Liu and J.Leyland, (1973)"Scheduling algorithm for multiprogramming in a hard real-time environment"J.Amer. Compt.Mach. Vol. 20 No. 1, pp 4M1.
3. Singh, T.P (1985), "On 2xn flow-shop problem involving job-block, transportation time, arbitrary time and breakdown machine time", PAMS Vol. XXI ,No 1-2.
4. Ramamritham,K. Stankovic,J.A.( January 1994), "*Scheduling algorithms and operating systems support for real-time systems*".
5. Chen S.H & Hseih, C.J (1999), "Graded Mean Integration Representation of Generalized Fuzzy Numbers", Journal of Chinese Fuzzy System,Vol.5(2),pp 1-7.
6. Tannenbaum Andrew S. And Woodhull Albert S. ( 2003), "*Operating Systems: Design and Implementation*",2nd Edition, Prentice Hall,India.
7. Silber Schartz, Galvin & Gague (2009) 'Operating System Concept' 8th Edition Wiley & Sons.6
8. Yashuwanth. C, Dr. R. Ramesh,(March 2010) "*Design of Real Time scheduler simulator and development of modified Round Robin architecture*", IJCSNS International Journal of Computer Science and Network Security,VOL.10 No.3.
9. Sunita Gupta ,T.P Singh (2010), "Bi-objective in fuzzy scheduling on parallel machines" Aryabhatta Journal of Mathematics & Informatics Vol. 2 No. 1, pp 149-152.
10. Garg, Rajesh Kumar & Singh, Vikram,[Dec,2010] "*Simulator for performance evaluation of process scheduling policies for embedded real-time operating systems*", Arya Bhatta Journal of Mathematics & Informatics Vol. 2, No. 2,

11. Singh T.P, Meenu Mittal and D.Gupta (2013), "A Heuristic algorithm for general weightage job scheduling under uncertain environment including transportation time", Aryabhatta Journal of Mathematics & Informatics Vol. 5 No. 2, pp 381-388.
12. Sachin, Silky Miglani and Singh T. P. [March2014] "Performance Measure Of CPU Scheduling For Real-Time Operating System In Fuzzy Environment" Yamuna J. Of Tech. & Business Research Vol.4 (1-2)pp 54.1-54.9