Performance Analysis of CPU Scheduling Algorithms with Novel OMDRRS Algorithm

Neetu Goel, Research Scholar
Department of Computer Science,
TMU, India

Dr. R. B. Garg, Ex- Professor
Delhi Universty, India

***Abstract*—CPU scheduling is one of the most primary and essential part of any operating system. It prioritizes processes to efficiently execute the user requests and help in choosing the appropriate process for execution. Round Robin (RR) & Priority Scheduling(PS) are one of the most widely used and acceptable CPU scheduling algorithm. But, its performance degrades with respect to turnaround time, waiting time & context switching with each recurrence. A New scheduling algorithm OMDRRS is developed to improve the performance of RR and priority scheduling algorithms. The new algorithm performs better than the popular existing algorithm. Drastic improvement is seen in waiting time, turnaround time, response time and context switching. Comparative analysis of Turn around Time(TAT), Waiting Time(WT), Response Time (RT) is shown with the help of ANOVA and t-test.**

***Keywords*—Round Robin; Turn-around time; Waiting Time; t-test; Anova test**

I. Introduction

CPU scheduling is similar to other types of scheduling, which have been studied over the years. CPU scheduling refers to the decision of allocating a single resource among multiple clients. It also tracks the order of allocation and duration. The primary objective of scheduling is to optimize system performance. Optimization system is considered as most de¬emed criteria by the system designers [1]. There are numerous algorithms for CPU Scheduling with distinct benefits and shortcomings. To understand and comprehend them in detail, they need to be simulated and performance indices must be studied. This paper depicts the usability of different scheduling algorithms, compare them on the basis of different performance criteria and introduce a newly designed improvised RR scheduling algorithm(OMDRRS). OMDRRS "An Optimum Multilevel Dynamic Round Robin Scheduling Algorithm" is a simulator that has been successfully implemented using VB6.0. The simulator demonstrates the algorithm behavior in opposition to a simulated mix of process loads. The results of all algorithms are compared for the scheduling criteria like turnaround time, waiting time, response time and context switch etc.

A. Scheduling Criteria

Schedulers use different scheduling criteria to enhance the performance of CPU.

Utilization/Efficiency: CPU should be best utilized by allocating the significant tasks; so that it should not be ideal.

Throughput: To increase the number of processed jobs per hour.

Turnaround time: Total time taken from submission of the process till the completion. Turnaround time should minimize the time of users who wait for the output.

Waiting time: Should be minimized as it is the total time spent in ready queue

Response Time: Is the duration after submission till the response. It should be minimal in case of interactive users.

Fairness: CPU should be unbiased and every process should get its fair time to execute.

B. Organization of the Paper

This paper is divided into four sections. Section I gives a brief introduction on the various aspects of the scheduling algorithms, the approach of the current paper and the motivational factors leading to this improvement. Section II presents the hypothesis of study, research methodology, data collection, methods used and the pseudo code of Dynamic Round Robin Algorithm (OMDRRS). In Section III, an experimental analysis and Result of our algorithm OMDRRS and its comparison with the RR algorithm, Priority scheduling and FCFS algorithm is presented. Conclusion and future scope is presented in Section IV.

II. Objective

The objective of the study is to compare the performances of different CPU scheduling Algorithms with OMDRRS.

A. Hypothesis of Study

The main or the principal instrument in the research is hypothesis. Its main factors are to suggest new experiments and observations. A number of experiments have been conducted with the deliberate objective of testing the hypothesis.

- Hypothesis 1.1

Ho1.1: There is no significant difference in Turnaround time of various CPU Scheduling Algorithms

H11.1: There is significant difference in Turnaround time of various CPU Scheduling Algorithms.

- Hypothesis 1.2

Ho1.2: There is no significant difference in Waiting time of various CPU Scheduling Algorithms

H1 1.2: There is significant difference in Waiting time of various CPU Scheduling Algorithms

- Hypothesis 1.3

Ho 1.3: There is no significant difference in Response time of various CPU Scheduling Algorithms.

H1 1.3: There is significant difference in Response time of various CPU Scheduling Algorithms.

B. *Research Methodology*

- Sample Size

The sample size is taken as 50.

C. *Data* Collection Method

Primary data is entered through the designed simulator by 50 respondents and performance of various CPU Scheduling algorithms is calculated on each of them.

D. *Choice of Respondents*

The study focused upon the performance analysis of various CPU Scheduling algorithms. The respondents were filtered on the basis of their knowledge on operating systems and CPU scheduling algorithms.

E. *Simulator*

The purpose of designing the simulator was to provide exactly the same processes to all CPU Scheduling Algorithms for performance comparison without any variations. The simulator was implemented to simulate the operations of First Come First Serve, Shortest Job First(Non Preemptive & Preemptive), Highest Priority, Round Robin and Improved Round Robin scheduling algorithms. These algorithms were implemented in order to establish a valid premise for effective comparison. Simulator has two operating modes: Manual Process Entry and Automatic Process Generator. In Automatic Process Generator system, it fetches all the active processes with burst time and assumes that all the processes were arrived at same arrival time and have no priority. Whereas in the manual entered process, user enters the burst time as per their requirement, as well as arrival time and priority of the each process. In this study respondents entered primary data in manual mode. Based on the selected scheduling algorithm, the average turnaround time, average waiting time, response time, context switch and Gantt chart were calculated. The simulation was run several times to ensure fairness to all datasets each algorithm is evaluated using Average Turn- around Time, Average Waiting Time, Response Time and Gantt chart as the performance evaluation indices.

F. *Proposed Algorithm*

The proposed algorithm[8] combines the working principle of fundamental scheduling algorithms. Dynamically Time Slice (DTS) is calculated which allocates different time quantum for each process based on priority, shortest CPU burst time and context switch avoidance time.

Step 1:

Compute the factor analysis F= Burst time * 0.2 + Arrival time * 0.3 + Priority of the process * 0.5

Step 2:

Shuffle the processes in ascending order according to the factor of each process in the ready queue (RQ) such that the head of the ready queue contains the lowest factor process based on the burst time, arrival time & priority of the process.

Step 3:

low= RQ(burst value of the first process),
high=RQ(burst value of the last process)
TQ=(low + high) / 2

Step 4:

Assign the time quantum and apply for each process say k=TQ.

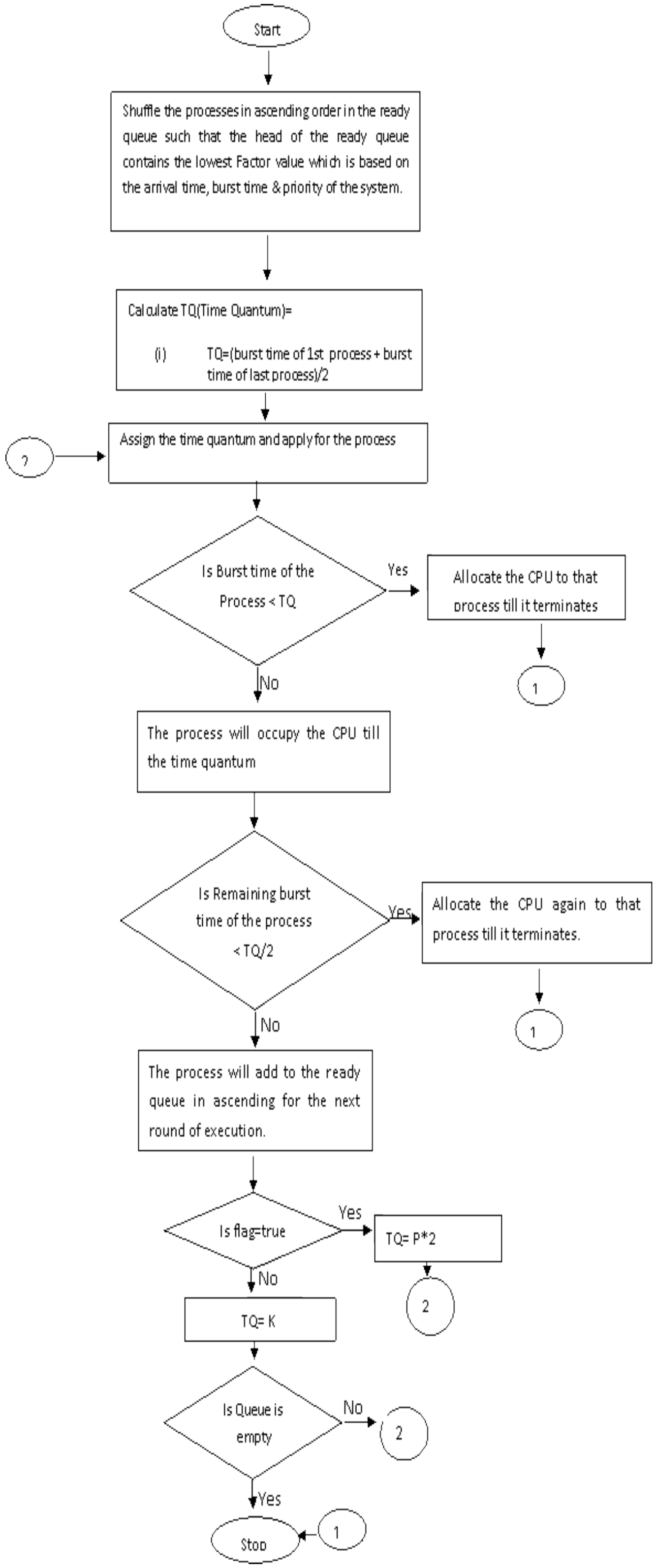
Step 5:

```
IF (burst time of the process < k)
{
    Allocate the CPU to that process till it
    terminates.
}
ELSE IF (Remaining burst time of the process <
k/2)
{
    Allocate the CPU again to that process till it
    terminates.
}
ELSE
{
    The process will occupy the CPU till the time
    quantum and it is added to the ready queue in
    ascending order according to the remaining
    burst time for the next round of execution.
    TQ= TQ *2 or TQ=TQ/2
    K=TQ
    Goto Step 4
}
```

G. *Time Complexity of OMDRRS Algorithm*

The OMDRRS algorithm would be maintaining all jobs based on the factor analysis that is ready for execution in a queue. Insertion of each job will be achieved in O(1), but deletion would require O(n) time, where n is the number of processes in the queue. Whenever a process arrives, a record for it can be inserted into the queue based on its burst time, arrival time and priority in O(n) time, where n is the number of processes in the queue. Thus, the time complexity of OMDRRS is equal to that of a typical linear sorting algorithm which is O(n). If a new task arrives it is then sorted with the remaining processes and then executed in the same way. To find a task with the lowest burst time the scheduler needs to search in the ready queue, then the order of searching would be O(n).

H. Logic Diagram of OMDRRS Algorithm



III. Data Analysis Method

Burst time, Arrival time and Priority of the processes were filled by 50 respondents in developed simulator. The simulator generated results of various scheduling algorithms and novel OMDRRS which were further analyzed and compared with anova and t-test.

- *Ethical Considerations*

The author didn't modified the existing CPU scheduling algorithm concepts and implemented them as it is in the simulator with the new version of round robin algorithm i.e. OMDRR.

A. Experimental Analysis

Fifty processes have been defined by CPU burst time, arrival time and priority of the processes. These fifty processes are scheduled in first come first serve, shortest job first, priority scheduling, round robin scheduling and also in the proposed algorithm. The turnaround time and waiting time have been calculated through simulator and the results were compared.

TABLE I. CALCULATED TURNAROUND TIME OF FCFS, SJF(NP), SJF, PS, RR AND OMDRRS ALGORITHM THROUGH SIMULATOR

PID	Burst Time	Arrival Time	PRIORTY	FCFS	SJF(NP)	SJF(P)	PS	RR	OMDRRS
1	23	0	3	23	23	428	231	637	175
2	34	5	1	523	766	766	72	827	789
3	34	3	3	321	698	698	269	825	803
4	12	6	4	660	169	147	393	571	211
5	8	8	2	744	83	62	209	305	75
6	10	4	5	358	111	90	493	477	140
7	31	1	1	54	631	631	32	765	841
8	23	2	4	120	495	495	355	668	377
9	9	3	5	272	92	71	483	465	84
10	16	6	1	593	301	279	88	551	130
11	1	5	2	547	24	9	164	177	16
12	12	8	3	756	181	159	296	583	237
13	15	9	9	778	269	247	822	622	613
14	6	6	1	666	47	26	94	265	26
15	7	2	5	127	54	33	460	67	36
16	9	3	4	348	101	80	364	464	45
17	11	5	8	369	133	123	712	527	418
18	7	8	9	763	75	54	807	344	461
19	4	9	6	822	35	14	606	356	241
20	15	5	2	455	254	232	179	492	114
21	20	6	4	613	428	406	413	739	454
22	14	3	5	263	209	187	474	471	225
23	7	2	6	97	61	40	579	74	67
24	24	1	2	78	542	542	118	653	199
25	22	5	5	407	450	450	515	702	483
26	16	8	3	736	333	311	312	591	332
27	33	5	8	440	664	664	745	830	830
28	12	3	7	339	157	135	618	475	296
29	22	6	2	688	472	472	201	735	354
30	19	9	9	841	368	346	841	758	680
31	34	5	5	489	732	732	549	829	817
32	40	9	4	818	841	841	453	841	760
33	12	1	8	90	145	101	662	400	273
34	15	3	3	235	239	217	284	449	99
35	7	5	6	574	68	47	586	227	161
36	14	3	4	249	195	173	378	463	154
37	16	5	6	385	285	263	602	524	434
38	28	2	8	155	570	570	690	775	641
39	35	6	9	648	801	801	800	833	775
40	16	8	7	704	317	295	634	599	542
41	20	2	4	175	388	366	332	672	261
42	15	3	2	287	224	202	163	442	60
43	4	2	3	179	31	6	235	44	20
44	6	3	1	327	41	20	38	96	15
45	23	5	5	546	518	518	572	717	506
46	16	8	7	720	349	327	650	607	558
47	11	2	8	190	122	112	701	427	284
48	20	5	9	567	408	386	765	729	598
49	30	2	2	220	600	600	148	771	407
50	3	5	4	577	27	10	381	188	29

TABLE II. CALCULATED WAITING TIME OF FCFS, SJF(NP), SJF, PS, RR AND OMDRRS ALGORITHM THROUGH SIMULATOR

PID	Burst Time	Arrival Time	PRIORTY	FCFS	SJF(NP)	SJF(P)	PS	RR	OMDRRS
1	23	0	3	0	0	405	208	614	152
2	34	5	1	484	727	727	33	788	750
3	34	3	3	284	661	661	232	788	766
4	12	6	4	642	151	129	375	553	193
5	8	8	2	728	67	46	193	289	59
6	10	4	5	344	97	76	479	463	126
7	31	1	1	22	599	599	0	733	809
8	23	2	4	95	470	470	330	643	352
9	9	3	5	260	80	59	471	453	72
10	16	6	1	571	279	257	66	529	108
11	1	5	2	541	18	3	158	171	10
12	12	8	3	736	161	139	276	563	217
13	15	9	9	754	245	223	798	598	589
14	6	6	1	654	35	14	82	253	14
15	7	2	5	118	45	24	451	58	27
16	9	3	4	336	89	68	352	452	33
17	11	5	8	353	117	107	696	511	402
18	7	8	9	748	60	39	792	329	446
19	4	9	6	809	22	1	593	343	228
20	15	5	2	435	234	212	159	472	94
21	20	6	4	587	402	380	387	713	428
22	14	3	5	246	192	170	457	454	208
23	7	2	6	88	52	31	570	65	58
24	24	1	2	53	517	517	93	628	174
25	22	5	5	380	423	423	488	675	456
26	16	8	3	712	309	287	288	567	308
27	33	5	8	402	626	626	707	792	792
28	12	3	7	324	142	120	603	460	281
29	22	6	2	660	444	444	173	707	326
30	19	9	9	813	340	318	813	730	652
31	34	5	5	450	693	693	510	790	778
32	40	9	4	769	792	792	404	792	711
33	12	1	8	77	132	88	649	387	260
34	15	3	3	217	221	199	266	431	81
35	7	5	6	562	56	35	574	215	149
36	14	3	4	232	178	156	361	446	137
37	16	5	6	364	264	242	581	503	413
38	28	2	8	125	540	540	660	745	611
39	35	6	9	607	760	760	759	792	734
40	16	8	7	680	293	271	610	575	518
41	20	2	4	153	366	344	310	650	239
42	15	3	2	269	206	184	145	424	42
43	4	2	3	173	25	0	229	38	14
44	6	3	1	318	32	11	29	87	6
45	23	5	5	518	490	490	544	689	478
46	16	8	7	696	325	303	626	583	534
47	11	2	8	177	109	99	688	414	271
48	20	5	9	542	383	361	740	704	573
49	30	2	2	188	568	568	116	739	375
50	3	5	4	569	19	2	373	180	21

B. *Results of Analysis*

In order to analyze the difference in performance of the various CPU scheduling algorithms, ANOVA and t-test are used and the following results are obtained as follows:

TABLE III. COMPARISON OF TURNAROUND TIME IN FCFS, SJF(NP), SJF, PS, RR AND OMDRRS ALGORITHM

ANOVA						
Source of Variation	*SS*	*Df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	2164333	5	432866.7	7.398212	1.49E-06	2.244703
Within Groups	17201834	294	58509.64			
Total	19366167	299				

In the above table of ANOVA turnaround time is significantly differ for different scheduling algorithms, because the computed value (F) is greater than tabulated value (F crit) at 5% level of significance. Hence H0 1.1 is rejected. Rejecting null hypothesis proves that there is significant difference in Turnaround time of various CPU Scheduling Algorithms.

TABLE IV. COMPARISON OF WAITING TIME IN FCFS, SJF(NP), SJF, PS, RR AND OMDRRS ALGORITHM

ANOVA						
Source of Variation	*SS*	*Df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	2164333	5	432866.7	7.808022	6.44E-07	2.244703
Within Groups	16298981	294	55438.71			
Total	18463314	299				

In the above table of ANOVA shows that overall waiting time is significant. Since the computed value (F) is greater than tabulated value (F crit) at 5% level of significance. Hence H01.2 is rejected. Rejecting null hypothesis proves that there is significant difference in waiting time of various CPU Scheduling Algorithms.

TABLE V. COMPARISON OF RESPONSE TIME IN FCFS, SJF(NP), SJF, PS, RR AND OMDRRS ALGORITHM

ANOVA						
Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	2064121	5	412824.3	8.606909	1.26E-07	2.244703
Within Groups	14101500	294	47964.29			
Total	16165622	299				

In the above table of ANOVA depicts that overall response time is significant. Since the computed value (F) is greater than tabulated value (F crit) at 5% level of significance. Hence H01.3 is rejected. Rejecting null hypothesis proves that there is significant difference in response time of various CPU Scheduling Algorithms.

TABLE VI. COMPARISON OF TURN AROUND TIME BETWEEN PRIORITY SCHEDULING AND DYNAMIC ROUND ROBIN ALGORITHM WITH THE HELP OF T-TEST

	PS	*OMDRRS*
Mean	431.36	342.92
Variance	55263.05	69012.8098
Observations	50	50
Pearson Correlation	0.397149	
Hypothesized Mean Difference	0	
Df	49	
t Stat	2.280126	
P(T<=t) one-tail	0.013495	
t Critical one-tail	1.676551	
P(T<=t) two-tail	0.026989	
t Critical two-tail	2.009575	

At 5% level of significance h01.1 is rejected hence there is a significant difference between turnaround time of priority scheduling and proposed dynamic round robin algorithm(OMDRRS).

TABLE VII. COMPARISON OF TURN AROUND TIME BETWEEN ROUND ROBIN SCHEDULING AND DYNAMIC ROUND ROBIN ALGORITHM WITH THE HELP OF T-TEST

	RR	*OMDRRS*
Mean	532.98	342.92
Variance	50676.26	69012.81
Observations	50	50
Pearson Correlation	0.810487	
Hypothesized Mean Difference	0	
Df	49	
t Stat	8.706293	
P(T<=t) one-tail	8.11E-12	
t Critical one-tail	1.676551	
P(T<=t) two-tail	1.62E-11	
t Critical two-tail	2.009575	

At 5% level of significance h01.1 is rejected hence there is a significant difference between turnaround time of round robin algorithm and dynamic round robin algorithm(OMDRRS)

TABLE VIII. COMPARISON OF WAITING TIME BETWEEN PRIORITY SCHEDULING AND DYNAMIC ROUND ROBIN ALGORITHM WITH THE HELP OF T-TEST

	PS	*OMDRRS*
Mean	409.94	321.5
Variance	54987.81	64623.32
Observations	50	50
Pearson Correlation	0.372315	
Hypothesized Mean Difference	0	
Df	49	
t Stat	2.280126	
P(T<=t) one-tail	0.013495	
t Critical one-tail	1.676551	
P(T<=t) two-tail	0.026989	
t Critical two-tail	2.009575	

At 5% level of significance h11.2 is rejected hence there is a significant difference between waiting time of Priority Scheduling and Dynamic Round Robin Algorithm. Above table proves that Proposed algorithm OMDRRS is better than the Priority scheduling algorithm in terms of waiting time.

TABLE IX. COMPARISON OF WAITING TIME BETWEEN ROUND ROBIN SCHEDULING AND DYNAMIC ROUND ROBIN ALGORITHM WITH THE HELP OF T-TEST

	RR	*OMDRRS*
Mean	511.56	321.5
Variance	46738.5	64623.32
Observations	50	50
Pearson Correlation	0.796369	
Hypothesized Mean Difference	0	
Df	49	
t Stat	8.706293	
P(T<=t) one-tail	8.11E-12	
t Critical one-tail	1.676551	
P(T<=t) two-tail	1.62E-11	
t Critical two-tail	2.009575	

At 5% level of significance h11.2 is rejected hence there is a significant difference between waiting time of Round Robin Scheduling and Dynamic Round Robin Algorithm. Hence OMDRRS is also better than the round robin scheduling algorithm.

TABLE X. COMPARISON OF TURNAROUND TIME BETWEEN PRIORITY SCHEDULING AND DYNAMIC ROUND ROBIN ALGORITHM WITH THE HELP OF T-TEST

	PS	*OMDRRS*
Mean	431.36	342.92
Variance	55263.05	69012.8098
Observations	50	50
Pearson Correlation	0.397149	
Hypothesized Mean Difference	0	
Df	49	
t Stat	2.280126	
P(T<=t) one-tail	0.013495	
t Critical one-tail	1.676551	
P(T<=t) two-tail	0.026989	
t Critical two-tail	2.009575	

At 5% level of significance h11.1 is rejected hence there is a significant difference between turnaround time of Priority Scheduling and Dynamic Round Robin Algorithm. Above table proves that our proposed algorithm OMDRRS is better than the Priority scheduling algorithm in terms of turnaround time.

TABLE XI. COMPARISON OF TURNAROUND TIME BETWEEN ROUND ROBIN SCHEDULING AND DYNAMIC ROUND ROBIN ALGORITHM WITH THE HELP OF T-TEST

	RR	*OMDRRS*
Mean	532.98	342.92
Variance	50676.26	69012.81
Observations	50	50
Pearson Correlation	0.810487	
Hypothesized Mean Difference	0	
Df	49	
t Stat	8.706293	
P(T<=t) one-tail	8.11E-12	
t Critical one-tail	1.676551	
P(T<=t) two-tail	1.62E-11	
t Critical two-tail	2.009575	

At 5% level of significance h11.1 is rejected hence there is a significant difference between turnaround time of Round Robin Scheduling and Dynamic Round Robin Algorithm. Above table proves that our proposed algorithm OMDRRS is better than the Round Robin scheduling algorithm in terms of turnaround time.

TABLE XII. DETERMINISTIC STATISTICS OF VARIOUS SCHEDULING ALGORITHM

	FCFS	*PS*	*RR*	*OMDRRS*
Mean	438.72	431.36	532.98	342.92
Standard Error	34.32485908	33.24546629	31.83591208	37.15179936
Median	423.5	433	561	278.5
Standard Deviation	242.7134062	235.0809465	225.1138932	262.7028926
Confidence Level(95.0%)	68.97838553	66.80926454	63.97665936	74.65933461
95% confidence interval for execution of TAT	(335.75, 541.69)	(331.62, 531.09)	(437.47, 628.49)	(231.46, 454.37)

The confidence interval for execution time of turnaround time of various scheduling algorithms at 95% shows that the time is varying between 231.46 nanoseconds to 454.37 nanoseconds that is 95% of the jobs will complete its execution within this time frame.

IV. CONCLUSION AND FUTURE SCOPE

This paper statistically analyzes and compares various scheduling algorithms with proposed algorithm. It analysis the performance of various CPU scheduling algorithms with respect to dynamic time slice concept in Round Robin CPU scheduling. The suggested algorithm and other scheduling algorithm are executed on the simulator and evaluated on Anova and t-test. The Results clearly depicts that OMDRRS performs better than existing algorithms on the basis of Anova and t-test results comparative analysis of Turnaround time, waiting time & context switch clearly shows that OMDRRS gives much better turnaround time with very less waiting time. Deterministic statistics shows that the confidence index is improved in the case of OMDRRS algorithm. Results concludes that the proposed algorithm is superior then commonly used algorithm with less waiting response time, less turnaround time and context switching; thereby reducing the overhead and results in saving lots of memory space. Taking the base of proposed algorithm more improvement can be made the future.

REFERENCES

[1] A.S. Tanenbaun, Modern Operating Systems.3rd Edn, Prentice Hall, ISBN:13: 9780136006633, pp: 1104, 2008.

[2] Rakesh Mohanty, H.S. Behera and et. al, Design and Performance Evaluation of a new proposed Shortest Remaining Burst Round Robin(SRBRR) scheduling algorithm, Proceedings of the International Symposium on Computer Engineering and Technology(ISCET), March, 2010.

[3] Behera S. H., Mohanty R., Sahu S. & Bhoi K. S.; Design and Performance Evaluation of Multi Cyclic Round Robin (MCRR) Algorithm Using Dynamic Time Quantum, Journal of Global Research in Computer Science : Volume 2, No. 2, February 2011, ISSN-2229-371X

[4] Behera S. H., Patel S. & Panda B. ; A New Dynamic Round Robin and SRTN Algorithm with Variable Original Time Slice and Intelligent Time Slice for Soft Real Time Systems, International Journal of Computer Applications (0975 – 8887) Volume 16– No.1, February 2011

[5] S. H. Nazleeni, H. M. A. Anang, M. H. Hasan, A. A. Izzatdin, and M. W. Wirdhayu, "Time comparative simulator for distributed process scheduling algorithms," World Academy of Science, Engineering and Technology 19, pp.84-89,2006

[6] Silberschatz, A. P.B. Galvin and G. Gagne (2012), Operating System Concepts, 8th edition, Wiley India.

[7] Goel N., Garg R. B., "A Simulation of an Optimum Multilevel Dynamic Round Robin Scheduling Algorithm", International Journal of Computer Applications , ISSN 0975 – 8887, Volume 76– No.7, August 2013

[8] Goel N., Garg R. B., "Improvised Optimum Multilevel Dynamic Round Robin Algorithm for Optimizing CPU Scheduling", International Journal of Computer Applications , ISSN 0975 – 8887, Volume – No., August 2015.

[9] Goel N., Garg R. B., "A Comparative Study of CPU Scheduling Algorithms", International Journal of Graphics Image Processing (IJGIP), Vol 2 issue 4, Dec-2012

[10] Stallings, W.: Operating Systems Internals and Design Principles, 5th edition, Prentice Hall, (2004).

[11] Rajput I., Gupta D., "A Priority based Round Robin CPU Sche duling Algorithm for Real Time Systems", IJIET, Vol. 1 Issue 3 Oct. 2012

[12] http://vip.cs.utsa.edu/classes/cs3733s2004/notes/ps/runps.html

[13] http://www.codeplex.com/cpuss

[14] http://www.ats.ucla.edu/stat/spss/whatstat/whatstat.htm

[15] http://www.real-statistics.com/students-t-distribution/paired-sample-t-test/