



PERFORMANCE ANALYSIS OF CPU SCHEDULING ALGORITHMS WITH DFRRS ALGORITHM

C. Jayavarthini, Angsuman Chattopadhyay, Pratik Banerjee and Shounak Dutta

Department of Computer Software Engineering, SRM University, Kattankulathur, India

ABSTRACT

CPU scheduling is an integral part of any operating system and defines the basic functionality of an operating system. A scheduling algorithm is in-tended to execute user and system requests with the highest efficiency possible. The algorithm is responsible for analysing the processes, choosing and dispatching the most appropriate process for execution. Two of the most commonly used scheduling algorithms are the round robin (RR) algorithm and the priority scheduling algorithm. However, they both have their own pros and cons with respect to the qualities of service provided by a good scheduling algorithm. A new scheduling algorithm DFRRS (Dynamically Factored Round Robin Scheduling) has been developed to improve the performance of Round Robin Scheduler by incorporating the features of priority scheduling and SJF algorithm. A comparative analysis of turnaround and waiting Time is shown with the help of bar graphs (Histograms).

Keywords: operating system, algorithm, scheduling, efficiency, turnaround time, waiting time, context switching, priority scheduling, round robin scheduling, histograms.

1. INTRODUCTION

CPU scheduling can be defined as allocation and de-allocation of the resources available to the operating system among various outstanding processes and the decisions surrounding it. The order in which a process is allocated and its duration is also determined by the algorithm. A scheduling algorithm's primary objective is to optimize the overall performance of a system while ensuring fairness to all processes. Optimizing a system is what system designers want. There are numerous algorithms for CPU Scheduling, each with their own benefits and shortcomings. A comparative study of these schedulers is needed to fully understand the relative performance of each. A newly designed and improved scheduling algorithm is introduced in this paper Dynamically Factored Round Robin Scheduling (DFRRS). The algorithm was developed and simulated in Java 8. The simulator is used to demonstrate how the algorithm behaves, in comparison to the other existing scheduling algorithms on scheduling parameters such as waiting time, turnaround time, context switch, et cetera.

2. PARAMETERS OF SCHEDULING

A scheduler's performance is evaluated based on the parameters defined below.

Waiting Time: The amount of time from arrival that a process is idle in the system.

Throughput: Defined as the number of processes completed per unit time.

Fairness: The ability of a system to assign resources in an unbiased fashion.

Turnaround Time: The amount of time spent by a process in the system.

3. AIM OF THE STUDY

The aim of the study is to qualitatively compare

the mainstream scheduling algorithms with the newly proposed DFRRS Algorithm

a) Sample size

The attributes of 50 different processes have been taken as the sample pool for conducting the study.

b) Sample generation

The sample has been collected from a pre-existing source, which were already scheduled by the OMDRRS simulator. This was done to ensure the comparative study could be carried out with the highest possible accuracy.

c) Simulator

The algorithm was coded into a simulator to schedule and calculate the turnaround time, average turnaround time and the waiting time of each process given to it. The simulator computed only the DFRRS algorithm scheme along with the four basic scheduling algorithms First Come First Server (FCFS), Shortest Job First (SJF Non-pre-emptive and pre-emptive), Priority Scheduling (PS) and Round Robin Scheduling. The data for the OMDRRS algorithm was taken from a reliable source. The simulator operates by taking an active input from the users a process ID, arrival time, burst time and priority are taken. Based on the inputted data, the simulator would simulate the functionality of the DFRRS algorithm in a real time scenario. The output was given as a tabular representation of each process and its turnaround time, and the total and average turnaround time, and the total and average waiting time.

d) Proposed algorithm

The algorithm's time slice has been calculated dynamically and it thus allocates a different time quantum every time a process is dispatched. This allows the algorithm to change the value of the time quanta after every cycle, to ensure the best performance. The factor analysis is calculated by multiplying a constant factor to



remaining burst time and priority where the remaining burst time has a higher weightage. The trade-off between the priority of the process and the remaining burst time of the process enables an improved average turn-around time, ensuring that performance is not negatively affected based only on the priority. The calculated factor is denoted by F.

- 1) Step 1: Each process as it arrives is added to the ready queue and the factors are calculated for each.
- 2) Step 2: The ready queue is sorted as per the factor, and a dynamic time slice is calculated according to the formula, $TQ = (1st \text{ process in the ready queue} + last \text{ process in the ready queue})/2$
- 3) Step 3: IF (Remaining burst time of the process $< TQ$) The process is allocated the CPU time and other resources till it terminates. ELSE IF (Remaining burst time of the process $< TQ/2$). The process is again allocated the CPU time and other resources till it terminates.
- 4) Step 4: Go to step 1.

e) Time complexity of DFRRS algorithm

The DFRRS algorithm maintains all processes that are present in the ready queue according to their corresponding dynamically calculated factor in an ascending order. This ensures that at any given time the process to be dispatched is at the top of the ready queue, which can be retrieved in $O(1)$ complexity. The insertion into the queue is achieved in $O(n)$, where n denotes the number of processes that have arrived and are yet to be added into the ready queue. The dynamically calculated factor is sorted using a general linear sorting algorithm which gives it a $O(n)$ complexity. Finally, the deletion of a process from the ready queue after completion is achieved with $O(n)$ complexity. Thus the overall time complexity of the proposed DFRRS algorithm is $O(n)$.

4. PERFORMANCE ANALYSIS

The process attributes were taken from a previous scientific study and fed into the simulator. The simulator generated the total and average turnaround and waiting time for the various scheduling algorithms along with the DFRRS algorithm which was further compared by plotting a process by process bar chart and a Histogram for the existing and the proposed algorithm.

a) Consideration

The existing CPU scheduling algorithm concepts were not modified and were implemented as they were along with the proposed algorithm i.e. DFRRS in the simulator.

b) Experimentation

Fifty processes along with their attributes were fed into the DFRRS simulator. These processes have been scheduled using the existing algorithms and the proposed algorithm. The turnaround time and waiting time was calculated via the simulator and the results were

compared.

c) Result of analysis

Given below are the bar graphs and histograms representing the various scheduling algorithms.

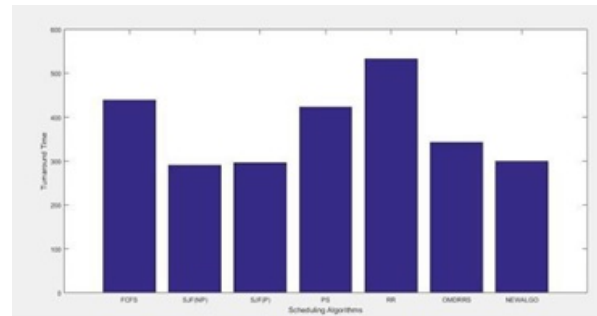


Figure-1. Average turnaround time for each scheduling algorithm.

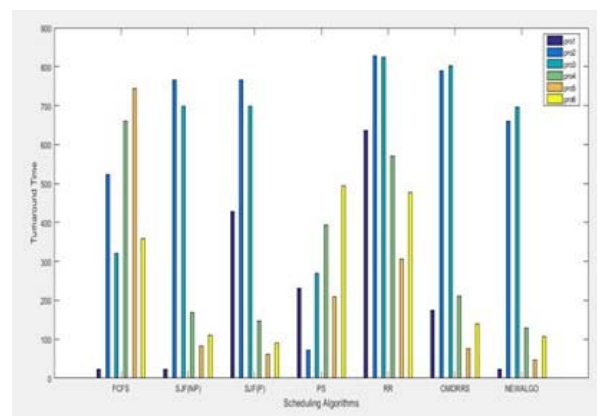


Figure-2. Process-wise study of turnaround time for each scheduling algorithm.

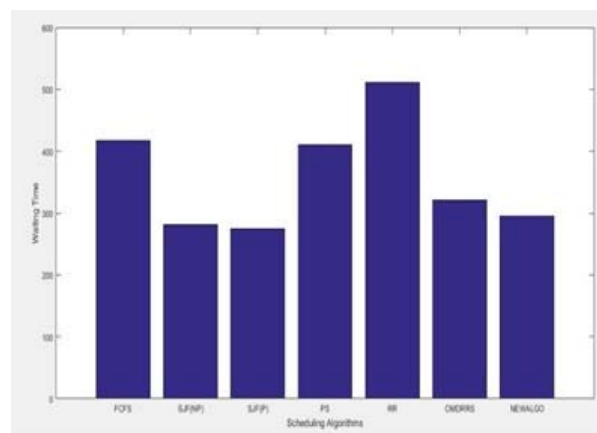


Figure-3. Average waiting time for each scheduling algorithm.



www.arpnjournals.com

Table-1. Turnaround time.

PID	Burst Time	Arrival Time	Priority	FCFS	SJF(NP)	SJF(P)	PS	RR	OMDRRS	DFRRS
1	23	0	3	23	23	428	231	637	175	23
2	34	5	1	523	766	766	72	827	789	660
3	34	3	3	321	698	698	269	825	803	696
4	12	6	4	660	169	147	393	571	211	129
5	8	8	2	744	83	62	209	305	75	47
6	10	4	5	358	111	90	493	477	140	107
7	31	1	1	54	631	631	32	765	841	602
8	23	2	4	120	495	495	355	668	377	497
9	9	3	5	272	92	71	483	465	84	91
10	16	6	1	593	301	279	88	551	130	197
11	1	5	2	547	24	9	164	177	16	19
12	12	8	3	756	181	159	296	583	237	115
13	15	9	9	778	269	247	822	622	613	340
14	6	6	1	666	47	26	94	265	26	37
15	7	2	5	127	54	33	460	67	36	60
16	9	3	4	348	101	80	364	464	45	82
17	11	5	8	369	133	123	712	527	418	171
18	7	8	9	763	75	54	807	344	461	93
19	4	9	6	822	35	14	606	356	241	38
20	15	5	2	455	254	232	179	492	114	160
21	20	6	4	613	428	406	413	739	454	383
22	14	3	5	263	209	187	474	471	225	267
23	7	2	6	97	61	40	579	74	67	67
24	24	1	2	78	542	542	118	653	199	453
25	22	5	5	407	450	450	515	702	483	471
26	16	8	3	736	333	311	312	591	332	278
27	33	5	8	440	664	664	745	830	830	761
28	12	3	7	339	157	135	618	475	296	226
29	22	6	2	688	472	472	201	735	354	405
30	19	9	9	841	368	346	841	758	680	421
31	34	5	5	489	732	732	549	829	817	728
32	40	9	4	818	841	841	453	841	760	832
33	12	1	8	90	145	101	662	400	273	255
34	15	3	3	235	239	217	284	449	99	241
35	7	5	6	574	68	47	586	227	161	71
36	14	3	4	249	195	173	378	463	154	214
37	16	5	6	385	285	263	602	524	434	297
38	28	2	8	155	570	570	690	775	641	629
39	35	6	9	648	801	801	800	833	775	795
40	16	8	7	704	317	295	634	599	542	326
41	20	2	4	175	388	366	332	672	261	367
42	15	3	2	287	224	202	163	442	60	147
43	4	2	3	179	31	6	235	44	20	29
44	6	3	1	327	41	20	38	96	15	34
45	23	5	5	546	518	518	572	717	506	537
46	16	8	7	720	349	327	650	607	558	310
47	11	2	8	190	122	112	701	427	284	185
48	20	5	9	567	408	386	765	729	598	514
49	30	2	2	220	600	600	148	771	407	570
50	3	5	4	577	27	10	381	188	29	22



www.arpnjournals.com

Table-2. Waiting time.

PID	Burst Time	Arrival Time	PRIORITY	FCFS	SJF(NP) SJF(P)	PS	RR		OMDRRS	DFRRS
1	23	0	3	0	0	405	208	614	152	23
2	34	5	1	484	727	727	33	788	750	655
3	34	3	3	284	661	661	232	788	766	693
4	12	6	4	642	151	129	375	553	193	123
5	8	8	2	728	67	46	193	289	59	39
6	10	4	5	344	97	76	479	463	126	103
7	31	1	1	22	599	599	0	733	809	601
8	23	2	4	95	470	470	330	643	352	495
9	9	3	5	260	80	59	471	453	72	88
10	16	6	1	571	279	257	66	529	108	191
11	1	5	2	541	18	3	158	171	10	14
12	12	8	3	736	161	139	276	563	217	107
13	15	9	9	754	245	223	798	598	589	331
14	6	6	1	654	35	14	82	253	14	31
15	7	2	5	118	45	24	451	58	27	58
16	9	3	4	336	89	68	352	452	33	79
17	11	5	8	353	117	107	696	511	402	166
18	7	8	9	748	60	39	792	329	446	85
19	4	9	6	809	22	1	593	343	228	29
20	15	5	2	435	234	212	159	472	94	155
21	20	6	4	587	402	380	387	713	428	377
22	14	3	5	246	192	170	457	454	208	264
23	7	2	6	88	52	31	570	65	58	65
24	24	1	2	53	517	517	93	628	174	452
25	22	5	5	380	423	423	488	675	456	466
26	16	8	3	712	309	287	288	567	308	270
27	33	5	8	402	626	626	707	792	792	756
28	12	3	7	324	142	120	603	460	281	223
29	22	6	2	660	444	444	173	707	326	399
30	19	9	9	813	340	318	813	730	652	412
31	34	5	5	450	693	693	510	790	778	723
32	40	9	4	769	792	792	404	792	711	823
33	12	1	8	77	132	88	649	387	260	254
34	15	3	3	217	221	199	266	431	81	238
35	7	5	6	562	56	35	574	215	149	66
36	14	3	4	232	178	156	361	446	137	211
37	16	5	6	364	264	242	581	503	413	292
38	28	2	8	125	540	540	660	745	611	627
39	35	6	9	607	760	760	759	792	734	789
40	16	8	7	680	293	271	610	575	518	318
41	20	2	4	153	366	344	310	650	239	365
42	15	3	2	269	206	184	145	424	42	144
43	4	2	3	173	25	0	229	38	14	27
44	6	3	1	318	32	11	29	87	6	31
45	23	5	5	518	490	490	544	689	478	532
46	16	8	7	696	325	303	626	583	534	302
47	11	2	8	177	109	99	688	414	271	183
48	20	5	9	542	383	361	740	704	573	509
49	30	2	2	188	568	568	116	739	375	568
50	3	5	4	569	19	2	373	180	21	17

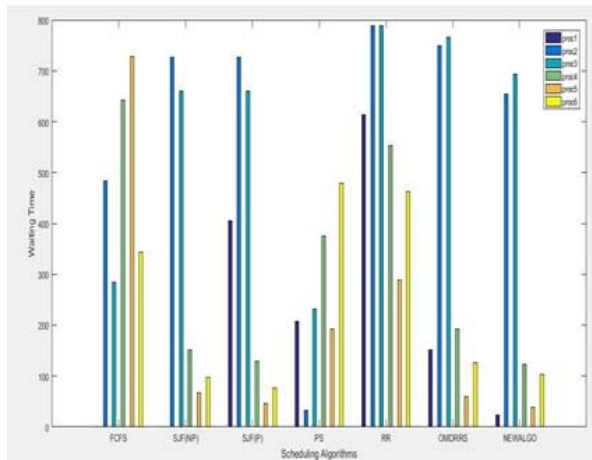


Figure-4. Process-wise study of turnaround time for each scheduling algorithm.

5. CONCLUSIONS

This paper compares and analyses the four basic scheduling algorithms with the newly developed DFRRS algorithm. The analysis was based on the performance of the basic CPU scheduling algorithms and also the OMDRRS algorithm with respect to the factor calculation and dynamic time slice concept. The turnaround time and the average waiting time is calculated for each of the existing algorithms and also of the suggested algorithm and a graph is plotted for the same to compare the results. The graphs clearly depict that the suggested DFRRS algorithm has lower values than those of the existing algorithms in terms of both turnaround time and waiting time. Thus the conclusion can be drawn that the DFRRS algorithm is better than the commonly used ones on all qualitative sectors – including throughput and fairness. In the coming future more improvements can be made to this algorithm.

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] Dr. R.B. Garg, Neetu Goel: Performance Analysis of CPU Scheduling Algorithms
- [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., “Improved 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), Vol2 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 Scheduling Algorithm for Real Time Systems”, IJIT, Vol. 1 Issue 3 Oct. 2012
- [12] <http://vip.cs.utsa.edu/classes/cs3733s2004/notes/ps/ru nps.html>
- [13] <http://www.codeplex.com/cpuss>
- [14] <http://www.ats.ucla.edu/stat/spss/whatstat/whatstat.htm>