Efficient Dual Nature Round Robin CPU Scheduling Algorithm: A Comparative Analysis

Sajida Fayyaz[1], Hafiz Ali Hamza[2], Saira Moin U Din[3] and Iqra[4]
[1-4]Department of Computer Science, University of Lahore (Sargodha Campus)
[1]sajida.fayyaz@cs.uol.edu.pk, [2]alihamzagondal12@yahoo.com, [3]saira.moin@cs.uol.edu.pk, [4]iqraakram213@gmail.com

***Abstract*– Operating system is the soul computer which helps the CPU in scheduling of all the processes that are performed. There are many CPU scheduling algorithms including the First Come First Serve (FCFS), Shortest Job First (SJF), Round Robin (RR) and SJF with priority. All these above mention algorithms are now in use for providing satisfactory results regarding CPU utilization. Using all these algorithms we have proposed the new algorithm that is named as EDNRR (Efficient Dual Nature round Robin). Our study used the concept of RR, Improved Shortest Remaining Burst Round Robin (ISRBRR) and Shortest Remaining Burst Round Robin (SRBRR). The objective of proposed algorithm is to reduce the starvation, total turnaround time and wait time using RR by setting the time quantum in the increasing order and decreasing order. The performance of CPU is based on the scheduling of processes, according to the calculation results; the wait time of processor is reduced up to 20%.**

***Keywords*– Scheduling Algorithm, Operating System, ENDRR (Efficient Dual Nature Round Robin), SRBRR (Small Remaining Burst Round Robin), RR (Round Robin), FCFS and SJF (Shortest Job First)**

I. INTRODUCTION

An operating system [12] acts as the intermediate between the user and the computer hardware. Different application programs are run in the user mode and system software runs on the Kernel mode.

Both modes interact with each other and execute various programs. It allows the processor to perform operation on various processes by scheduling the tasks. It's worth mentioning here that operating system act as the soul of our computer. In order to minimize the time and to increase the efficiency of the CPU scheduling that increases its performance we have tried to make the waiting time and the turnaround time of the processes execution as smaller as possible. FCFS [14] Algorithm has the disadvantage that the processes having large burst time with early arrival use more CPU and other processes suffer from starvation. To avoid this in some cases SJF [15] algorithm is used. RR is the effective algorithm because its time quantum and CPU utilization is maximum. Using the static time quantum in RR, it has some disadvantages. During context switching the CPU will take more time to schedule the processes and will face starvation. In context switching some time waste because a process will move to and fro in a queue in order get the processor by resuming the execution of recently processed process. The good scheduling algorithm has increased CPU usage, increased throughput, decrease turnaround time and minimize the waiting time.

The new proposed algorithm uses the SRBRR [9] and ISRBRR [19]. The time quantum in new algorithm is given in ascending order and descending order [18] - [20]. This will cause decrease in the average wait time of a processor and the total turnaround time also.

II. RELATED WORK

From the previous years we are learning on an idea of the CPU scheduling. We have come to know that CPU utilization is made maximum and the performance rate is increased using the different scheduling criteria. SJF and FCFS are most commonly used algorithms. A process is assigned a fixed amount of time that is waiting in the queue known as the ready queue, only in very first iteration and then we make use of SJF algorithm to pick up the next in coming process in an Improved Round Robin Scheduling Algorithm for the processor's scheduling [1,4]. Time quantum adjustment is made in a continuous manner according to the processes B.T in the Self-Arranged Time Quantum in RR Algorithm [2]. The RR algorithm use the fixed time period to schedule processes. The all processes in ready queue are given the fixed quantum that is adjusted, then processes are checked and those having small burst time are then processed. The RR algorithm uses the technique of Shortest Job First algorithm [3].

A) The Round Robin Algorithm

This algorithm [4] is defined in the Steps listed below:

- System contains processor and processor maintains the scheduler that keeps a list of processes that has been swapped out in ready queue (RQ) which is blocked.
- The process in process control block (PCB), whose execution is finished, will be excluded from data structure of scheduling queue and the PCB of new processes is saved at the end of the RQ.
- PCB present at the top most position of RQ which always selected by this algorithm.

- When the process in a running state complete its time period it goes down at the end of RQ that contains all completed processes.
- The event handler performs the following function:
  - If there is a process and it request for an input, output and swapping, the PCB of that particular process is aloof from the RQ in a swapped out list.
  - At the same time when a process made any I/O request, the processor will have to wait, and that leads to several issues [5] of RR like:

Fixed ratio of time.

Amplified quantity of wait time.

Increase the context switching time

Output will be truncated.

We can prove from the above discussion that the RR algorithm is not consider as good for the scheduling purposes in the real time system. The proposed algorithm will solve this problem to some extent.

B) The scheduling Terms and Algorithms used for scheduling:

Scheduling Terms

CPU scheduling is the task that is performed by our operating system. The scheduling of a process involve few important terms that are listed below with their associated equations [12], [13]:

Utilization of CPU: A term defined as the phenomenon in which our CPU is in a busy state or performing executions.

Throughput: The work done by the processor of a system using specific time period to complete the task. If the number of processes are executed by the processor are larger in amount than the work to be done will be more efficient.

Wait Time: The time duration in which a process can wait for the other process to be completed and release the resources is called wait time. It is given by [12]:

$$\textit{Wait Time (W.T)} = (1^{st}\ start) - (Arrival) + (2^{nd}\ start) - (1^{st}\ leave) \ldots\ (nth\ start) - (n\text{-}1\ leave) \quad (1)$$

$$\textit{Avg. Wait time} = \sum W.T/n \quad (2)$$

Where n is total number of process to be schedule.

Turnaround time: The time when CPU is assigned to a process and the time it gets to be complete the task is called the turnaround time. It is given by:

$$\textit{Turnaround Time} = \textit{last finish} - \textit{arrival time} \quad (3)$$

$$\textit{Avg. TA time} = \sum \textit{TA time} / n \quad (4)$$

Where n is total number of process to be scheduled.

Responsive time: The time taken by a processor when we submit request and the very first outcome is generated in return.

$$\textit{Response time} = \textit{First start} - \textit{Arrival} \quad (5)$$

Priority: The kind of preference that we will give to different processes. Higher the number low will be the priority and vice versa.

Fairness: The processor of a system is allowed to give equal time to the processes to get executed. This prevents starvation.

.

A Summary of Surviving Scheduling Algorithms of CPU: First Comes First Serve (FCFS) Scheduling algorithm:

One of the simple algorithms that are used for scheduling purpose is FCFS [14]. The processor is given to that process which is first appeared in RQ will be served first by the processor as it is clear from the above mentioned name of this algorithm. The process that reserves the CPU will leave the processor only when it will complete its task. Our CPU will remain occupied by that process for long time if the burst time (BT) of the CPU is long. The processes that have small BT will wait for long time. One advantage of using this algorithm is that it will reduce the situation of over heading due to its non-preemptive nature and no context switching.

Consider the following example:

Table I: FCFS Parameters

Processes	Start Time of processes	Burst Time of processes
P1	2	7
P2	8	10
P3	7	11
P4	3	5
P5	0	4

From the above Table I, it is given:

Wait time = 38

Average wait time = 7.6

Turnaround time = 75

Average turnaround time = 15

Shortest Job First (SJF) Scheduling Algorithm:

A kind of algorithm [15] in which the process that has the smallest BT is executed first and then the processor will check which process has the small burst time and allocate that process a CPU, the processor next. If there are processes that have same smallest burst time then the CPU have the choice to move on with the existing process or allocate the processor to the next one [6]. SJF can serve both in a primitive way and in a non-preemptive way of scheduling.

Average of turnaround time and wait time of process is condensed as compared to FCFS. But in SJF the processes with short burst time are given more importance and this will create the condition of over heading. It will choose processes as soon as the previous one completed, the next one will be ready in a ready queue for execution. It's good because it will save small processes from being remaining in a waiting state in order to wait until the large processes completed and the phenomenon of starvation occur.

Let solve the example of it:

Table II: SJF Parameters

Processes	Start Time of process	Burst Time of process
a	6	7
b	4	10
c	3	5
d	0	4

From the above Table II, it is given:
Wait time = 19
Average wait time = 4.75
Turnaround time = 45
Average turnaround time = 11.25

Priority Based Scheduling of CPU using RR:

It is a kind of CPU scheduling [15] in which process are executed on the base of their priority. Those processes having high priority are processed by the processor first and then the low priority processes are processed.
If there is a condition that multiple processes have same priority then they are arranged in the sequence of FCFS algorithm [7], [14]. There are two kind of Priority Scheduling:

- Primitive
- Non primitive

Consider the example for better understanding:

Table III: Priority RR parameters

Processes	Start time	Burst time	Priority
w	4	3	1
x	5	4	2
y	3	7	3
z	0	5	0

From the above table, it is given:
Wait time = 13
Average wait time = 3.25
Turnaround time = 32
Average turnaround time = 8

Round Robin (RR) Scheduling of CPU with SJF:

Round Robin is the algorithm [19] in which the fixed time quantum is given and it is primitive. CPU move from one process to other according to specifically assigned time period [8].
Round Robin strategy performance and efficiency depends upon the selection of the time quantum. It works more efficiently in combination with the SJF [16].

Table IV: RR with SJF

Processes	Start Time	Burst Time
R	0	11
f	7	12
t	5	3

From the above table, it is given:
Wait time = 12
Average wait time =4
Turnaround time = 38
Average turnaround time =12.666

III. PROPOSED METHOD

An overview of the idea

CPU scheduling is the one of the major task that is performed by the operating system. The operating system directs the computer to perform execution of the different task or programs one by one and keeping the system away from the busy state. This also saves the CPU from being idle. It also reduces the time that includes the entire accomplishment process time, waiting interval and the comeback time of processor.

Proposed Idea in Detail:

Flowchart

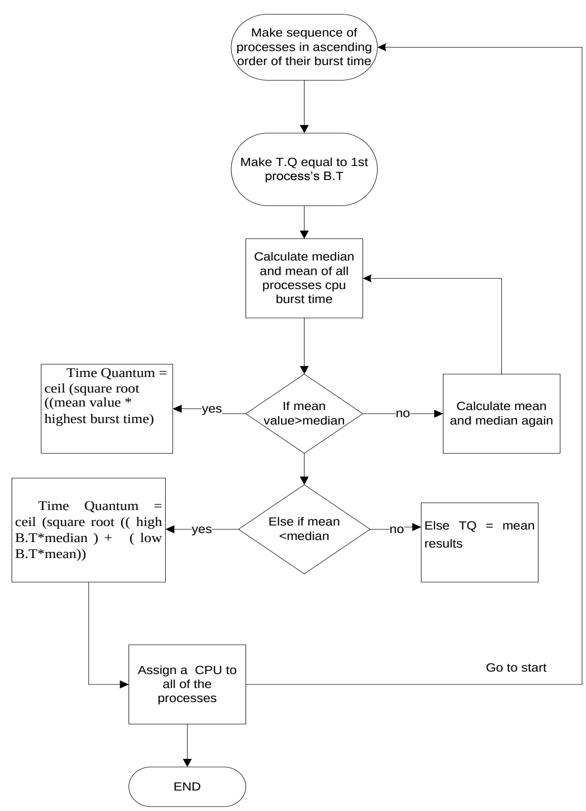

Fig. 1: Flowchart of the algorithm

As we come to know from the above examples that there are little variations among their outputs on the basis of those variations each algorithm is different from the other one. The proposed algorithm uses the technique of RR with SJF in place of FCFS [14]. This new idea will help us to eliminate the factors that are considered as disadvantages of an RR with FCFS. This RR algorithm is not producing the productive results in case of processes having small burst time. This will increase the response time and the wait time of a processor thus decreasing the system throughput. In the proposed idea it has been shown that the time quantum is not fixed. It is dynamic and its value changes according to classification of process's burst time. This new one algorithm will provide an ease to solve the problems that our processor is facing using the simple RR scheduling. This new algorithm will reduce the delay time and turnaround time. The algorithm has following steps [11]:

Step 1:

```
For x=1 to A-length
B.T <= A[x]
I <= x-1
While (I > 0 && A [y] > B.T)
{
A [y + 1] <= A [y]
I <= y-1
}
A [y+1] <= B.T
```

Step 2: Find the mean and median of CPU burst time of all processes.

```
Integer RR = R.T [0]
Median = array / 2
Median = mean
```

Step 3: Set the time quantum of an RR in an order that burst time of a first process will become identical to that time. Note that small process is selected first.

```
If(mean value > median)  Then:
    Time Quantum = cell (square root ((mean value * highest burst time) + (median * lowest burst time)))

Else If (Median > Mean)  Then:
    Time Quantum = cell (square root ((median * highest burst time) + (mean value * lowest burst time)))
Else
    Time Quantum = mean value

End If
```

Step 4: Use following method to arrange the time quantum

Step 5: Carry on the process of assigning the CPU according to the RR algorithm.

Step 6: Go back towards the step1.

Case 1:

Increasing Order:

The time quantum in this case is arranged in an ascending order of the B.T of the processes [19]. The table 6 will show the comparison of results using the shortest remaining burst RR, improved shortest remaining burst time RR [19] and the newly proposed algorithm that is efficient dual nature RR.

Consider seven processes named A, B, C, D, E, F and G with their CPU burst time.

Table V: Ascending order RR with SJF

Processes	B.T in ms (mili seconds)
A	20
B	25
C	35
D	50
E	80
F	90

Let the time quantum is 40 milliseconds.

Context switches that process make: 12

Average wait Time of a process:152.14 millisecond

Average Turnaround Time of a process:212.14 millisecond

According to the proposed idea the results are as follows:

Context switches that a process makes: 6 Average wait Time taken by a process:112.14 millisecond

Average Turnaround Time taken by a process: 172.14 millisecond

Table VI: Comparison between New Ideas, RR, SRBRR (Case 1)

Algorithm	T.Q	Avg. W.T	Avg. TA time	CS
RR	40	152.14	212.14	12
SRBRR	50,40,30	137.57	193.57	9
ISRBRR	70	179.00	205.59	9
EDNRR	91,92	112.14	172.14	6

The difference between wait times,
Given by following equation:

$$Difference = \frac{|V1-V2|}{(V1+V2)/2} x100 \quad [21] \tag{6}$$

Where V_1 and V_2 are the least values of weight time:

$V_1 = 137.57$, $V_2 = 112.4$

Result: 20.36 %

Fig. 2: Comparison using the increasing orders of T.Q.

Case 2:

Decreasing Order:

The same procedure as we have used in case 1 is used but in the decreasing order sequence of CPU burst time and the time quantum is also arranged in the decreasing order [20].

Consider the 7 different processes with different B.T e.g., 80, 50, 40, 20, 15, 10 and 5 milliseconds as CPU B.T.

Table VII: Comparison between RR, New idea and SRBRR (Case 2)

Algo's	T.Q	Average	Average	CS
		Wait time	TA time	
RR	20	108.57	140	11
SRBRR	30	55.71	87.14	9
ISBRR	40	64.28	84.28	8
EDNRR	51,29	47.14	78.57	6

Using equation (6)
Where V_1 and V_2 are the least values of weight time:
$V_1 = 55.71$, $V_2 = 47.14$
Result:16.66 %

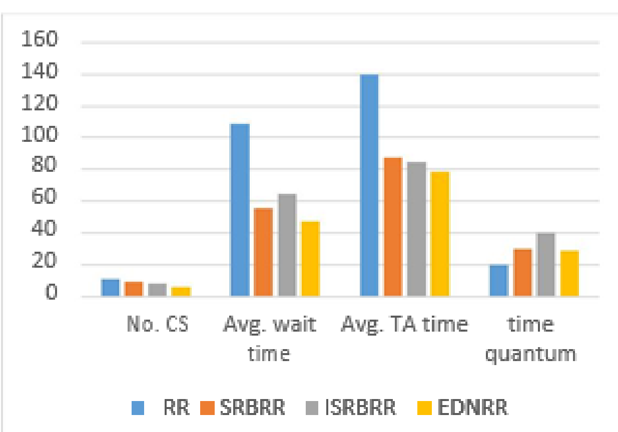

Fig. 3: Comparison using the decreasing orders of time quantum

Case 3:

Random TQ:

According to this kind of technique we maintain the T.Q according to the B.T of the process and it will change at random [20]. If process which is under observation is small then the time quantum is also made small and if there is a large process then the time quantum is changed and it is increased. In this way our CPU will never go in the idle state because in case of a static time quantum the CPU may get idle if a process with small burst time than that of the time quantum is maintained. This algorithm uses the technique of RR with SJF but the time quantum will change for the each next in coming process according to its burst time.

Consider 5 processes with 80, 60, 20, 10 and 30 milliseconds CPU B.T.

Table VIII: Comparison between New idea, SRBRR and RR (Case 3)

Algo's	T.Q	Averag eWait time	Avera geT.A time	CS
RR	30	94	134	7
SRBRR	30,50	50	90	6
ISRBR R	49	73.40	93.80	6
EDNRR	59,19	44	84	4

Where V_1 and V_2 are the least values of weight time: $V_1 = 50$, $V_2 = 44$
Result: 12.76 %

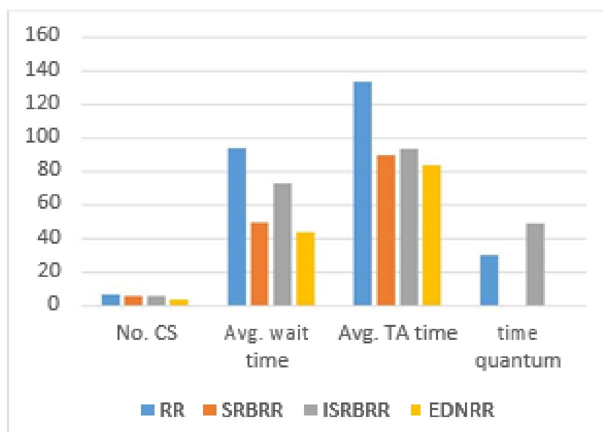

Fig. 4: Comparison using the random TQ

IV. CONCLUSION

CPU scheduling is the most important phenomenon that will perform the execution of various processes by scheduling the processes using different algorithms. We have come to know the idea of FCFS algorithm, SJF algorithm, the RR algorithm, and priority based algorithm. This research paper show the comparison of RR, SRBRR [17], ISRBRR [17], [18] and the other scheduling algorithms especially simple RR, FCFS with RR, SJF with RR to propose EDNRR. We have shown that the simple RR with SJF (Shortest Job First) work more efficient and fast with the dynamic time quantum that changes in an increasing, decreasing and random order of a sequence. The performance and utilization of a CPU has been increased using this approach by considering the comparison between their waiting time and turnaround time. SJF with RR and the time quantum is arranged in a different sequence of its burst time is producing much better results than the simple ISBRR and SRBRR. So the result of ENDRR is more efficient in this respect.

REFERENCES

[1]. Dalal, S., & Singh, V. IMPROVED DYNAMIC ROUND ROBIN PROCESS SCHEDULING.

[2]. Rami J. Matarneh,, "Self-Adjustment Time Quantum in Round Robin Algorithm Depending on Burst Time of Now Running Processes", American J. of Applied Sciences, Vol. 6, No. 10, pp1831- 1837. (2009)

[3]. Peterson, J.L. and Silberschatz, A., 1985. Operating system concepts (Vol. 2). Reading, MA: Addison-Wesley.

[4]. Mittal, P.K., 2014. An Efficient Dynamic Round Robin CPU Scheduling Algorithm. *IJARCSSE Journal*, *4*(5).

[5]. Ishwari Singh Rajput," A Priority based Round Robin CPU Scheduling Algorithm for Real Time Systems", (IJIET)International Journal of Innovations in Engineering and Technology Vol. 1 Issue 3 Oct 2012.

[6]. P. Balakrishna Prasad, "Operating Systems" Second Edition.

[7]. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, "Operating System Concepts", Sixth Edition.

[8]. Milan Milenkovic, "Operating Systems Concepts and Design", McGRAM-HILL, Computer Science Series, second edition.

[9]. Sourav Kumar Bhoi, Sanjaya Kumar Panda, DebasheeTarai, "Enhancing cpu performance using sub contrary mean dynamic round robin (Smdrr) scheduling algorithm", JGRCS, Volume 2, No.12, December 2011, pp.17-21.

[10]. Ishwari Singh Rajput," A Priority based Round Robin CPU Scheduling Algorithm for Real Time Systems", (IJIET)International Journal of Innovations in Engineering and Technology Vol. 1 Issue 3 Oct 2012.

[11]. P. Balakrishna Prasad, "Operating Systems" Second Edition.

[12]. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, "Operating System Concepts", Sixth Edition.

[13]. Milan Milenkovic, "Operating Systems Concepts and Design", McGRAM-HILL, Computer Science Series, second edition.

[14]. Sourav Kumar Bhoi, Sanjaya Kumar Panda, DebasheeTarai, "Enhancing cpu performance using sub contrary mean dynamic round robin (smdrr) scheduling algorithm" ,JGRCS, Volume 2, No. 12, December 2011, pp.17-21.

[15]. W. Athas, J. Koller, L. Svenson, "An energy-efficient CMOS line driver using adiabatic switching. 1994", *IEEE Fourth GreatLakes Symposium on VLSI*, 1993.

[16]. A. Chandrakasan, S. Sheng, R. Broderson, "Low-power CMOS digital design", *JSSC*, vol. 27, no. 4, pp. 473-484, 1992.

[17]. Goel, N., &Garg, R. B. (2013). A comparative study of CPU scheduling algorithms.*ArXiv preprint arXiv: 1307.4165*.

[18]. Jones, M.B., Roşu, D. and Roşu, M.C., 1997. CPU reservations and time constraints: Efficient, predictable scheduling of independent activities (Vol. 31, No. 5, pp. 198-211). ACM.

[19]. Abdul Rahim, A., Abdullahi, S.E. and Sahalu, J.B., 2014. A new improved round robin (NIRR) CPU scheduling algorithm. *International Journal of Computer Applications*, *90*(4).

[20]. Noon, A., Kalakech, A. and Kadry,S.,2011. A new round robin based scheduling algorithm for operating systems: dynamic quantum using the mean average. *ArXivpreprint arXiv:1111.5348*.

[21]. Lang, L.H. and Litzenberger, R.H., " Dividend announcements: Cash flow signalling vs. free cash flow hypothesis", " *journal of Financial Economics"*, *24*(1), pp.181-191,1989