Chittagong University of Engineering & Technology

Dept. of Computer Science & Engineering

Implementation of a novel CPU scheduling algorithm and conduct a comparative study with the existing CPU scheduling algorithms

Course Code: CSE-336
Course Title: Operating Systems (Sessional)
Date of submission: 8th August, 2023

Submitted By-

Name: Abu Md. Masbah Uddin
1904001

Name: Rakibul Hasan
ID: 1904002

Name: Hasin Ahmed Samin
ID: 1904003

Name: Mohammad Minhaz Zisun
ID: 1904004

Name: Tofayel Ahmmed Babu
ID: 1904005

Name: Safiul Alam
ID: 1904006

Submitted To-

Md. Shafiul Alam Forhad
Assistant Professor, Dept. of CSE, CUET

Hasan Murad
Lecturer, Dept. of CSE, CUET

REMARKS

Objectives: The main purposes of this assignment are –

- To study different kinds of CPU scheduling algorithm
- Implementation of different kinds of CPU scheduling algorithm taught us in the theory class
- To propose a new CPU scheduling algorithm and its implementation
- To conduct a comparative study among different kinds of scheduling algorithms

Abstract: Central Processing Unit (CPU) scheduling algorithms are an integral part of an operating system's process management. They determine the order in which processes are executed on a CPU, ensuring fair and efficient utilization of computing resources. The primary goal of a CPU scheduling algorithm is to optimize system performance by minimizing the response time, turnaround time, waiting time, and maximizing throughput. In a multitasking environment, where multiple processes to execute next and how long it should run before switching to another process. This decision-making process is crucial for achieving good system performance and user satisfaction. There are several types of CPU scheduling algorithms such as – First Come, First Served (FCFS), Priority based CPU scheduling, Shortest Job First (SJF), Round Robin algorithm (RR) and many more. In this report a new CPU scheduling algorithm has been proposed that used Round Robin algorithm, named as Priority Based Hybrid CPU scheduling (PBHS). That uses dynamic time quantum instead of static time quantum used in RR. The performance of the proposed algorithm experimentally compared with the already existing algorithms and a research paper's algorithm (DABRR – Dynamic Average Burst Round Robin). The results of our approach presented in this report and tries to improve performance in terms of average waiting time, average turn around time and average response time.

Keywords: CPU scheduling, FCFS, Priority Based CPU scheduling, SJF, RR, DABRR, PBHS, turn around time, waiting time, response time.

1.Introduction:

A process is the instance of a computer program that is being executed by one or many threads. The execution of a process must be oriented and progressed in a sequential fashion. When a program is loaded into the memory and it becomes a process. When a process executed, CPU needs make a decision how manner the process will be executed and how much time it will occupies the CPU. Besides, it's need to make a decision when a new process will

arrive, what step should be taken to the current process. All of these characteristics are named as performance measurement such as turnaround time, waiting time, response time and context switching. To handle all of these activities and makes the CPU as efficient, we have introduced

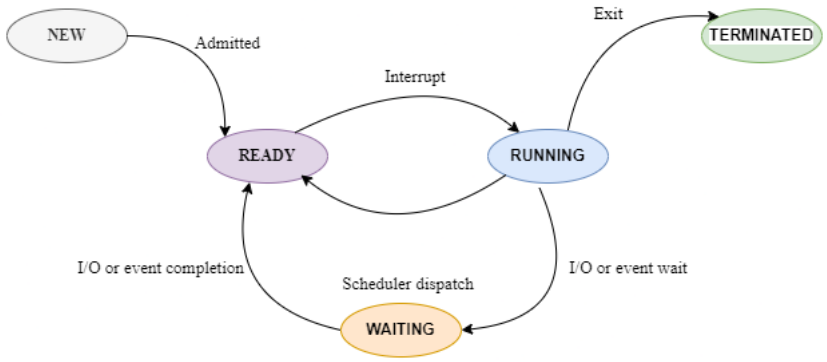


Figure-01: Diagram of Process State

some CPU scheduling algorithms such as FCFS, Priority Based CPU scheduling, SJF, RR etc. The less turnaround time, the less waiting time and the less response time makes the algorithm better and OS uses the CPU efficiently and wisely.

2.Our Proposal: In this assessment we have introduced a new CPU scheduling algorithm, named as Priority Based Hybrid Scheduling (PBHS) that uses the Round Robin concept with priority at a time. Here, we have combined both of the priority scheduling and round robin to make a dynamic time quantum. Thus, we have reduced the average response time, average waiting time and average turnaround time compared by other existing CPU scheduling algorithms. After that for more visualization, we make comparison among all of the existing algorithms with our own algorithms. Here, we include the PBHS algorithm below–

PBHS (algorithm):

```
Q: Ready Queue
n: number of process
Pi: Priority of i'th process
Bi: burst time of i'th process
Ai: arrival time of i'th process
Wi: waiting time of i'th process
Ti: turnaround time of i'th process
Si: start time of i'th process
```

```
Fi: End time of i'th process
Ri: response time of i'th process
i, j: used as index of ready queue
avgT: average turnaround time
avgW: average waiting time
avgR: average response time
[1] Arrange the processes in ascending order based on arriva
l time.
[2] n = number of processes in Q
[3] complete = 0, t = 0
[4] Repeat step 5 to 21 till complete < n
[5] id = Q.pop()
[6] TQ := n - P[id] + sqrt(∑(Bi * Ai)) + 1
[7] if(burst time of Pi) > TQ then Execute the process
[8]     burst time of Pi = burst time of Pi - TQ
[9]     t = t + TQ
[10] else
[11]  execute the process
[12]  complete++
[13]  t = t + B[id]
[14]  B[id] = 0;
[15] Declare C[n]
[16] repeat step 16 and 17 for j = 1 till j < n + 1
[17]    if any new process arrive in time t
[18]         Pi push into C[]
[19] sort C[] based on arrival time
[20] Push C[] into Q
[21] goto step 4
[22] return
```

2.1. Illustration: In this section we analyzed the execution of the proposed algorithm. Suppose we have 5 processes P1, P2, P3, P4, P5 and their arrival time are 0 millisecond and burst time 10 20 34 40 55 respectively. Also let the priority of the processes are 1 2 3 4 5 respectively. Then Initially, all process will enter into the ready queue. Then, the process P1 will be pop from the ready queue and the time quantum for this process is TQ = n – P[i] + 1 = 5 – 1 + 1 = 5, here the lower priority gets the higher time slice. As a result, CPU reduce the response time. Thus, it can scale down the starvation. Similarly, the burst time for P2, P3, P4, P5 are 4, 3, 2, 1 respectively. After the first iteration, the remaining burst time will be 5 16 31 38 and 54 for P1 P2 P3 P4 and P5 respectively. In the second iteration P1 will be terminated. Then the remaining time for p2, p3, p4 and p5 will be 12 28 26 and 51 respectively. Thus, the processes will go through several iteration until complete not equal 5.

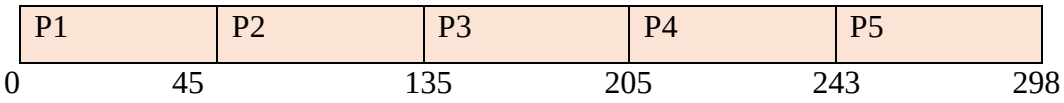
3.Experimental Analysis: There are lots of CPU scheduling algorithm. In this report, the most common CPU scheduling algorithm included.

Table-01: Common Input for all CPU scheduling algorithms

Process	Arrival Time	Burst Time
P1	0	45
P2	5	90
P3	8	70
P4	15	38
P5	20	55

3.1. FCFS (First Come, First Served): First-Come, First-Served (FCFS) is one of the simplest and most straightforward CPU scheduling algorithms. Basically, it is a non-preemptive scheduling algorithm. In FCFS, the processes are executed in the order they arrive in the ready queue. The CPU is assigned to the first process that enters the system and it continues executing until it completes its CPU burst time.

Gannt Chart:



Average Response Time = (0+45+135+205+24) / 5 = 125.6

Average Turnaround Time = (45+130+197+228+278) / 5 = 175.6

Average Waiting Time = (0+40+127+190+223) / 5 = 116

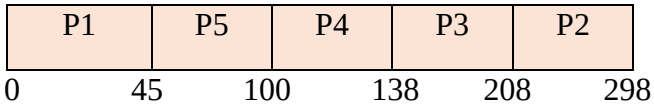
3.2. Priority Based scheduling(non-preemptive): In the non-preemptive Priority scheduling, The Processes are scheduled according to the priority number assigned to them. Once the process gets scheduled, it will run till the completion. Generally, the higher the priority number, the higher is the priority of the process.

Table-02: Priority Value of processes

Process	Priority
P1	1
P2	2

P3	3
P4	4
P5	5

Gantt Chart:



Average Response Time = (0+203+130+85+25) / 5 = 88.6

Average Turnaround Time = (45+293+200+123+80) / 5 = 148.2

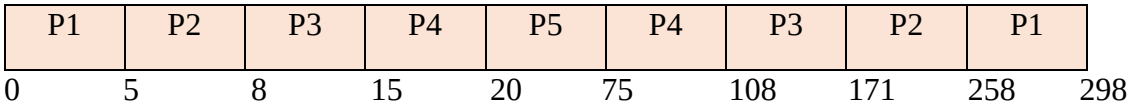
Average Waiting Time = (0+203+130+85+25) / 5 = 88.6

3.3. Priority Based Scheduling (Preemptive): Preemptive Priority CPU Scheduling Algorithm is a pre-emptive method of CPU scheduling algorithm that works based on the priority of a process. In this algorithm, the scheduler schedules the tasks to work as per the priority, which means that a higher priority process should be executed first. In case of any conflict, i.e., when there is more than one process with equal priorities, then the pre-emptive priority CPU scheduling algorithm works on the basis of FCFS (First Come First Serve) algorithm.

Table-03: Priority Value of processes

Process	Priority
P1	1
P2	2
P3	3
P4	4
P5	5

Gantt Chart:



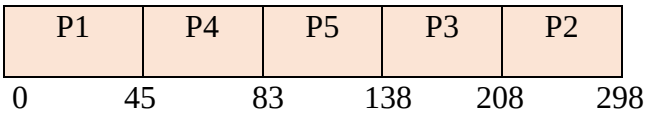
Average Response Time = (0+5+8+15+20) / 5 = 9.6

Average Turnaround Time = (298+253+163+93+55) / 5 = 172.4

Average Waiting Time = (253+163+93+55+0) / 5 = 112.8

3.4. SJF (non-preemptive): Shortest Job First (SJF) is an algorithm in which the process having the smallest execution time is chosen for the next execution. This scheduling method can be preemptive or non-preemptive. It significantly reduces the average waiting time for other processes awaiting execution. In non-preemptive scheduling, once the CPU cycle is allocated to process, the process holds it till it reaches a waiting state or terminated.

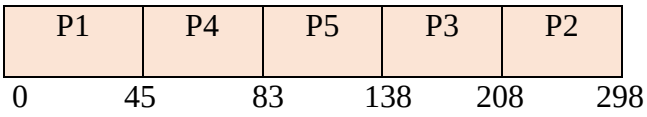
Gannt Chart:



Average Response Time = (0+203+130+30+63) / 5 = 85.2
Average Turnaround Time = (45+293+200+68+118) / 5 = 144.8
Average Waiting Time = (0+203+130+30+63) / 5 = 85.2

3.5. SJF (preemptive): In Preemptive SJF Scheduling, jobs are put into the ready queue as they come. A process with shortest burst time begins execution. If a process with even a shorter burst time arrives, the current process is removed or preempted from execution, and the shorter job is allocated CPU cycle.

Gannt Chart:



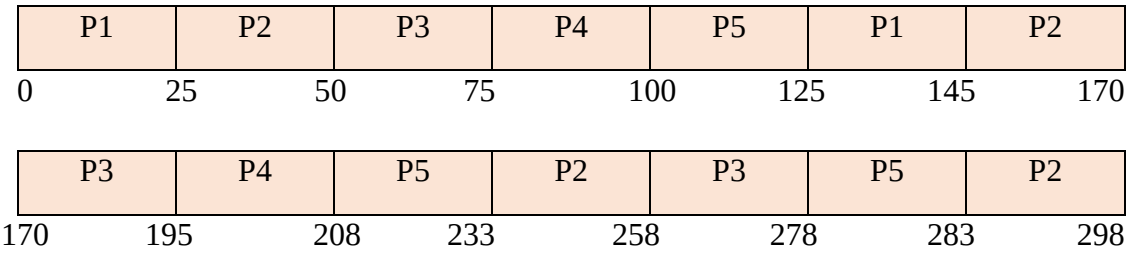
Average Response Time = (0+203+130+30+63) / 5 = 85.2

Average Turnaround Time = (45+293+200+68+118) / 5 = 144.8

Average Waiting Time = (0+203+130+30+63) / 5 = 85.2

3.6. RR: The name of this algorithm comes from the round-robin principle, where each person gets an equal share of something in turns. It is the oldest, simplest scheduling algorithm, which is mostly used for multitasking. In Round-robin scheduling, each ready task runs turn by turn only in a cyclic queue for a limited time quantum.

Gannt Chart:



Average Response Time = (0+20+42+60+80) / 5 = 40.4

Average Turnaround Time = (145+293+270+193+263) / 5 = 232.8

Average Waiting Time = (100+203+200+155+208) / 5 = 173.2

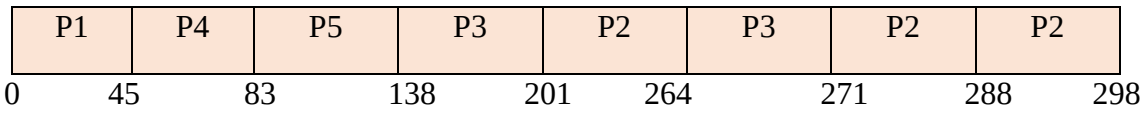
3.7. DABRR: DABRR is an optimized round robin CPU scheduling algorithm with dynamic time quantum proposed by Amar Ranjan Dash1, Sandipta Kumar Sahu2 and Sanjay Kumar Samantra3. There proposed algorithm is given below. According to their proposed algorithm we implemented the algorithm using C++ and made the comparison with our own proposed algorithm and among the existing algorithms.

ALGORITHM:

```
TQ: Time Quantum
RQ: Ready Queue
n: number of process
Pi: Process at ith index
i, j: used as index of ready queue
TBT: Total Burst Time
[1] Arrange the processes in ascending order.
[2] n = number of processes in RQ
[3] i=0, TBT=0
[4] Repeat step 5 and 6 till i < n
[5] TBT += burst time of process Pi
[6] i++
[7] TQ = TBT/n
[8] j = 0
[9] Repeat from step 12 to 19 till j<n
[10] if (burst time of Pi) <= TQ
[11] Execute the process
```

```
[12] Take the process out of RQ
[13] n--
[14] Else
[15] Execute the process for a time interval up to 1 T
Q
[16] Burst time of Pi = Burst time of Pi - TQ
[17] Add the process to ready queue for next round of
execution
[18] j++
[19] If new process arrives
[20] goto step 1
[21] If RQ is not empty
[22] goto step 2
```

Gannt Chart:



Average Response Time = (0+196+130+30+63) / 5 = 83.8

Average Turnaround Time = (45+293+263+68+118) / 5 = 157.4

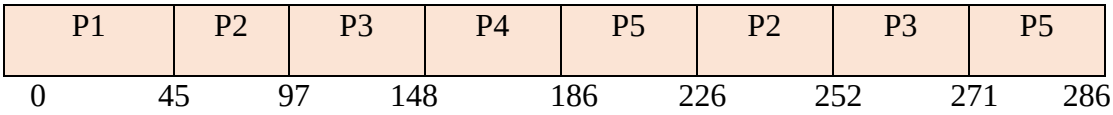
Average Waiting Time = (0+203+193+30+63) / 5 = 97.8

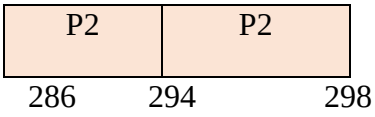
3.8. PBHS:

Table-04: Priority for the processes

Process	Priority
P1	2
P2	5
P3	4
P4	1
P5	3

Gannt Chart:





Average Response Time = (+203+193+133+211) / 5 = 148

Average Turnaround Time = (45+293+263+171+266) / 5 = 207.6

Average Waiting Time = (0+40+89+133+166) / 5 = 85.6

4. Comparative Analysis: This section provides the comparative analysis of 6 CPU scheduling algorithms in terms of average waiting time, average response time and average turn around time based on their calculated results.

Table – 05: Comparative analysis of different CPU scheduling algo.

Algorithms	Average Response Time	Average Turnaround time	Average waiting time
FCFS	125.6	175.6	116
Priority CPU scheduling	9.6	172.4	112.8
SJF	85.2	144.8	85.2
RR	40.4	232.8	173.2
DABRR	83.3	157.4	97.8
PBHS	148	207.6	85.6

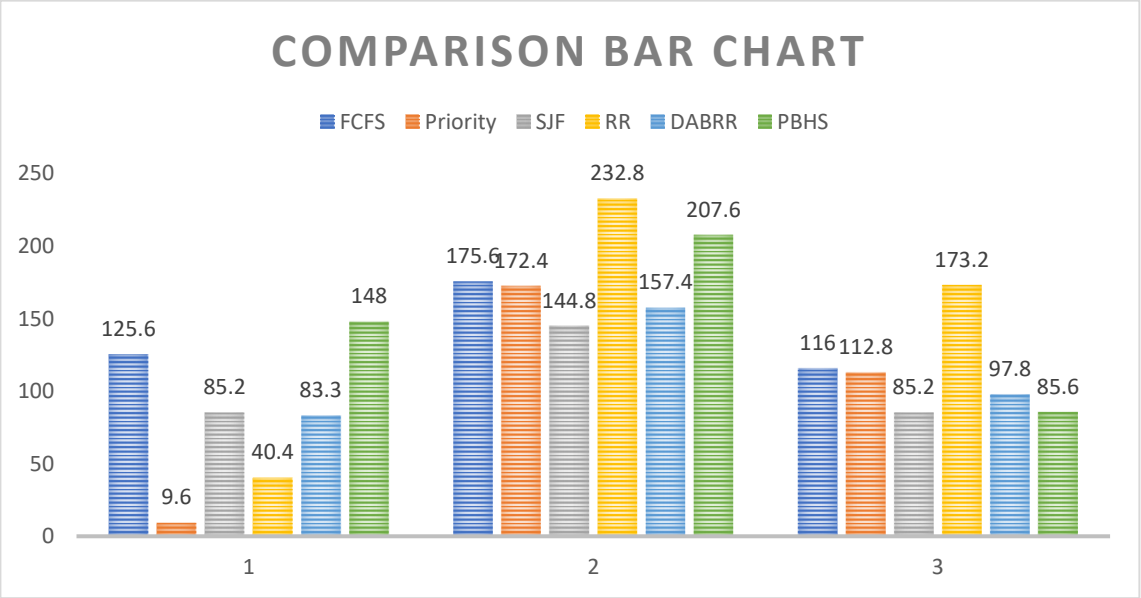
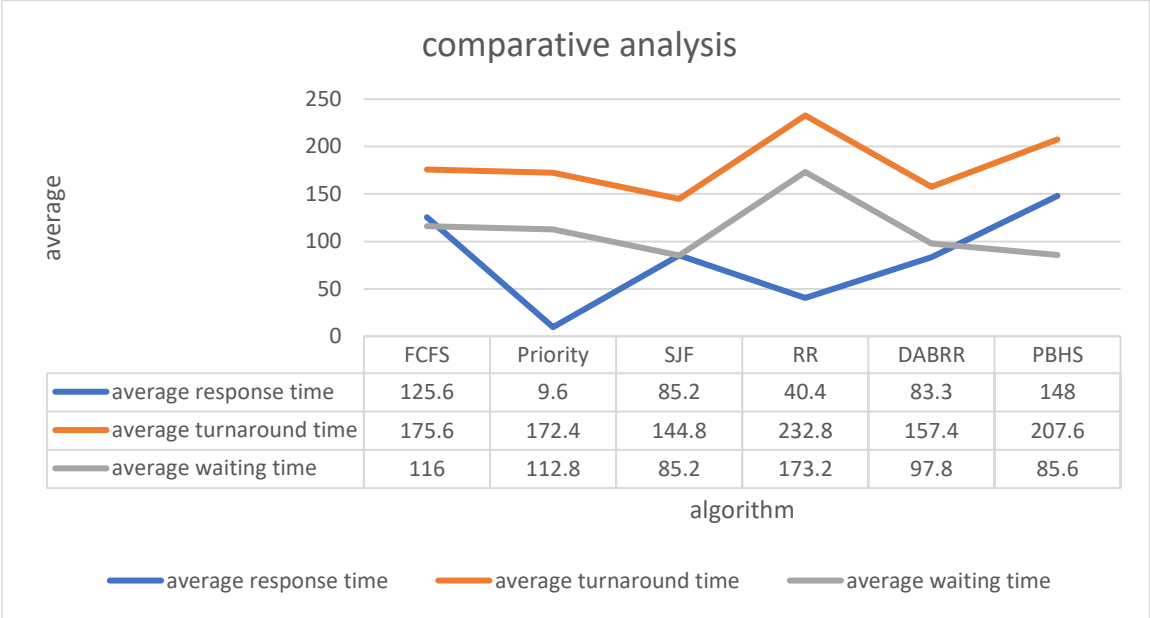


Figure-02: Comparative analysis of different types of CPU scheduling using bar charts



	FCFS	Priority	SJF	RR	DABRR	PBHS
average response time	125.6	9.6	85.2	40.4	83.3	148
average turnaround time	175.6	172.4	144.8	232.8	157.4	207.6
average waiting time	116	112.8	85.2	173.2	97.8	85.6

Figure-03: Comparative analysis of different types of CPU scheduling algorithms using line graph

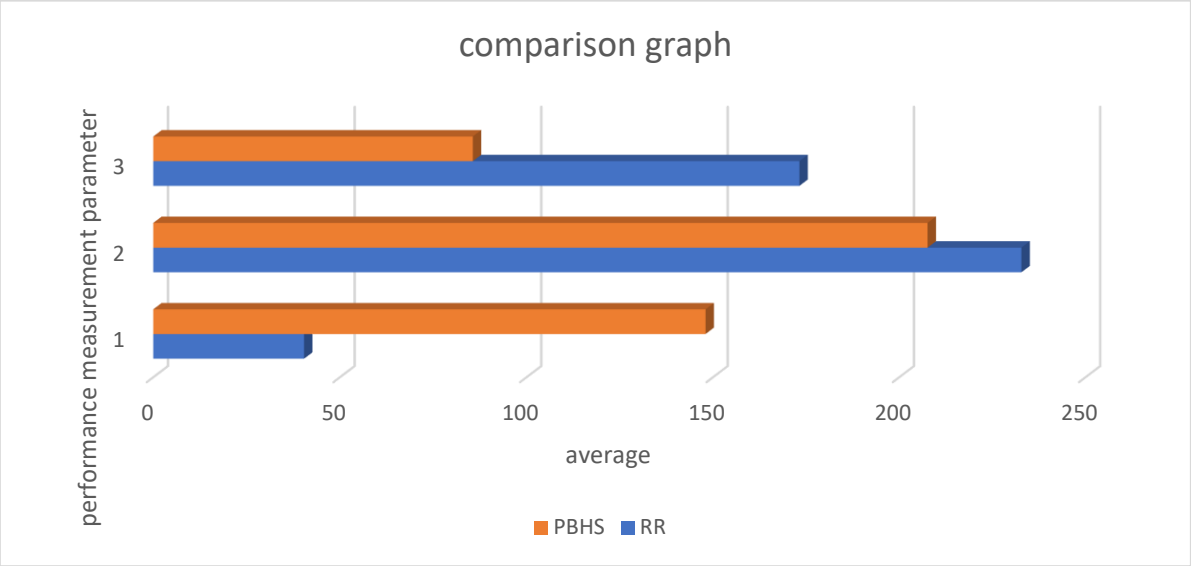


Figure-04: Comparison between PBHS and RR

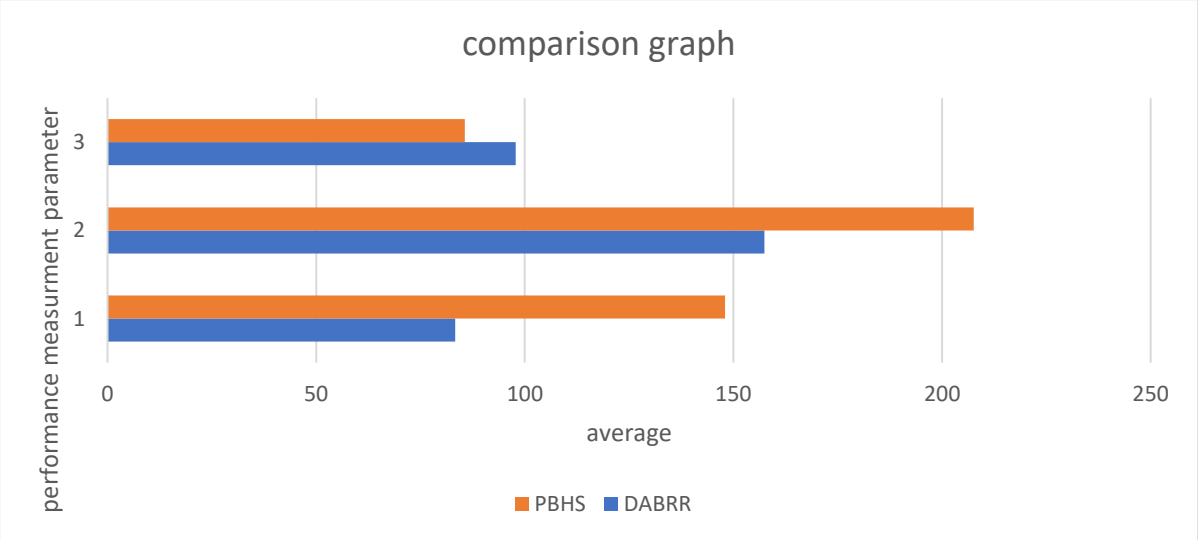


Figure-05: Comparison between PBHS and DABRR

5. Conclusion: This report presents Priority Based Hybrid CPU Scheduling (PBHS) algorithm that combine the Priority and RR together. This report also presents comparative discussion between PBHS and other existing algorithms. Here, we have seen our proposed algorithm gives the better performance measure than RR algorithm and any other mention in this report. In this report, we have also compared our proposed algorithm with another proposed algorithm named DABRR given by a group of researchers. We have seen that Our algorithm gives better average response time than the DABRR. Further added, our proposed algorithm much better than the other existing algorithm. Thus, we conclude that PBHS uses CPU more efficiently other than existing algorithms.