CPU scheduling is one of the important concepts of operating system. It is a method used to schedule jobs for execution. CPU scheduling involves a careful examination of pending processes to determine the most efficient way to service the requests. Round-Robin (RR) is one of the algorithms employed by the process and network schedulers in computing. As the term is generally used, time slices are assigned to each process in equal portions and in circular order, handling all the processes without priority. More context switching, large response time, large waiting time, large turnaround time and less throughput are the main disadvantages. Approached algorithm removed the disadvantages of Round robin algorithm (Context switching). It reduces the average waiting time, turnaround time and increase throughput.

I. INTRODUCTION
Round Robin is the preemptive process scheduling algorithm. The Round Robin scheduling algorithm has the disadvantage of more context switching. So in this paper we try to reduce the disadvantages of RR scheduling algorithm and a new approach is used...
to reduce the number of context switches and also minimise the turnaround time and waiting time to maximise the throughput as well as CPU utilization.

Features in a good scheduling algorithm:

- Minimum turnaround time
- Minimum waiting time and response time
- Minimum context switching
- Maximum CPU utilization
- Maximum throughput

II. OBJECTIVE AND PERFORMANCE CRITERIA

The main objectives of CPU scheduling and their performance criteria are listed below:

2.1 Objective

There are many objectives that must be considered in the design of scheduling, such as:

- Throughput: Throughput is the rate at which processes are completed per unit of time.
- Fairness: Fairness represents the degree to which all processes are given equal opportunity of execution. This criterion is codified in time shared system.
- Indefinite postponement: Avoiding indefinite postponement of any process so that each process is executed in a certain amount of time.
- Priority: Give proper priority to every process so that CPU can do maximum task within a minimum time.
- Resources: Scheduling mechanism should keep the resources of the system busy.

2.2 Performance Criteria

Scheduling Criteria of CPU for a scheduler varies from one scheduler to another. There are many scheduling algorithms. Different scheduling algorithms have different properties. The selection of a proper scheduling algorithm may improve the system performance. Many criteria have been suggested for evaluating the scheduling algorithm. Some commonly used scheduling criteria are:

- CPU Utilization: The CPU must be busy as much as possible to perform different activities. CPU utilization is very important in real time and multiprogramming system. The high CPU utilization is achieved on heavily loaded system.
- Throughput: The number of processes executed by the system in a specific period of time, this time unit is called throughput.
- Turnaround Time: Time required for a particular process to complete from submission time to completion. Turnaround time is the sum of periods spent waiting to get into memory, waiting in the ready queue, executing in the CPU and doing input output.
- Waiting Time: How much time processes spend in the ready queue waiting their turn to get on the CPU, is known as waiting time.
- Response Time: The interval from the time of submission of a process until the first response is produced from the CPU, is known as response time.
III. PREVIOUS WORK DONE
Various scheduling algorithms already in use and terms used in this scheduling are:

- **CPU Scheduling:** CPU scheduling is a process which allows one process to use the CPU while the execution of another process is in waiting state due to unavailability of any resource like I/O etc., thereby making full use of CPU. The aim of CPU scheduling is to make the system efficient, fast and fair.

- **Round Robin CPU Scheduling:** One of the oldest, simplest and most widely used algorithms round robin (RR). In the round robin scheduling, processes are dispatched in a FIFO (First In First Out) manner but are given a limited amount of CPU time called a time slice or a quantum. If a process does not complete before its CPU time, the CPU is pre-empted and given to the next process waiting in a queue. The pre-empted process is then placed at the back of the ready list. Round Robin Scheduling is preemptive therefore it is effective in time sharing environments in which the system needs to guarantee reasonable response times for interactive users.

- **Burst Time:** Burst time is actually time that is required to complete execution of particular task or process.

- **Time Slice or, Time Quantum:** Time slicing is a scheduling mechanism/way used in time sharing systems. The aim of time slicing scheduling is to give all processes an equal opportunity to use CPU. In this type of scheduling, CPU time is divided into slices that are to be allocated to ready processes. Short processes may be executed within a single time quantum. Long processes may require several quanta.

- **Context Switch:** A context switch (sometimes referred as a process switch or a task switch) is the mechanism to store and restore the state or context of a CPU in Process Control Block (PCB) so that a process execution can be resumed from the same point at a later time. Context switching is an essential part of a multitasking operating system features.

- **Gantt chart:** A Gantt chart is a type of bar chart or, horizontal bar chart that shows the start and finish time of several processes. A Gantt chart provides a graphical illustration of schedule that helps to plan, coordinate and track specific tasks in a project.

IV. EXISTING SCHEDULING ALGORITHM AND REVIEW

4.1 Types of Scheduling
Non-preemptive Scheduling: In non-preemptive once a process starts its execution it will terminate only when it ends. Non-preemption scheduling is one which can be applied in the circumstances when a process terminates, or process switches from running to ready state. In non-preemptive scheduling, once the resources (CPU) are allocated to a process, the process holds the CPU till it gets terminated or it reaches a waiting state.
Preemptive Scheduling: Preemption scheduling is one which can be done in the circumstances when a process switches from running state to ready state or, from waiting state to ready state.

### 4.2 Existing Scheduling Algorithm

First Come First Serve (FCFS) algorithm is the simplest scheduling algorithm. Processes are dispatched according to their arrival time on the ready queue. Being a non-preemptive discipline, once a process has a CPU, it runs to completion.

Shortest Job First (SJF), also known as Shortest Job Next (SJN) or, Shortest Process Next (SPN), is a scheduling policy that selects for execution the waiting process with the smallest execution time. SJF is a non-preemptive algorithm. Shortest Remaining Time First (SRTF) is a preemptive variant of SJF.

Priority Scheduling is a method of scheduling processes based on priority. In this method, the scheduler chooses the tasks to work as per the priority, which is different from other type of scheduling.

Round Robin (RR) CPU Scheduling is a job scheduling algorithm that is considered to be very fair, as it uses time slice that are assigned to each process in the queue or line. Each process is then allowed to use the CPU for a given amount of time, and if it does not finish within the allotted time, it is pre-empted and then moved at the back of the line so that the next process in line is able to use the CPU for the same amount of time.

### 4.3 Review

These all are traditional scheme of CPU scheduling algorithm. The closest work in this area is to modify the RR CPU scheduling algorithm and we show that the proposed algorithm gives better average waiting time and average turnaround time than the traditional one.

### V. PROPOSED ALGORITHM

This proposed round robin scheduling is based on the small changes in traditional round robin scheduling. In this proposed scheduling, the jobs are first rearranged in an ascending order according to their burst time, and also set the time slice as the average burst time of the jobs. The proposed algorithm has less number of context switching and also, its average turnaround time and average waiting time are less than the traditional one.

1. **Step 1**: The processes are rearranged in an ascending order according to their burst time.
2. **Step 2**: Set the time slice as average burst time of all the processes.
3. **Step 3**: Round Robin Scheduling will take place to execute all the processes inside the CPU. If any process does not complete before it’s time slice, it will go to the end of the ready queue and waits for its next turn to execute in the CPU.

**Example 1**

Suppose there are three jobs. And their burst time is given below:
According to traditional Round Robin scheduling algorithm:
Suppose, time slice = 2 ms.
Gantt chart:

<table>
<thead>
<tr>
<th>Process</th>
<th>Burst Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>5</td>
</tr>
<tr>
<td>P2</td>
<td>2</td>
</tr>
<tr>
<td>P3</td>
<td>3</td>
</tr>
</tbody>
</table>

Total Turnaround Time = (10+4+9) ms = 23 ms
Average Turnaround Time = Total Turnaround Time / Total number of processes 
= (23/3) ms = 7.667 ms
Total Waiting Time = (5+2+6) ms = 13 ms
Average Waiting Time = Total Waiting Time / Total number of processes 
= (13/3) ms = 4.333 ms

Now the proposed one is like this:
Rearrange the processes according to their burst time in ascending order and set the time slice as the average burst time of the processes.
So, according to the proposed round robin scheduling algorithm:

<table>
<thead>
<tr>
<th>Process</th>
<th>Burst Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>P2</td>
<td>2</td>
</tr>
<tr>
<td>P3</td>
<td>3</td>
</tr>
<tr>
<td>P1</td>
<td>5</td>
</tr>
</tbody>
</table>

Time Slice = Total burst time of all the processes/Number of processes 
= (2+3+5) / 3
= 10/3 ms = 3.33 ms
Gantt chart:

<table>
<thead>
<tr>
<th>P2</th>
<th>P3</th>
<th>P1</th>
<th>8.33</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2</td>
<td>5</td>
<td>8.33</td>
<td>10</td>
</tr>
</tbody>
</table>

Total Turnaround Time = (2+5+10) ms = 17 ms
Average Turnaround Time = Total Turnaround Time / Total number of processes 
= (17/3) ms = 5.667 ms
Total Waiting Time = (0+2+5) ms = 7 ms
Average Waiting Time = Total Waiting Time / Total number of processes 
= (7/3) ms = 2.333 ms

Example 2
Suppose there are four jobs. And their burst time is given below:

<table>
<thead>
<tr>
<th>Process</th>
<th>Burst Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>4</td>
</tr>
<tr>
<td>P2</td>
<td>6</td>
</tr>
</tbody>
</table>
According to Round Robin scheduling algorithm:
Suppose, time slice=2 ms.

Gantt chart:

<table>
<thead>
<tr>
<th>P1</th>
<th>P2</th>
<th>P3</th>
<th>P4</th>
<th>P1</th>
<th>P2</th>
<th>P3</th>
<th>P4</th>
<th>P2</th>
<th>P3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2</td>
<td>4</td>
<td>6</td>
<td>8</td>
<td>10</td>
<td>12</td>
<td>14</td>
<td>16</td>
<td>18</td>
</tr>
</tbody>
</table>

Total Turnaround Time = (10+17+18+15) ms=60ms
Average Turnaround Time = Total Turnaround Time / Total number of processes
= (60/4) ms=15ms
Total Waiting Time = (6+11+13+12) ms=42ms
Average Waiting Time = Total Waiting Time / Total number of processes
= (42/4) ms=10.5ms

Now the proposed one is like this:
Rearrange the processes according to their burst time in ascending order and set the time slice as the average burst time of the processes.
So, according to the proposed round robin scheduling algorithm:

<table>
<thead>
<tr>
<th>Process</th>
<th>Burst time(ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>P4</td>
<td>3</td>
</tr>
<tr>
<td>P1</td>
<td>4</td>
</tr>
<tr>
<td>P3</td>
<td>5</td>
</tr>
<tr>
<td>P2</td>
<td>6</td>
</tr>
</tbody>
</table>

Time Slice = Total burst time of all the processes / Number of processes
= (3+4+5+6)/4
=18/4=4.5 ms

Gantt chart:

<table>
<thead>
<tr>
<th>P4</th>
<th>P1</th>
<th>P3</th>
<th>P2</th>
<th>P3</th>
<th>P2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>3</td>
<td>7</td>
<td>11.5</td>
<td>16</td>
<td>16.5</td>
</tr>
</tbody>
</table>

Total Turnaround Time= (7+18+16.5+3) ms=44.5ms
Average Turnaround Time=Total Turnaround Time/Total number of processes
= (44.5/4) ms =11.125ms
Total Waiting Time= (3+12+11.5+0) ms=26.5ms
Average Waiting Time=Total Waiting Time / Total number of processes
= (26.5/4) ms=6.625ms

VI. CONCLUSION
This paper provides us a new CPU scheduling algorithm which uses the concept of Round Robin scheduling algorithm with some changes. In our proposed modified Round Robin scheduling algorithm we can check that less number of context switches takes place with comparison to the traditional Round Robin scheduling algorithm as well as it
also gives the less amount of average turnaround time and average waiting time. So it is concluded that our proposed modified Round Robin scheduling algorithm is more efficient than traditional Round Robin scheduling algorithm.

VII. REFERENCES


PAPER CITATION