Lab - 4

Name - Nayansi Trikarwar

Roll no. - 64

Aim - Implement Round-Robin CPU scheduling algorithm, given the list of processes, their CPU burst times and arrival times (take inputs from the user like No. of processes etc.):

1. Display/print the Gantt chart.

2. Print waiting time and turnaround time for each process.

3. Print average waiting time and average turnaround time.

Round Robin Algorithm Code :

C: > Users > HP > RoundRobin.cpp > main()

```cpp
#include <iostream>
using namespace std;

int main()
{
    int n, timeQuantum;
    cout << "Enter the number of processes: ";
    cin >> n;
    cout << "Enter the time quantum: ";
    cin >> timeQuantum;

    int *burstTime = new int[n];
    int *remainingTime = new int[n];
    int *waitingTime = new int[n];
    int *turnaroundTime = new int[n];
    int *completionTime = new int[n];
    int ganttChart[1000];
    int gcIndex = 0;

    for (int i = 0; i < n; i++)
    {
        waitingTime[i] = 0;
    }

    for (int i = 0; i < n; i++)
    {
        cout << "Enter burst time for Process " << i + 1 << ": ";
        cin >> burstTime[i];
        remainingTime[i] = burstTime[i];
```

C: > Users > HP > RoundRobin.cpp > main()

```
4   int main()
25      for (int i = 0; i < n; i++)
30      }
31
32      int currentTime = 0;
33      int completedProcesses = 0;
34
35      while (completedProcesses < n)
36      {
37          bool allIdle = true;
38
39          for (int i = 0; i < n; i++)
40          {
41              if (remainingTime[i] > 0)
42              {
43                  allIdle = false;
44
45                  int timeSpent = min(timeQuantum, remainingTime[i]);
46                  currentTime += timeSpent;
47                  remainingTime[i] -= timeSpent;
48
49                  ganttChart[gcIndex++] = i + 1;
50
51                  if (remainingTime[i] == 0)
52                  {
53                      completedProcesses++;
54                      completionTime[i] = currentTime;
55                      turnaroundTime[i] = completionTime[i];
56                      waitingTime[i] = turnaroundTime[i] - burstTime[i];
```

C: > Users > HP > RoundRobin.cpp > main()

```
 4  int main()
35      while (completedProcesses < n)
39          for (int i = 0; i < n; i++)
41              if (remainingTime[i] > 0)
51                  if (remainingTime[i] == 0)
57                  }
58              }
59          }
60
61          if (allIdle)
62              break;
63      }
64
65      double totalWaitingTime = 0, totalTurnaroundTime = 0;
66      for (int i = 0; i < n; i++)
67      {
68          totalWaitingTime += waitingTime[i];
69          totalTurnaroundTime += turnaroundTime[i];
70      }
71
72      double avgWaitingTime = totalWaitingTime / n;
73      double avgTurnaroundTime = totalTurnaroundTime / n;
74
75      cout << "\nProcess\tBurst Time\tWaiting Time\tTurnaround Time\n";
76      for (int i = 0; i < n; i++)
77      {
78          cout << "P" << i + 1 << "\t\t" << burstTime[i]
79               << "\t\t" << waitingTime[i]
80               << "\t\t" << turnaroundTime[i] << "\n";
```

C: > Users > HP > RoundRobin.cpp > main()

```
int main()
    for (int i = 0; i < n; i++)
                << "\t\t" << turnaroundTime[i] << "\n";
    }

    cout << "\nAverage Waiting Time: " << avgWaitingTime << endl;
    cout << "Average Turnaround Time: " << avgTurnaroundTime << endl;

    cout << "\nGantt Chart:\n";
    for (int i = 0; i < gcIndex; i++)
    {
        cout << "| P" << ganttChart[i] << " ";
    }
    cout << "|\n";

    delete[] burstTime;
    delete[] remainingTime;
    delete[] waitingTime;
    delete[] turnaroundTime;
    delete[] completionTime;

    return 0;
}
```

Output :-

```
PS C:\Users\HP> cd "c:\Users\HP\" ; if ($?) { g++ RoundRobin.cpp -o RoundRobin } ; if ($?) { .\RoundRobin }
Enter the number of processes: 5
Enter the time quantum: 2
Enter burst time for Process 1: 6
Enter burst time for Process 2: 7
Enter burst time for Process 3: 8
Enter burst time for Process 4: 9
Enter burst time for Process 5: 5

Process Burst Time      Waiting Time    Turnaround Time
P1              6               16              22
P2              7               23              30
P3              8               24              32
P4              9               26              35
P5              5               24              29

Average Waiting Time: 22.6
Average Turnaround Time: 29.6

Gantt Chart:
| P1 | P2 | P3 | P4 | P5 | P1 | P2 | P3 | P4 | P5 | P1 | P2 | P3 | P4 | P5 | P2 | P3 | P4 | P4 |
PS C:\Users\HP>
```