Assignment No. 6

Name: Bhavin Patil

Roll No. 66

Preemptive Process scheduling Algorithms. Round Robin algorithms.

Code :

```
#include <bits/stdc++.h>
using namespace std;

typedef struct process
{
   int id, at, bt, st, ft, pr;
   float wt, tat;
} process;

process p[10], p1[10], temp;
queue<int> q1;

int accept(int ch);
void turnwait(int n);
void display(int n);
void ganttrr(int n);

int main()
{
   int i, n, ts, ch, j, x;

   p[0].tat = 0;
   p[0].wt = 0;

   n = accept(ch);
   ganttrr(n);
   turnwait(n);
   display(n);
```

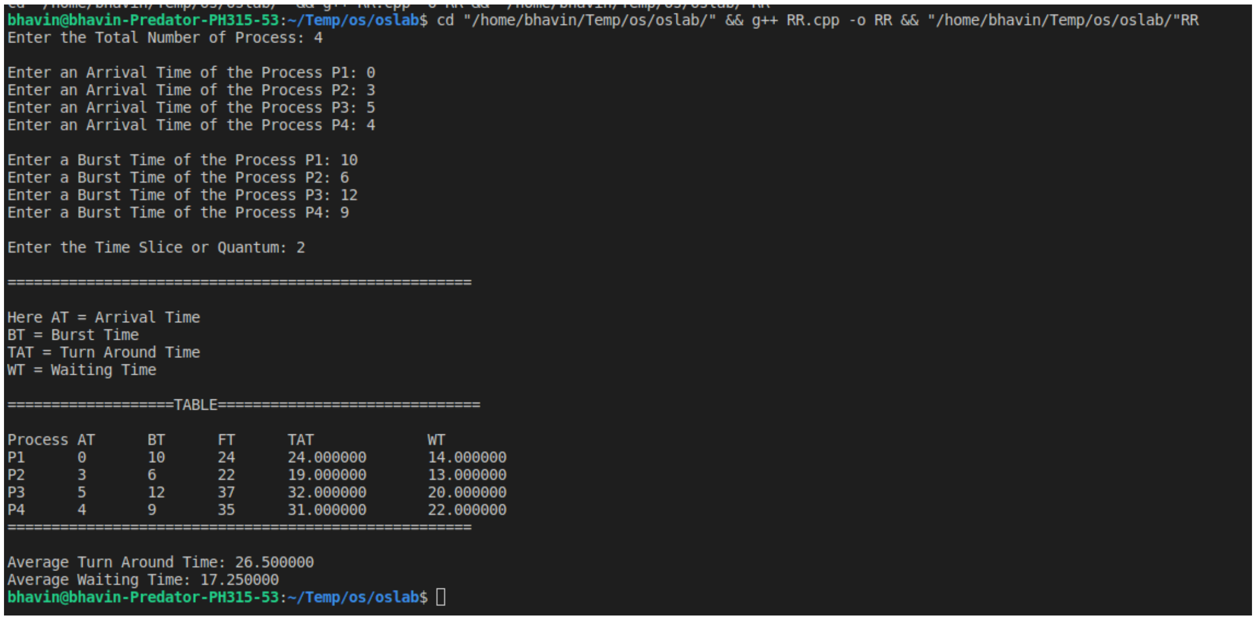
```
    return 0;
}

int accept(int ch)
{
    int i, n;

    printf("Enter the Total Number of Process: ");
    scanf("%d", &n);

    if (n == 0)
    {
        printf("Invalid");
        exit(1);
    }

    cout << endl;

    for (i = 1; i <= n; i++)
    {
        printf("Enter an Arrival Time of the Process P%d: ", i);
        scanf("%d", &p[i].at);
        p[i].id = i;
    }

    cout << endl;

    for (i = 1; i <= n; i++)
    {
        printf("Enter a Burst Time of the Process P%d: ", i);
        scanf("%d", &p[i].bt);
    }

    for (i = 1; i <= n; i++)
    {
        p1[i] = p[i];
    }
    return n;
}
```

```
void ganttrr(int n)
{
   int i, ts, m, nextval, nextarr;

   nextval = p1[1].at;
   i = 1;

   cout << "\nEnter the Time Slice or Quantum: ";
   cin >> ts;

   for (i = 1; i <= n && p1[i].at <= nextval; i++)
   {
       q1.push(p1[i].id);
   }

   while (!q1.empty())
   {
       m = q1.front();
       q1.pop();

       if (p1[m].bt >= ts)
       {
           nextval = nextval + ts;
       }
       else
       {
           nextval = nextval + p1[m].bt;
       }
       if (p1[m].bt >= ts)
       {
           p1[m].bt = p1[m].bt - ts;
       }
       else
       {
           p1[m].bt = 0;
       }

       while (i <= n && p1[i].at <= nextval)
       {
           q1.push(p1[i].id);
```

```
            i++;
        }

        if (p1[m].bt > 0)
        {
            q1.push(m);
        }
        if (p1[m].bt <= 0)
        {
            p[m].ft = nextval;
        }
    }
}

void turnwait(int n)
{
    int i;

    for (i = 1; i <= n; i++)
    {
        p[i].tat = p[i].ft - p[i].at;
        p[i].wt = p[i].tat - p[i].bt;
        p[0].tat = p[0].tat + p[i].tat;
        p[0].wt = p[0].wt + p[i].wt;
    }

    p[0].tat = p[0].tat / n;
    p[0].wt = p[0].wt / n;
}

void display(int n)
{
    int i;

    cout << "\n=====================================================\n";
    cout << "\nHere AT = Arrival Time\nBT = Burst Time\nTAT = Turn Around
Time\nWT = Waiting Time\n";

    cout << "\n==================TABLE==============================\n";
    printf("\nProcess\tAT\tBT\tFT\tTAT\t\tWT");
```

```
   for (i = 1; i <= n; i++)
   {
       printf("\nP%d\t%d\t%d\t%d\t%f\t%f", p[i].id, p[i].at, p[i].bt,
p[i].ft, p[i].tat, p[i].wt);
   }
   cout << "\n=====================================================\n";
   printf("\nAverage Turn Around Time: %f", p[0].tat);
   printf("\nAverage Waiting Time: %f\n", p[0].wt);
}
```

Output:



```
bhavin@bhavin-Predator-PH315-53:~/Temp/os/oslab$ cd "/home/bhavin/Temp/os/oslab/" && g++ RR.cpp -o RR && "/home/bhavin/Temp/os/oslab/"RR
Enter the Total Number of Process: 4

Enter an Arrival Time of the Process P1: 0
Enter an Arrival Time of the Process P2: 3
Enter an Arrival Time of the Process P3: 5
Enter an Arrival Time of the Process P4: 4

Enter a Burst Time of the Process P1: 10
Enter a Burst Time of the Process P2: 6
Enter a Burst Time of the Process P3: 12
Enter a Burst Time of the Process P4: 9

Enter the Time Slice or Quantum: 2

=====================================================

Here AT = Arrival Time
BT = Burst Time
TAT = Turn Around Time
WT = Waiting Time

===================TABLE==============================

Process AT      BT      FT      TAT             WT
P1      0       10      24      24.000000       14.000000
P2      3       6       22      19.000000       13.000000
P3      5       12      37      32.000000       20.000000
P4      4       9       35      31.000000       22.000000
=====================================================

Average Turn Around Time: 26.500000
Average Waiting Time: 17.250000
bhavin@bhavin-Predator-PH315-53:~/Temp/os/oslab$ 
```