Lab Assignment 3: CPU Scheduling

Name:Shivam Ganesh Gavandi
Roll no: 80
Class: TY-A

1)FCFS

```
#include <iostream>

using namespace std;


void findWaitingTime(int processes[], int n,

                     int bt[], int wt[])

{


   wt[0] = 0;


   for (int i = 1; i < n; i++)

       wt[i] = bt[i - 1] + wt[i - 1];

}

void findTurnAroundTime(int processes[], int n,

                        int bt[], int wt[], int tat[])

{


   for (int i = 0; i < n; i++)

       tat[i] = bt[i] + wt[i];

}


void findavgTime(int processes[], int n, int bt[])
```

```
{
   int wt[n], tat[n], total_wt = 0, total_tat = 0;

   findWaitingTime(processes, n, bt, wt);

   findTurnAroundTime(processes, n, bt, wt, tat);

   cout << "Processes  "
        << " Burst time  "
        << " Waiting time  "
        << " Turn around time\n";

   for (int i = 0; i < n; i++)
   {
       total_wt = total_wt + wt[i];
       total_tat = total_tat + tat[i];
       cout << "   " << i + 1 << "\t\t" << bt[i] << "\t    "
            << wt[i] << "\t\t  " << tat[i] << endl;
   }

   cout << "Average waiting time = "
        << (float)total_wt / (float)n;
   cout << "\nAverage turn around time = "
        << (float)total_tat / (float)n;
}

int main()
{
```

```
    int processes[] = {1, 2, 3};

    int n = sizeof processes / sizeof processes[0];


    int burst_time[] = {10, 5, 8};


    findavgTime(processes, n, burst_time);

    return 0;
}
```

Output=

Processes Burst time Waiting time Turn around time

1 10 0 10

2 5 10 15

3 8 15 23

Average waiting time = 8.33333

Average turn around time = 16

2)SJF

```
#include <stdio.h>

int main()

{

    int A[100][4];

    int i, j, n, total = 0, index, temp;

    float avg_wt, avg_tat;

    printf("Enter number of process: ");

    scanf("%d", &n);

    printf("Enter Burst Time:\n");
```

```
for (i = 0; i < n; i++)
{
    printf("P%d: ", i + 1);
    scanf("%d", &A[i][1]);
    A[i][0] = i + 1;
}

for (i = 0; i < n; i++)
{
    index = i;
    for (j = i + 1; j < n; j++)
        if (A[j][1] < A[index][1])
            index = j;
    temp = A[i][1];
    A[i][1] = A[index][1];
    A[index][1] = temp;

    temp = A[i][0];
    A[i][0] = A[index][0];
    A[index][0] = temp;
}
A[0][2] = 0;

for (i = 1; i < n; i++)
{
    A[i][2] = 0;
    for (j = 0; j < i; j++)
        A[i][2] += A[j][1];
    total += A[i][2];
```

```
    }
    avg_wt = (float)total / n;
    total = 0;
    printf("P     BT     WT     TAT\n");

    for (i = 0; i < n; i++)
    {
        A[i][3] = A[i][1] + A[i][2];
        total += A[i][3];
        printf("P%d     %d     %d      %d\n", A[i][0],
               A[i][1], A[i][2], A[i][3]);
    }
    avg_tat = (float)total / n;
    printf("Average Waiting Time= %f", avg_wt);
    printf("\nAverage Turnaround Time= %f", avg_tat);
}
```

3)round robin

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

void findWaitingTime(int processes[], int n,
                     int bt[], int wt[], int quantum)
{

    int rem_bt[n];
    for (int i = 0; i < n; i++)
        rem_bt[i] = bt[i];
```

```
int t = 0;


.while (1)
{
    bool done = true;


    for (int i = 0; i < n; i++)
    {


        if (rem_bt[i] > 0)
        {
            done = false;


            if (rem_bt[i] > quantum)
            {


                t += quantum;


                rem_bt[i] -= quantum;
            }


            else
            {


                t = t + rem_bt[i];


                wt[i] = t - bt[i];
```

```
                    rem_bt[i] = 0;

                }

            }

        }


        if (done == true)

            break;

    }

}

void findTurnAroundTime(int processes[], int n,

                        int bt[], int wt[], int tat[])

{


    for (int i = 0; i < n; i++)

        tat[i] = bt[i] + wt[i];

}


void findavgTime(int processes[], int n, int bt[],

                 int quantum)

{

    int wt[n], tat[n], total_wt = 0, total_tat = 0;


    findWaitingTime(processes, n, bt, wt, quantum);


    findTurnAroundTime(processes, n, bt, wt, tat);


    cout << "PN\t "

         << " \tBT "

         << "  WT "
```

```
        << " \tTAT\n";

    for (int i = 0; i < n; i++)
    {
        total_wt = total_wt + wt[i];
        total_tat = total_tat + tat[i];
        cout << " " << i + 1 << "\t\t" << bt[i] << "\t "
             << wt[i] << "\t\t " << tat[i] << endl;
    }

    cout << "Average waiting time = "
         << (float)total_wt / (float)n;
    cout << "\nAverage turn around time = "
         << (float)total_tat / (float)n;
}

int main()
{

    int processes[] = {1, 2, 3};
    int n = sizeof processes / sizeof processes[0];

    int burst_time[] = {10, 5, 8};

    int quantum = 2;
    findavgTime(processes, n, burst_time, quantum);
    return 0;
}
```

Output

PN	BT	WT	TAT
1	10	13	23
2	5	10	15
3	8	13	21

Average waiting time = 12

Average turn around time = 19.6667