Lab Assignment-10

ROLL: 2005535 | NAME: SAHIL SINGH | DATE: 29/03/22

QUES 1: WAP in C to implement the SJF scheduling algorithm.

SOLUTION:

```c
#include <stdio.h>
int main()
{
    int n, temp, tt = 0, min, d, i, j;
    float atat = 0, awt = 0, stat = 0, swt = 0;

    printf("Enter the number of processes: ");
    scanf("%d", &n);
    int a[n], b[n], e[n], tat[n], wt[n];

    for (i = 0; i < n; i++)
    {
        printf("Enter arival time: ");
        scanf("%d", &a[i]);
    }
    for (i = 0; i < n; i++)
    {
        printf("Enter brust time: ");
        scanf("%d", &b[i]);
    }
    for (i = 0; i < n; i++)
    {
        for (j = i + 1; j < n; j++)
        {
            if (b[i] > b[j])
            {
                temp = a[i];
                a[i] = a[j];
                a[j] = temp;

                temp = b[i];
                b[i] = b[j];
                b[j] = temp;
            }
        }
    }
    min = a[0];
    for (i = 0; i < n; i++)
    {
        if (min > a[i])
        {
            min = a[i];
            d = i;
        }
    }
    tt = min;
    e[d] = tt + b[d];
    tt = e[d];

    for (i = 0; i < n; i++)
    {
        if (a[i] != min)
        {
            e[i] = b[i] + tt;
            tt = e[i];
        }
    }
    for (i = 0; i < n; i++)
    {

        tat[i] = e[i] - a[i];
        stat = stat + tat[i];
        wt[i] = tat[i] - b[i];
        swt = swt + wt[i];
    }
    atat = stat / n;
    awt = swt / n;
    printf("Process  Arrival-time(s)  Burst-time(s)  Waiting-time(s)  Turnaround-time(s)\n");
    for (i = 0; i < n; i++)
    {
        printf("P%d\t\t%d\t\t%d\t\t%d\t\t%d\n", i + 1, a[i], b[i], wt[i], tat[i]);
    }
    printf("\nAerage Waiting Time: %.3f", awt);
    printf("\nAverage Turn Around Time: %.3f", atat);
}
```

OUTPUT:

```
Enter the number of processes: 5
Enter arival time: 3
Enter arival time: 1
Enter arival time: 4
Enter arival time: 0
Enter arival time: 2
Enter brust time: 1
Enter brust time: 4
Enter brust time: 2
Enter brust time: 6
Enter brust time: 3
Process  Arrival-time(s)  Burst-time(s)  Waiting-time(s)  Turnaround-time(s)
P1              3               1               3               4
P2              4               2               3               5
P3              2               3               7               10
```

```
P4              1               4               11              15
P5              0               6               0               6

Aerage Waiting Time: 4.800
Average Turn Around Time: 8.000
```

QUES 2: WAP in C to implement the SRTF scheduling algorithm.

SOLUTION:

```c
#include <stdio.h>
int main()
{
    int a[10], b[10], x[10];
    int waiting[10], turnaround[10], completion[10];
    int i, j, smallest, count = 0, time, n;
    double avg = 0, tt = 0, end;

    printf("Enter the number of processes: ");
    scanf("%d", &n);
    for (i = 0; i < n; i++)
    {
        printf("Enter arrival time: ");
        scanf("%d", &a[i]);
    }
    for (i = 0; i < n; i++)
    {
        printf("Enter burst time: ");
        scanf("%d", &b[i]);
    }
    for (i = 0; i < n; i++)
        x[i] = b[i];

    b[9] = 9999;
    for (time = 0; count != n; time++)
    {
        smallest = 9;
        for (i = 0; i < n; i++)
        {
            if (a[i] <= time && b[i] < b[smallest] && b[i] > 0)
                smallest = i;
        }
        b[smallest]--;

        if (b[smallest] == 0)
        {
            count++;
            end = time + 1;
            completion[smallest] = end;
            waiting[smallest] = end - a[smallest] - x[smallest];
            turnaround[smallest] = end - a[smallest];
        }
    }
    printf("Process  Burst-time  Arrival-time  Waiting-time  Turnaround-time  Completion-
time\n");
    for (i = 0; i < n; i++)
    {
        printf("P%d\t\t%d\t\t%d\t\t%d\t\t%d\t\t%d\n", i + 1, x[i], a[i], waiting[i],
turnaround[i], completion[i]);
        avg = avg + waiting[i];
        tt = tt + turnaround[i];
    }
    printf("\nAverage waiting time: %.3f", avg / n);
    printf("\nAverage Turnaround time: %.3f", tt / n);
}
```

OUTPUT:

```
Enter the number of processes: 5
Enter arrival time: 3
Enter arrival time: 1
Enter arrival time: 4
Enter arrival time: 0
Enter arrival time: 2
Enter burst time: 1
Enter burst time: 4
Enter burst time: 2
Enter burst time: 6
Enter burst time: 3
Process  Burst-time  Arrival-time  Waiting-time  Turnaround-time  Completion-time
P1              1               3               0               1               4
P2              4               1               1               5               6
P3              2               4               2               4               8
P4              6               0               10              16              16
P5              3               2               6               9               11

Average waiting time: 3.800
Average Turnaround time: 7.000
```
