Lab Assignment 10

# Priority Scheduling (Non-Preemptive)

#include <stdio.h>

int main()

{

    int bt[20], p[20], wt[20], tat[20], pr[20], i, j, n, total = 0, pos, temp, avg\_wt, avg\_tat;

    printf("Enter Total Number of Process:");

    scanf("%d", &n);

    printf("\nEnter Burst Time and Priority\n");

    for (i = 0; i < n; i++)

    {

        printf("\nP[%d]\n", i + 1);

        printf("Burst Time:");

        scanf("%d", &bt[i]);

        printf("Priority:");

        scanf("%d", &pr[i]);

        p[i] = i + 1; *// contains process number*

    }

*// sorting burst time, priority and process number in ascending order using selection sort*

    for (i = 0; i < n; i++)

    {

        pos = i;

        for (j = i + 1; j < n; j++)

        {

            if (pr[j] < pr[pos])

                pos = j;

        }

        temp = pr[i];

        pr[i] = pr[pos];

        pr[pos] = temp;

        temp = bt[i];

        bt[i] = bt[pos];

        bt[pos] = temp;

        temp = p[i];

        p[i] = p[pos];

        p[pos] = temp;

    }

    wt[0] = 0; *// waiting time for first process is zero*

*// calculate waiting time*

    for (i = 1; i < n; i++)

    {

        wt[i] = 0;

        for (j = 0; j < i; j++)

            wt[i] += bt[j];

        total += wt[i];

    }

    avg\_wt = total / n; *// average waiting time*

    total = 0;

    printf("\nProcess\t    Burst Time    \tWaiting Time\tTurnaround Time");

    for (i = 0; i < n; i++)

    {

        tat[i] = bt[i] + wt[i]; *// calculate turnaround time*

        total += tat[i];

        printf("\nP[%d]\t\t  %d\t\t    %d\t\t\t%d", p[i], bt[i], wt[i], tat[i]);

    }

    avg\_tat = total / n; *// average turnaround time*

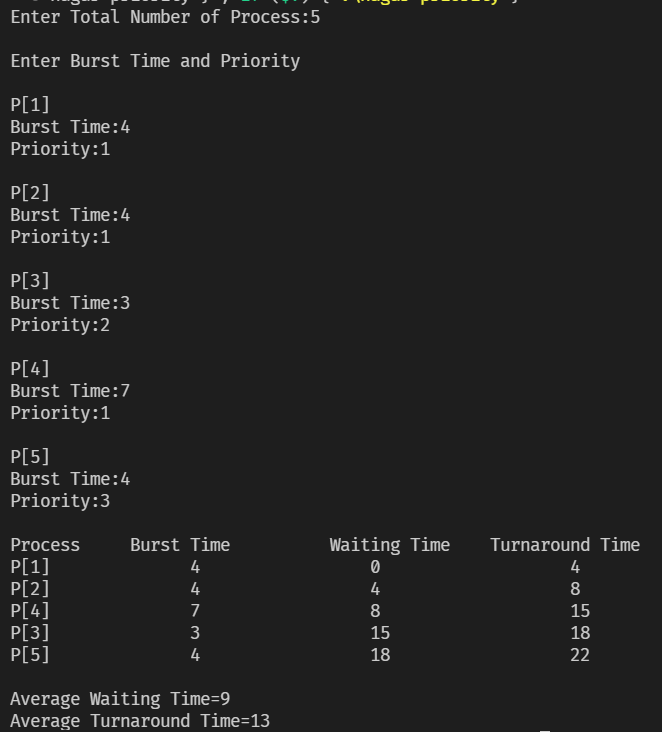
    printf("\n\nAverage Waiting Time=%d", avg\_wt);

    printf("\nAverage Turnaround Time=%d\n", avg\_tat);

    return 0;

}

Output



Priority Scheduling (Preemptive)

#include <stdio.h>

struct process

{

    int WT, AT, BT, TAT, PT;

};

struct process a[10];

int main()

{

    int n, temp[10], t, count = 0, short\_p;

    float total\_WT = 0, total\_TAT = 0, Avg\_WT, Avg\_TAT;

    printf("Enter the number of the process\n");

    scanf("%d", &n);

    printf("Enter the arrival time , burst time and priority of the process\n");

    printf("AT BT PT\n");

    for (int i = 0; i < n; i++)

    {

        scanf("%d%d%d", &a[i].AT, &a[i].BT, &a[i].PT);

        temp[i] = a[i].BT;

    }

    a[9].PT = 10000;

    for (t = 0; count != n; t++)

    {

        short\_p = 9;

        for (int i = 0; i < n; i++)

        {

            if (a[short\_p].PT > a[i].PT && a[i].AT <= t && a[i].BT > 0)

            {

                short\_p = i;

            }

        }

        a[short\_p].BT = a[short\_p].BT - 1;

        if (a[short\_p].BT == 0)

        {

            count++;

            a[short\_p].WT = t + 1 - a[short\_p].AT - temp[short\_p];

            a[short\_p].TAT = t + 1 - a[short\_p].AT;

            total\_WT = total\_WT + a[short\_p].WT;

            total\_TAT = total\_TAT + a[short\_p].TAT;

        }

    }

    Avg\_WT = total\_WT / n;

    Avg\_TAT = total\_TAT / n;

    printf("ID\tAT\tWT\tTAT\tPR \n");

    for (int i = 0; i < n; i++)

    {

        printf("%d\t%d\t%d\t%d\t%d\n", i + 1, a[i].AT, a[i].WT, a[i].TAT, a[i].PT);

    }

    printf("Avg waiting time of the process  is %f\n", Avg\_WT);

    printf("Avg turn around time of the process is %f\n", Avg\_TAT);

    return 0;

}

Output

