Lab Assignment 10

# **Priority Scheduling (Non-Preemptive)**

#include <stdio.h>

struct process

{

    int id, WT, AT, BT, TAT, PR;

};

struct process a[10];

void swap(int \*b, int \*c)

{

    int tem;

    tem = \*c;

    \*c = \*b;

    \*b = tem;

}

int main()

{

    int n, check\_ar = 0;

    int Cmp\_time = 0;

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

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

    scanf("%d", &n);

    printf("Enter the Arrival time , Burst time and priority of the process\n");

    printf("AT BT PR\n");

    for (int i = 0; i < n; i++)

    {

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

        a[i].id = i + 1;

        if (i == 0)

            check\_ar = a[i].AT;

        if (check\_ar != a[i].AT)

            check\_ar = 1;

    }

    if (check\_ar != 0)

    {

        for (int i = 0; i < n; i++)

        {

            for (int j = 0; j < n - i - 1; j++)

            {

                if (a[j].AT > a[j + 1].AT)

                {

                    swap(&a[j].id, &a[j + 1].id);

                    swap(&a[j].AT, &a[j + 1].AT);

                    swap(&a[j].BT, &a[j + 1].BT);

                    swap(&a[j].PR, &a[j + 1].PR);

                }

            }

        }

    }

    if (check\_ar != 0)

    {

        a[0].WT = a[0].AT;

        a[0].TAT = a[0].BT - a[0].AT;

        Cmp\_time = a[0].TAT;

        Total\_WT = Total\_WT + a[0].WT;

        Total\_TAT = Total\_TAT + a[0].TAT;

        for (int i = 1; i < n; i++)

        {

            int min = a[i].PR;

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

            {

                if (min > a[j].PR && a[j].AT <= Cmp\_time)

                {

                    min = a[j].PR;

                    swap(&a[i].id, &a[j].id);

                    swap(&a[i].AT, &a[j].AT);

                    swap(&a[i].BT, &a[j].BT);

                    swap(&a[i].PR, &a[j].PR);

                }

            }

            a[i].WT = Cmp\_time - a[i].AT;

            Total\_WT = Total\_WT + a[i].WT;

            Cmp\_time = Cmp\_time + a[i].BT;

            a[i].TAT = Cmp\_time - a[i].AT;

            Total\_TAT = Total\_TAT + a[i].TAT;

        }

    }

    else

    {

        for (int i = 0; i < n; i++)

        {

            int min = a[i].PR;

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

            {

                if (min > a[j].PR && a[j].AT <= Cmp\_time)

                {

                    min = a[j].PR;

                    swap(&a[i].id, &a[j].id);

                    swap(&a[i].AT, &a[j].AT);

                    swap(&a[i].BT, &a[j].BT);

                    swap(&a[i].PR, &a[j].PR);

                }

            }

            a[i].WT = Cmp\_time - a[i].AT;

            Cmp\_time = Cmp\_time + a[i].BT;

            a[i].TAT = Cmp\_time - a[i].AT;

            Total\_WT = Total\_WT + a[i].WT;

            Total\_TAT = Total\_TAT + a[i].TAT;

        }

    }

    Avg\_WT = Total\_WT / n;

    Avg\_TAT = Total\_TAT / n;

    printf("The process are\n");

    printf("PID \tAT \tWT \tTAT\tPR\n");

    for (int i = 0; i < n; i++)

    {

        printf("%d\t%d\t%d\t%d%d\t\n", a[i].id, a[i].AT, a[i].WT, a[i].TAT, a[i].PR);

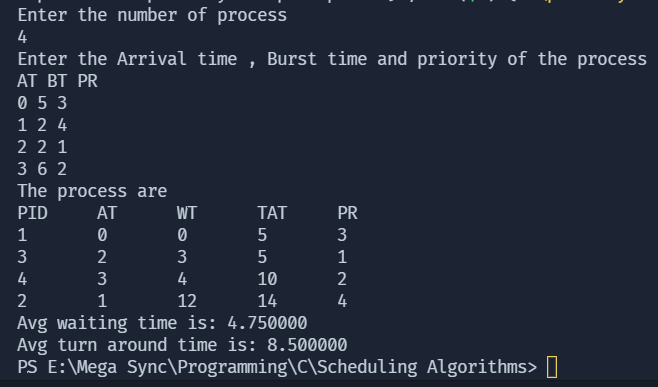
    }

    printf("Avg waiting time is: %f\n", Avg\_WT);

    printf("Avg turn around time is: %f", 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

