First Come First Serve

#include <stdio.h>

#include <stdlib.h>

struct process\_struct

{

    int pid;

    int at;

    int bt;

    int ct, wt, tat, rt, start\_time;

} ps[20];

int comparatorAT(const void \*a, const void \*b)

{

    int x = ((struct process\_struct \*)a)->at;

    int y = ((struct process\_struct \*)b)->at;

    if (x < y)

        return -1;

    else if (x >= y)

        return 1;

}

int max(int a, int b)

{

    return a > b ? a : b;

}

int main()

{

    int n;

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

    scanf("%d", &n);

    float sum\_tat = 0, sum\_wt = 0;

    for (int i = 0; i < n; i++)

    {

        printf("Enter Process %d Arrival Time: ", i);

        scanf("%d", &ps[i].at);

        ps[i].pid = i;

    }

    for (int i = 0; i < n; i++)

    {

        printf("\nEnter Process %d Burst Time: ", i);

        scanf("%d", &ps[i].bt);

    }

    for (int i = 0; i < n; i++)

    {

        ps[i].start\_time = (i == 0) ? ps[i].at : max(ps[i].at, ps[i - 1].ct);

        ps[i].ct = ps[i].start\_time + ps[i].bt;

        ps[i].tat = ps[i].ct - ps[i].at;

        ps[i].wt = ps[i].tat - ps[i].bt;

        sum\_tat = sum\_tat + ps[i].tat;

        sum\_wt = sum\_wt + ps[i].wt;

    }

    qsort((void \*)ps, n, sizeof(struct process\_struct), comparatorAT);

    printf("\nProcess No. \tAT\tBurst Time\tCT\tTAT\tWT\n");

    for (int i = 0; i < n; i++)

    {

        printf("%d\t\t%d\t%d\t\t%d\t%d\t%d\n", ps[i].pid, ps[i].at, ps[i].bt, ps[i].ct, ps[i].tat, ps[i].wt);

        printf("\n");

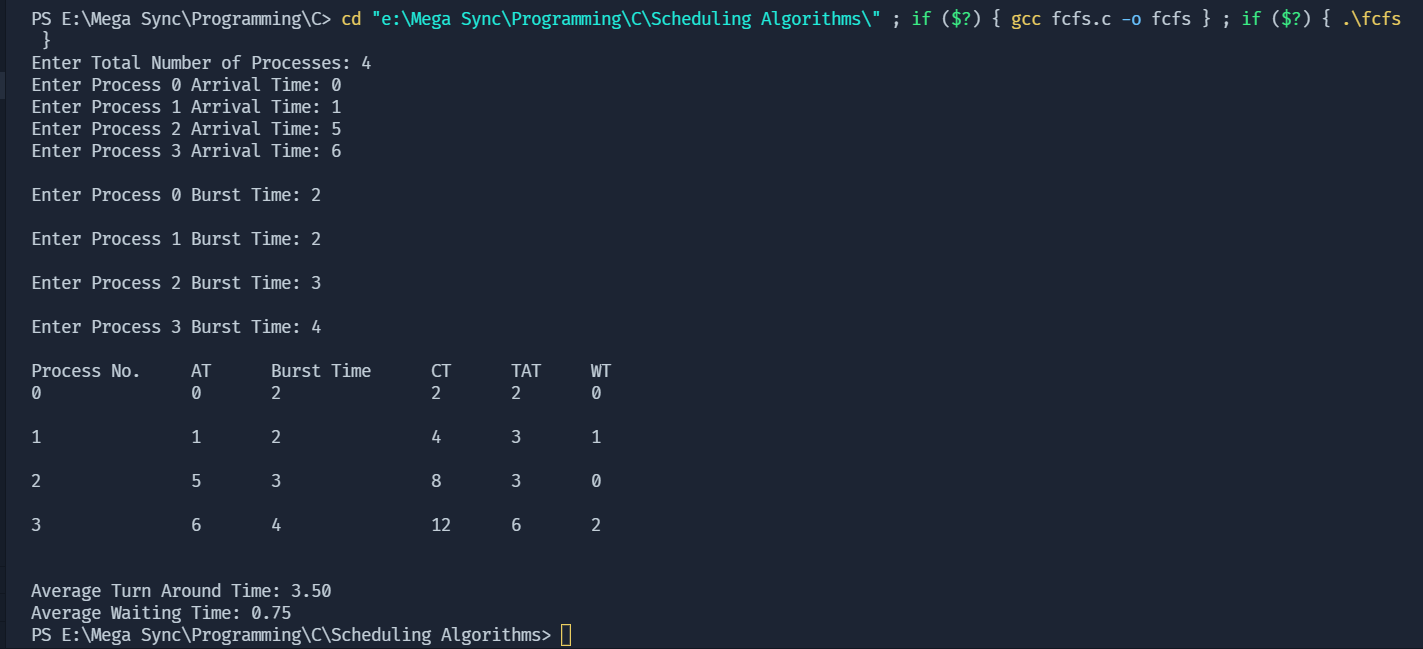
    }

    printf("\nAverage Turn Around Time: %.2f", sum\_tat / n);

    printf("\nAverage Waiting Time: %.2f", sum\_wt / n);

    return 0;

}

Output