Lab -2(fcfs)

#include<stdio.h>

void sort(int processes[],int n,int at[],int bt[])

{

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

{

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

{

if(at[j]>at[j+1])

{

int temp=at[j];

at[j]=at[j+1];

at[j+1]=temp;

temp=bt[j];

bt[j]=bt[j+1];

bt[j+1]=temp;

temp = processes[j];

processes[j]=processes[j+1];

processes[j+1]=temp;

}

}

}

}

void calculateTimes(int processes[], int n, int bt[], int at[], int wt[], int tat[], int rt[]) {

int service\_time[n];

service\_time[0] = at[0];

wt[0] = 0;

rt[0] = 0;

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

service\_time[i] = service\_time[i - 1] + bt[i - 1];

wt[i] = service\_time[i] - at[i];

if (wt[i] < 0)

wt[i] = 0;

rt[i] = service\_time[i] - at[i];

}

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

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

}

}

void displayResults(int processes[], int n, int at[], int bt[], int wt[], int tat[], int rt[]) {

int total\_wt = 0, total\_tat = 0, total\_rt = 0;

printf("\nProcess\tArrival Time\tBurst Time\tWaiting Time\tTurnaround Time\tResponse Time\n");

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

total\_wt += wt[i];

total\_tat += tat[i];

total\_rt += rt[i];

printf("%d\t%d\t\t%d\t\t%d\t\t%d\t\t%d\n", processes[i], at[i], bt[i], wt[i], tat[i], rt[i]);

}

printf("\nAverage Waiting Time = %.2f", (float)total\_wt / n);

printf("\nAverage Turnaround Time = %.2f", (float)total\_tat / n);

printf("\nAverage Response Time = %.2f\n", (float)total\_rt / n);

}

int main() {

int n;

printf("Enter the number of processes: ");

scanf("%d", &n);

int processes[n], arrival\_time[n], burst\_time[n], waiting\_time[n], turnaround\_time[n], response\_time[n];

printf("Enter Process IDs, Arrival Time, and Burst Time:\n");

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

printf("Process %d: ", i + 1);

scanf("%d %d %d", &processes[i], &arrival\_time[i], &burst\_time[i]);

}

sort(processes,n,arrival\_time,burst\_time);

calculateTimes(processes, n, burst\_time, arrival\_time, waiting\_time, turnaround\_time, response\_time);

displayResults(processes, n, arrival\_time, burst\_time, waiting\_time, turnaround\_time, response\_time);

return 0;

}

