Program-3

AIM: To write a c program to simulate the CPU scheduling algorithm First Come First Serve (FCFS)

ALGORITHM:

Step 1: Start the process

Step 2: Accept the number of processes in the ready Queue

Step 3: For each process in the ready Q, assign the process name and the burst time

Step 4: Set the waiting of the first process as ‗0‘and its burst time as its turnaround time

Step 5: for each process in the Ready Q calculate a). Waiting time (n) = waiting time (n-1) + Burst time (n-1) b).Turnaround time (n)= waiting time(n)+Burst time(n)

Step 6: Calculate a) Average waiting time = Total waiting Time / Number ofprocess b) Average Turnaround time = Total Turnaround Time / Number ofprocess

Step 7: Stop the process

**Code:**

#include <stdio.h>

#define MAX 20

int main(void) {

//Priority queue using array

int readyQueue[MAX];//AT of processes

int burstTime[MAX];//BT of processes

int waitingTime = 0;//WT

int numProcess = 0;//Number of processes

int tat = 0;//Turn around time

int totalTAT = 0;

int totalWT = 0;

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

scanf("%d", &numProcess);

printf("\nEnter the arrival time : ");//Input the arrival time

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

scanf("%d", &readyQueue[i]);

}

printf("\nEnter the burst time :\n");//Input the burst time

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

scanf("%d", &burstTime[i]);

}

for(int i = 0; i<numProcess-1; i++){//Sorting for converting the queue into priority queue

for(int j = i+1; j<numProcess; j++){

if(readyQueue[j]>readyQueue[j+1]){

int temp1 = burstTime[j];

burstTime[j] = burstTime[j+1];

burstTime[j+1] = temp1;

}

}

}

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

tat = 0;

if(i==0){//Waiting time of first process = 0

printf("WT\t\t\tTAT\n");

printf("0\t\t\t");

tat += waitingTime + burstTime[i];

printf("%d\n", tat);

}

if(i>0){

waitingTime += burstTime[i-1];//WT(n) = WT(n-1) + BT(n-1)

printf("%d\t\t\t", waitingTime);

tat += waitingTime + burstTime[i];//TAT(n) = WT(n) + BT(n)

printf("%d\n", tat);

}

totalWT += waitingTime;

totalTAT += tat;

}

float avWT = 0.0;

avWT = totalWT/numProcess;//Average waiting time

float avTAT = 0.0;

avTAT = totalTAT/numProcess;//Average turn around time

printf("Average waiting time = %.2f\n", avWT);

printf("Average turn around time = %.2f", avTAT);

return 0;

}

**Output:**

