Assignment 2a

Code:

//FCFS

#include<iostream>

#include<algorithm>

using namespace std;

struct Process {

int processID;

int burstTime;

int arrivalTime;

int ct;

};

void findWaitingTime(Process processes[], int n, int wt[]) {

sort(processes, processes + n, [](const Process& a, const Process& b) {

return a.arrivalTime < b.arrivalTime;

});

int ct = 0;

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

ct += processes[i].burstTime;

processes[i].ct = ct; /

wt[i] = ct-processes[i].burstTime - processes[i].arrivalTime;

}

}

void findTurnAroundTime(Process processes[], int n, int wt[], int tat[]) {

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

tat[i] = processes[i].burstTime + wt[i];

}

void findavgTime(Process processes[], int n) {

int wt[n], tat[n], total\_wt = 0, total\_tat = 0;

findWaitingTime(processes, n, wt);

findTurnAroundTime(processes, n, wt, tat);

cout << "Processes " << " Arrival time " << " Burst time "

<< " Waiting time " << " Turn around time\n";

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

total\_wt = total\_wt + wt[i];

total\_tat = total\_tat + tat[i];

cout << " " << processes[i].processID << "\t\t" << processes[i].arrivalTime << "\t\t"

<< processes[i].burstTime << "\t " << wt[i] << "\t\t " << tat[i] << endl;

}

cout << "Average waiting time = " << (float)total\_wt / (float)n;

cout << "\nAverage turn around time = " << (float)total\_tat / (float)n;

}

int main() {

int n;

cout << "Enter the number of processes: ";

cin >> n;

Process processes[n];

cout << "Enter burst time and arrival time for each process:\n";

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

cout << "Enter burst time for process " << i + 1 << ": ";

cin >> processes[i].burstTime;

cout << "Enter arrival time for process " << i + 1 << ": ";

cin >> processes[i].arrivalTime;

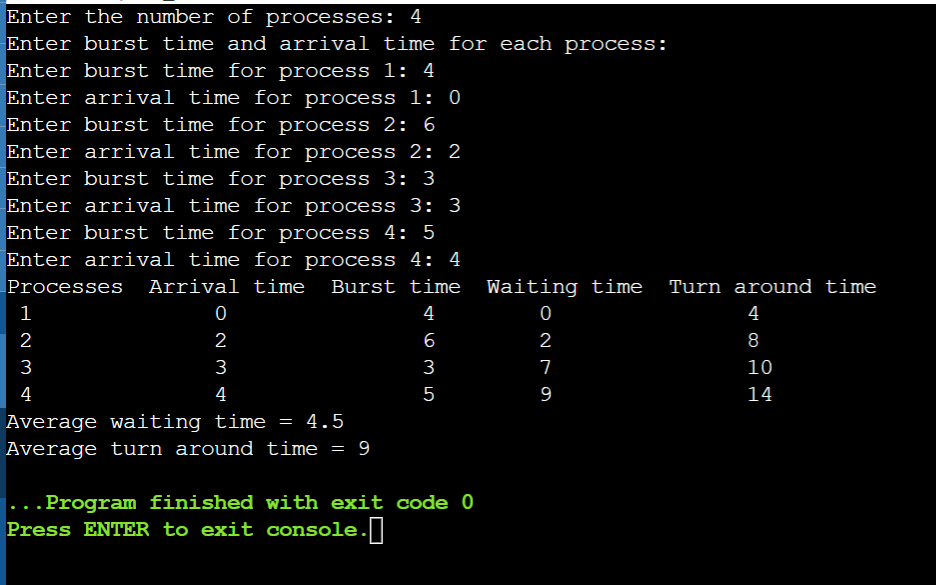
processes[i].processID = i + 1;

}

findavgTime(processes, n);

return 0;

}

Output:

Code:

//SJF

#include <iostream>

using namespace std;

int main() {

int A[100][4];

int i, j, n, total = 0, index, temp;

float avg\_wt, avg\_tat;

cout << "Enter number of process: ";

cin >> n;

cout << "Enter Burst Time:" << endl;

// User Input Burst Time and alloting Process Id.

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

cout << "P" << i + 1 << ": ";

cin >> A[i][1];

A[i][0] = i + 1;

}

// Sorting process according to their Burst Time.

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

index = i;

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

if (A[j][1] < A[index][1])

index = j;

temp = A[i][1];

A[i][1] = A[index][1];

A[index][1] = temp;

temp = A[i][0];

A[i][0] = A[index][0];

A[index][0] = temp;

}

A[0][2] = 0;

// Calculation of Waiting Times

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

A[i][2] = 0;

for (j = 0; j < i; j++)

A[i][2] += A[j][1];

total += A[i][2];

}

avg\_wt = (float)total / n;

total = 0;

cout << "P BT WT TAT" << endl;

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

A[i][3] = A[i][1] + A[i][2];

total += A[i][3];

cout << "P" << A[i][0] << " " << A[i][1] << " " << A[i][2] << " " << A[i][3] << endl;

}

avg\_tat = (float)total / n;

cout << "Average Waiting Time= " << avg\_wt << endl;

cout << "Average Turnaround Time= " << avg\_tat << endl;

}

Output:

