26 CPU Burst TIme

#include <stdio.h>

#define MAX\_PROCESS 3

struct Process {

int burstTime;

int priority;

};

void calculateWaitingTime(struct Process processes[], int n, int waitingTime[]) {

waitingTime[0] = 0;

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

waitingTime[i] = processes[i - 1].burstTime + waitingTime[i - 1];

}

}

void calculateTurnaroundTime(struct Process processes[], int n, int waitingTime[], int turnaroundTime[]) {

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

turnaroundTime[i] = processes[i].burstTime + waitingTime[i];

}

}

void calculateAverageTime(struct Process processes[], int n) {

int waitingTime[MAX\_PROCESS], turnaroundTime[MAX\_PROCESS], totalWaitingTime = 0, totalTurnaroundTime = 0;

calculateWaitingTime(processes, n, waitingTime);

calculateTurnaroundTime(processes, n, waitingTime, turnaroundTime);

printf("Process\tBurst Time\tPriority\tWaiting Time\tTurnaround Time\n");

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

totalWaitingTime += waitingTime[i];

totalTurnaroundTime += turnaroundTime[i];

printf("P%d\t\t%d\t\t%d\t\t%d\t\t%d\n", i + 1, processes[i].burstTime, processes[i].priority, waitingTime[i], turnaroundTime[i]);

}

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

printf("Average Turnaround Time: %.2f\n", (float)totalTurnaroundTime / n);

}

int main() {

struct Process processes[MAX\_PROCESS] = {

{30, 2},

{5, 1},

{12, 3}

};

calculateAverageTime(processes, MAX\_PROCESS);

return 0;

}

OUTPUT

