SJF SCHEDULING

#include<stdio.h>

#include<stdlib.h>

typedef struct {

int pid;

int btime;

int wtime;

} sp;

int main() {

int i, j, n, tbm = 0, totwtime = 0, totttime = 0;

sp \*p, t;

printf("\nSJF Scheduling..\n");

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

scanf("%d", &n);

p = (sp\*)malloc(n \* sizeof(sp));

printf("\nEnter the burst time\n");

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

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

scanf("%d", &p[i].btime);

p[i].pid = i + 1;

p[i].wtime = 0;

}

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

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

if(p[i].btime > p[j].btime) {

t = p[i];

p[i] = p[j];

p[j] = t;

}

}

}

printf("\nProcess Scheduling\n");

printf("\nProcess\tBurst Time\tWaiting Time");

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

totwtime += p[i].wtime = tbm;

tbm += p[i].btime;

printf("\n%d\t\t%d\t\t%d", p[i].pid, p[i].btime, p[i].wtime);

}

totttime = tbm + totwtime;

printf("\nTotal Waiting Time: %d", totwtime);

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

printf("\nTotal Turnaround Time: %d", totttime);

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

free(p); // Free allocated memory

return 0;

}

OUTPUT:

