Name: Shalu

Reg Num: 11804949

GitHub link: [**https://github.com/Shalu9951/OS-Assignments**](https://github.com/Shalu9951/OS-Assignments)

#include <stdio.h>

#include <stdlib.h>

#define LAST 100

typedef struct

{

int pid, burst, wait, turntime;

} Process;

void print\_table(Process p[], int n);

void print\_gantt\_chart(Process p[], int n);

int main()

{

Process p[LAST];

int i, j, n;

int sum\_wait = 0, sum\_turntime;

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

scanf("%d", &n);

printf("Enter burst time for each process:\n");

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

p[i].pid = i+1;

printf("P[%d] : ", i+1);

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

p[i].wait = p[i].turntime = 0;

}

// calculate waiting time and turnaround time

p[0].turntime = p[0].burst;

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

p[i].wait = p[i-1].wait + p[i-1].burst;

p[i].turntime = p[i].wait + p[i].burst;

}

// calculate sum of waiting time and sum of turnaround time

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

sum\_wait += p[i].wait;

sum\_turntime += p[i].turntime;

}

// print table

puts(""); // Empty line

print\_table(p, n);

puts(""); // Empty Line

printf("Waiting Time is : %-2d\n", sum\_wait);

printf("Average Waiting Time is : %-2.2lf\n", (double)sum\_wait / (double) n);

printf("Total Turnaround Time is : %-2d\n", sum\_turntime);

printf("Average Turnaround Time : %-2.2lf\n", (double)sum\_turntime / (double) n);

// print Gantt chart

puts(""); // Empty line

puts(" GANTT CHART ");

puts(" \*\*\*\*\*\*\*\*\*\*\* ");

print\_gantt\_chart(p, n);

return 0;

}

void print\_table(Process p[], int n)

{

int i;

puts("+-----+------------+--------------+-----------------+");

puts("| PID | Burst Time | Waiting Time | Turnaround Time |");

puts("+-----+------------+--------------+-----------------+");

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

printf("| %2d | %2d | %2d | %2d |\n"

, p[i].pid, p[i].burst, p[i].wait, p[i].turntime );

puts("+-----+------------+--------------+-----------------+");

}

}

void print\_gantt\_chart(Process p[], int n)

{

int i, j;

// print top bar

printf(" ");

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

for(j=0; j<p[i].burst; j++) printf("--");

printf(" ");

}

printf("\n|");

// printing process id in the middle

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

for(j=0; j<p[i].burst - 1; j++) printf(" ");

printf("P%d", p[i].pid);

for(j=0; j<p[i].burst - 1; j++) printf(" ");

printf("|");

}

printf("\n ");

// printing bottom bar

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

for(j=0; j<p[i].burst; j++) printf("--");

printf(" ");

}

printf("\n");

// printing the time line

printf("0");

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

for(j=0; j<p[i].burst; j++) printf(" ");

if(p[i].turntime > 9) printf("\b"); // backspace : remove 1 space

printf("%d", p[i].turntime);

}

printf("\n");

}