**Name : Shalu Reg. : 11804949 Section:K18TS**

**OS Assignment**

**Submitted To: Shivali Chopra Mam**

**C Program for the Cats and Mice problem.**

#include <stdio.h>

#include <stdlib.h>

#define MAX 100

typedef struct

{

int pid;

int burst\_time;

int waiting\_time;

int turnaround\_time;

} Process;

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

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

int main()

{

Process p[MAX];

int i, j, n;

int sum\_waiting\_time = 0, sum\_turnaround\_time;

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\_time);

p[i].waiting\_time = p[i].turnaround\_time = 0;

}

// calculate waiting time and turnaround time

p[0].turnaround\_time = p[0].burst\_time;

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

p[i].waiting\_time = p[i-1].waiting\_time + p[i-1].burst\_time;

p[i].turnaround\_time = p[i].waiting\_time + p[i].burst\_time;

}

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

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

sum\_waiting\_time += p[i].waiting\_time;

sum\_turnaround\_time += p[i].turnaround\_time;

}

// print table

puts(""); // Empty line

print\_table(p, n);

puts(""); // Empty Line

printf("Total Waiting Time : %-2d\n", sum\_waiting\_time);

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

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

printf("Average Turnaround Time : %-2.2lf\n", (double)sum\_turnaround\_time / (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\_time, p[i].waiting\_time, p[i].turnaround\_time );

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\_time; j++) printf("--");

printf(" ");

}

printf("\n|");

// printing process id in the middle

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

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

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

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

printf("|");

}

printf("\n ");

// printing bottom bar

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

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

printf(" ");

}

printf("\n");

// printing the time line

printf("0");

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

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

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

printf("%d", p[i].turnaround\_time);

}

printf("\n");

}

**Github Revision:**

* I have made 3 revision of code on Github.

**Github link** [**https://github.com/Shalu9951/OS-Assignmen**](https://github.com/Shalu9951/OS-Assignmen)**t**