**Practical : 5**

**Aim: To implement SJF (Shortest Job First) CPU Scheduling Algorithm.**

**Program:**

**#include <stdio.h>**

**struct process**

**{**

**int pid; int btime; int wtime; int ttime; int arrtime;**

**} p[10], temp;**

**int main()**

**{**

**int i,j,k,n;**

**float awat,atur,ttur,twat,totaltime=0,throughput;**

**int cpu\_util=0;**

**printf("Enter no. of process : ");**

**scanf("%d", &n);**

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

**{**

**printf("Burst time for process P%d (in ms) : ",(i+1));**

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

**p[i].pid = i+1;**

**}**

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

**{**

**printf("enter arrival time for process P%d (in ms) : ",(i+1));**

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

**p[i].pid = i+1;**

**}**

**for(i=0; i<n-1; i++)**

**{**

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

**{**

**if((p[i].btime > p[j].btime) || (p[i].btime == p[j].btime && p[i].pid > p[j].pid))**

**{**

**temp = p[i];**

**p[i] = p[j];**

**p[j] = temp;**

**}**

**}**

**}**

**p[0].wtime = 0;**

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

**{**

**p[i+1].wtime = p[i].wtime + p[i].btime;**

**p[i].ttime = p[i].wtime + p[i].btime;**

**totaltime += p[i].btime;**

**}**

**ttur = twat = 0;**

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

**{**

**ttur += p[i].ttime;**

**twat += p[i].wtime;**

**}**

**awat = (float)twat / n;**

**atur = (float)ttur / n;**

**printf("\n\nGANTT Chart:\n\n");**

**for(i=1; i<n\*8+2; i++)**

**{**

**printf("-");**

**}**

**printf("\n|");**

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

**{**

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

**printf("|");**

**}**

**printf("\n");**

**for(i=1; i<n\*8+2; i++)**

**{**

**printf("-");**

**}**

**printf("\n0");**

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

**{**

**if(p[i].ttime>10)**

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

**else**

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

**}**

**printf("\n\n\n...............SJF Scheduling...............\n\n");**

**for(i=0; i<60; i++)**

**{**

**printf("-");**

**}**

**printf("\n| Time | Process | T-Time= | W-Time= |\n");**

**printf("| taken | Completed | t(PC)-t(PS) | T.T - T(process time) | \n");**

**for(i=0; i<60; i++)**

**{**

**printf("-");**

**}**

**printf("\n");**

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

**{**

**printf("| %2d\t| P%d |\t%5d | \t%5d |",p[i].btime,p[i].pid,p[i].ttime,p[i].wtime);**

**printf("\n+----------------------------------------------------------+\n");**

**}**

**/\* for(i=0; i<60; i++)**

**{**

**printf("-");**

**}\*/**

**printf("\n");**

**printf("\n");**

**printf("\n\nAverage waiting time given by: total waiting time / no. of processes \nSo here: ");**

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

**{**

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

**for(j=0; j<n-1; j++)**

**{**

**printf("+");**

**break;**

**}**

**}**

**printf("/ %d = %5.2f ms \n",n,awat);**

**printf("\n\nAverage Turn-around time given by: total turn-around time / no. of processes \nSo here: ");**

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

**{**

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

**for(j=0; j<n-1; j++)**

**{**

**printf("+");**

**break;**

**}**

**}**

**printf("/ %d = %5.2f ms \n",n,atur);**

**cpu\_util= ( totaltime/ (totaltime + 0) ) \* 100;**

**printf("\n\nCPU utilization, given by: (cpu busy time)/(cpu busytime+ idle time) \nSo here it is %5.2f / (%5.2f + 0) = %d\n\n",totaltime,totaltime, cpu\_util);**

**throughput= n/ ( totaltime);**

**printf("\nThroughput, given by: No. of Processes/ Totaltime,\nSo here it is: %d / %5.2f = %5.2f processes per ms",n,totaltime,throughput);**

**return 0;**

**}**

**Output:**

