OS ASSIGNMENT CH SIVA VARMA K18QW ROLLNO 22 REGNO 11803635

GITHUB LINK:

CODE:

1.Write a program in C which reads input CPU bursts from a the first line of a text file named as CPU\_BURST.txt. Validate the input numbers whether the numbers are positive intergers or not. Consider the numbers as CPU burst.If there are 5 positive integers in the first line of the text file then the program treat those argument as required CPU bust for P1, P2, P3, P4, and P5 process and calculate average waiting time and average turn around time. Consider used scheduling algorithm as SJF and same arrival time for all the processes.

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

int main()

{

FILE \*fp = fopen("cpu\_burst.txt", "r");

int bt[20],p[20],wt[20],tat[20],i=0,j,n=5,total=0,pos,temp;

float avg\_wt,avg\_tat;

printf("\nReading CPU\_BURST.txt File\n");

//for(i=0;i<5;i++)

while((getc(fp))!=EOF)

{

fscanf(fp, "%d", &bt[i]);

if(bt[i]>0){

p[i]=i+1; i++;} //contains process number

}

n=i;

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

{

pos=i;

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

{

if(bt[j]<bt[pos])

pos=j;

}

temp=bt[i];

bt[i]=bt[pos];

bt[pos]=temp;

temp=p[i];

p[i]=p[pos];

p[pos]=temp;

}

wt[0]=0; //waiting time for first process will be zero

//calculate waiting time

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

{

wt[i]=0;

for(j=0;j<i;j++)

wt[i]+=bt[j];

total+=wt[i];

}

avg\_wt=(float)total/n; //average waiting time

total=0;

printf("\nProcess\t Burst Time \tWaiting Time\tTurnaround Time");

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

{

tat[i]=bt[i]+wt[i]; //calculate turnaround time

total+=tat[i];

printf("\np%d\t\t %d\t\t %d\t\t\t%d",p[i],bt[i],wt[i],tat[i]);

}

avg\_tat=(float)total/n; //average turnaround time

printf("\n\nAverage Waiting Time=%f",avg\_wt);

printf("\nAverage Turnaround Time=%f\n",avg\_tat);

fclose(fp);

return 0;

}

2ND question:

|  |
| --- |
| #include<stdio.h> |
|  |  |
|  | void sort(); |
|  | void calcwt(); |
|  | void printq(); |
|  |  |
|  | int bt[10],waittime[10],pdata[10],priority[10]; |
|  | int n, avgwt=0; |
|  | float avgtat=0; |
|  |  |
|  | int main() |
|  | { |
|  | int i,j; |
|  | printf("\nShortest Job First Scheduling:\n"); |
|  | printf("\nEnter the number of process: "); |
|  | scanf("%d",&n); |
|  | for(i=0;i<n;i++) |
|  | { |
|  | printf("Enter burst time for process P%d : ",i+1); |
|  | scanf("%d",&bt[i]); |
|  | pdata[i] = i; |
|  | } |
|  | sort(); |
|  | printf("\nProcess\t\tWaiting time\tTurn around time\tPriority"); |
|  | calcwt(); |
|  | printf("\n\nOrder of process execution: "); |
|  | printq(); |
|  | printf("Average waiting time : %f\nAverage Turn around time : %f\n\n",avgwt/(float)n,avgtat/(float)n); |
|  | } |
|  | void sort() |
|  | { |
|  | int x,y,temp; |
|  | for(x=0;x<n;x++) |
|  | { |
|  | for(y=x;y<n;y++) |
|  | { |
|  | if(bt[x]>bt[y]) |
|  | { |
|  | priority[x]=x; |
|  | temp = bt[x]; |
|  | bt[x] = bt[y]; |
|  | bt[y] = temp; |
|  | temp = pdata[x]; |
|  | pdata[x] = pdata[y]; |
|  | pdata[y] = temp; |
|  | } |
|  | } |
|  | } |
|  | } |
|  |  |
|  | void calcwt() |
|  | { |
|  | int j=0,z; |
|  | for(z=0;z<n;z++) |
|  | { |
|  | waittime[z] = j; |
|  | j = bt[z] + j; |
|  | printf("\nP%d\t\t\t%d\t\t%d\t\t\t%d",pdata[z]+1,waittime[z],waittime[z]+bt[z],priority[z]); |
|  | avgwt+=waittime[z]; |
|  | avgtat+=waittime[z]+bt[z]; |
|  | } |
|  | } |
|  |  |
|  | void printq() |
|  | { |
|  | int p; |
|  | for(p=0;p<n;p++) |
|  | { |
|  | printf("P%d -> ",pdata[p]); |
|  | } |
|  | printf("End\n"); |
|  | } |