Name-Nitin Kandol

Section-K17EW

Roll No.-45

Reg no.-11704666

Subject-Operating system

Submitted to- Cherry Khosla

Q1. There are 3 student processes and 1 teacher process. Students are supposed to do their assignments and they need 3 things for that-pen, paper and question paper. The teacher has an infinite supply of all the three things. One student has pen, another has paper and another has question paper. The teacher places two things on a shared table and the student having the third complementary thing makes the assignment and tells the teacher on completion. The teacher then places another two things out of the three and again the student having the third thing makes the assignment and tells the teacher on completion. This cycle continues. WAP to synchronise the teacher and the students.

* Two types of people can enter into a library- students and teachers. After entering the library, the visitor searches for the required books and gets them. In order to get them issued, he goes to the single CPU which is there to process the issuing of books. Two types of queues are there at the counter-one for students and one for teachers. A student goes and stands at the tail of the queue for students and similarly the teacher goes and stands at the tail of the queue for teachers (FIFO). If a student is being serviced and a teacher arrives at the counter, he would be the next person to get service (PRIORITY-non preemptive). If two teachers arrive at the same time, they will stand in their queue to get service (FIFO). WAP to ensure that the system works in a non-chaotic manner.

* If a teacher is being served and during the period when he is being served, another teacher comes, then that teacher would get the service next. This process might continue leading to increase in waiting time of students. Ensure in your program that the waiting time of students is minimized.

#include<stdio.h>

#include<stdlib.h>

#include<unistd.h>

* 0 - Pen
* 1 - Paper
* 2 - Question Paper

int temp[2];

int n,i,j=1;

int A[3] = {0,1,2};

int main()

{

while(j>0)

{

for(i=0;i<2;i++)

{

temp[i] = A[rand()%3];

}

if(temp[0]==0 && temp[1]==1)

{

printf("\nTeacher Placed :-\n\n"); printf("1st Item = Pen\n"); printf("2nd Item = Paper\n");

printf("\nStudent having Question-Paper completed

the work.\n");

sleep(2);

}

else if(temp[0]==1 && temp[1]==2)

{

printf("\nTeacher Placed :-\n\n");

printf("1st Item = Paper\n");

printf("2nd Item = Question-Paper\n"); printf("\nStudent having Pen completed the

work.\n");

sleep(2);

}

else if(temp[0]==0 && temp[1]==2)

{

printf("\nTeacher Placed :-\n\n");

printf("1st Item = Pen\n");

printf("2nd Item = Question-Paper\n"); printf("\nStudent having Paper completed the

work.\n");

sleep(2);

}

else if(temp[0]==1 && temp[1]==0)

{

printf("\nTeacher Placed :-\n\n");

printf("1st Item = Paper\n");

printf("2nd Item = Pen\n");

printf("\nStudent having Question-Paper completed

the work.\n");

sleep(2);

}

else if(temp[0]==2 && temp[1]==1)

{

printf("\nTeacher Placed :-\n\n"); printf("1st Item = Question-Paper\n");

printf("2nd Item = Paper\n");

printf("\nUser having Pen completed the work.\n"); sleep(2);

}

else if(temp[0]==2 && temp[1]==0)

{

printf("\nTeacher Placed :-\n\n"); printf("1st Item = Question-Paper\n");

printf("2nd Item = Pen\n");

printf("\nUser having Paper completed the work.\n"); sleep(2);

}

else

{

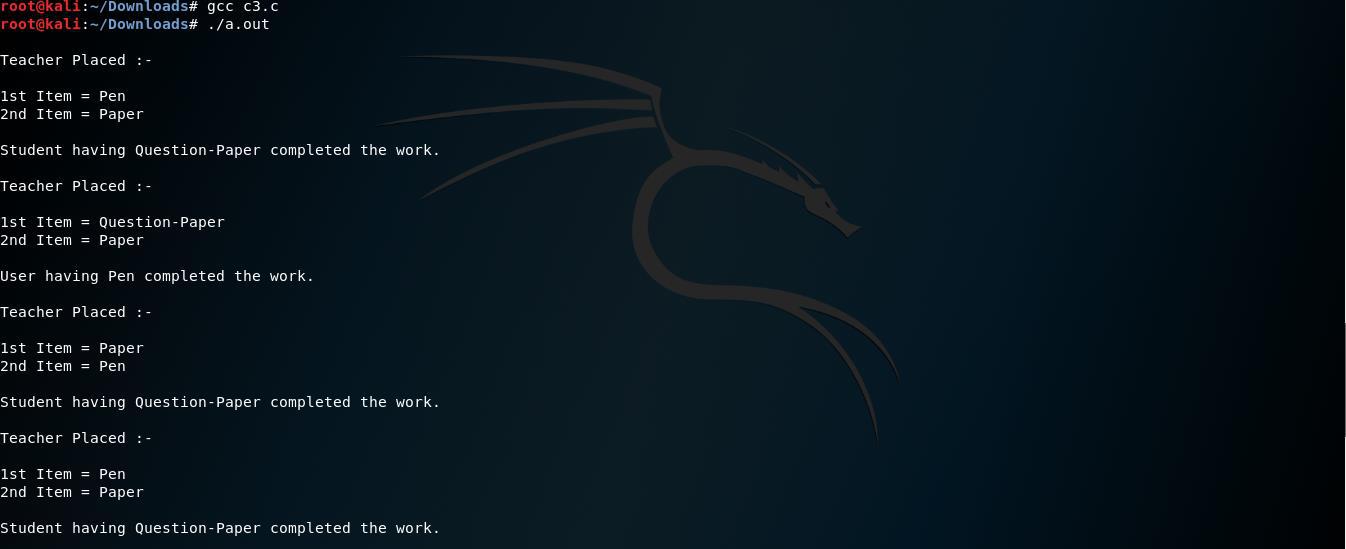
continue;

}

}

return 0;

}



#include<stdio.h>

void student()

{

int i,j,n,tot=0,pos,temp,avg\_wt,avg\_tat,total; printf("\nEnter Total Number of Student :");

scanf("%d",&n);

int at[n],wt[n],tat[n],pr[n],p[20];

printf("\nEnter Student arriving Time and Priority\n"); for(i=0;i<n;i++)

{

printf("\nstudent [%d]\n",i+1);

printf("Arriving Time:");

scanf("%d",&at[i]);

printf("Priority:");

scanf("%d",&pr[i]);

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

}

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

{

pos=i;

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

{

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

pos=j;

}

temp=pr[i];

pr[i]=pr[pos];

pr[pos]=temp;

temp=at[i];

at[i]=at[pos];

at[pos]=temp;

temp=p[i];

p[i]=p[pos];

p[pos]=temp;

}

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

//calculate waiting time

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

{

wt[i]=0;

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

wt[i]+=at[j];

total+=wt[i];

}

avg\_wt=total/n; total=0;

//average waiting time

printf("\nProcess\t

Arriving Time

\tWaiting Time\tTurnaround

Time");

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

{

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

total+=tat[i];

printf("\nStudent[%d]\t\t %d\t\t

%d\t\t\t%d",p[i],at[i],wt[i],tat[i]);

}

avg\_tat=total/n; //average turnaround time

//printf("\n\nAverage Waiting Time=%d",avg\_wt); printf("\nAverage Turnaround Time=%d\n",avg\_tat);

//return 0;

}

void teacher(){

int n,avwt=0,avtat=0,i,j;

printf("\nEnter total number of teacher: "); scanf("%d",&n);

int bt[n],wt[n],tat[n];

printf("\nEnter teacher arriving Time\n");

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

{

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

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

}

wt[0]=0; //waiting time for first process is 0

//calculating waiting time

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

{

wt[i]=0;

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

wt[i]+=bt[j];

}

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

//calculating turnaround time

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

{

tat[i]=bt[i]+wt[i];

avwt+=wt[i];

avtat+=tat[i];

printf("\nTeacher[%d]\t\t%d\t\t%d\t\t%d",i+1,bt[i],wt[i],tat[i]);

}

/\* avwt/=i;

avtat/=i;

printf("\n\nAverage Waiting Time:%d",avwt);\*/ printf("\nAverage Turnaround Time:%d",avtat);

// return 0;

}

int main(){

char choice;

printf("Select (T)eacher and (s)tudent or (b)oth : ");

scanf("%c",&choice);

switch(choice)

{

case 's':

case'S':

student();

break;

case 't':

case 'T':

teacher();

break;

case 'b':

case 'B':

student();

teacher();

break;

default:

printf("Wrong input");

}

//student();

return 0;

}



#include<stdio.h>

struct time

{

int at;

};

int sc,fc,isc=0, ifc= 0, min, flag,i,mc1=0,mc2=0,mc=0; void merger(struct time var1[],struct time var2[]) {

struct time f[fc],s[sc],m[sc+fc];

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

{

s[i]=var2[i];

}

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

{

f[i]=var1[i];

}

if( fc!=0 && sc!=0)

{

printf("\n");

while(isc<sc && ifc<fc)

{

if(f[ifc].at == s[isc].at)

{

m[mc1] = f[ifc];

printf("Faculty %d will issue the

book\n",mc1+1);

mc1++;

mc++;

ifc++;

m[mc2]= s[isc];

printf("Student %d will issue the

book\n",mc2+1);

mc2++;

isc++;

mc++;

}

else if(f[ifc].at < s[isc].at)

{

m[mc1]= f[ifc];

printf("Faculty %d will issue the

book\n",mc1+1);

mc1++;

ifc++;

mc++;

}

else if(f[ifc].at > s[isc].at)

{

m[mc2]= s[isc];

printf("Student %d will issue the

book\n",mc2+1);

mc2++;

mc++;

isc++;

}

else;

}

if(mc != (fc+sc))

{

if(fc!=ifc)

{

while(ifc!=fc)

{

m[mc1]= f[ifc];

printf("Faculty %d will issue the

book\n",mc1+1);

mc1++;

mc++;

ifc++;

}

}

else if(sc!=isc)

{

while(isc!=sc)

{

m[mc2]= s[isc];

printf("Student %d will issue the

book\n",mc2+1);

mc2++;

mc++;

isc++;

}

}

}

}

else if(fc==0)

{

while(isc!=sc)

{

printf("\n");

m[mc2]= s[isc];

printf("Student %d issued the book\n",mc2+1);

mc2++;

mc++;

isc++;

}

}

else if(sc==0)

{

while(ifc!=fc)

{

printf("\n");

m[mc1]= f[ifc];

printf("Faculty %d issued the book\n",mc1+1);

mc1++;

mc++;

ifc++;

}

}

else

{

printf("\n No one is in the Queue\n");

}

}

void main()

{

printf("Enter number of students : "); scanf("%d",&sc);

printf("Enter Number of Teacher : ");

scanf("%d",&fc);

struct time fac[fc],std[sc];

printf("\n");

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

{

printf("Enter arrival time of Student %d = ",i+1); scanf("%d",&std[i].at);

}

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

{

printf("Enter arrival time of Faculty %d = ",i+1); scanf("%d",&fac[i].at);

}

merger(fac,std);

}



THANK YOU