***Aim: -***Write a program to implements vector clock..

***Program:-***

#include<stdio.h>

struct time

{

int p1;

int p2;

int p3;

}

time[5][10];

struct msg

{

int p;

int e;

}

msg[5][10];

void main()

{

int n,n1,n2,n3,i,j,d=1,init=0,run=1,choice,ps,pr,s,r,rep=1,sev=1;

printf("\nEnter Number of Processes : ");

scanf("%d",&n);

printf("\nEnter Number of Events in P1 : ");

scanf("%d",&n1);

printf("\nEnter Number of Events in P2 : ");

scanf("%d",&n2);

printf("\nEnter Number of Events in P3 : ");

scanf("%d",&n3);

for(i=0;i<10;i++)

{

msg[0][i].p=-1;

msg[0][i].e=-1;

msg[1][i].p=-1;

msg[1][i].e=-1;

msg[2][i].p=-1;

msg[2][i].e=-1;

}

time[0][0].p1=init+d;

time[1][0].p2=init+d;

time[2][0].p3=init+d;

for(i=1;i<n1;i++)

{

time[0][i].p1=time[0][i-1].p1+d;

time[0][i].p2=0;

time[0][i].p3=0;

}

for(i=1;i<n2;i++)

{

time[1][i].p1=0;

time[1][i].p2=time[1][i-1].p2+d;

time[1][i].p3=0;

}

for(i=1;i<n3;i++)

{

time[2][i].p1=0;

time[2][i].p2=0;

time[2][i].p3=time[2][i-1].p3+d;

}

while(run)

{

printf("\n ASSIGN MSG TO PROCESSES : PRESS 1");

printf("\n TO SHOW TIME STAMP : PRESS 2");

printf("\n ENTER YOUR CHOICE : ");

scanf("%d",&choice);

switch(choice)

{

case 1:

printf("\nENTER SENDING PROCESS NO :");

scanf("%d",&ps);

printf("\nENTER SENDING EVENT NO :");

scanf("%d",&s);

printf("\nENTER RECIEVING PROCESS NO :");

scanf("%d",&pr);

printf("\nENTER RECIEVING EVENT NO :");

scanf("%d",&r);

msg[pr-1][r-1].p=ps-1;

msg[pr-1][r-1].e=s-1;

printf("\nMSG SENT FROM P%d-e%d To P%d-e%d:",ps,s,pr,r);

break;

case 2:

run=0;

break;

default :

printf("\n WRONG CHOICE PLEASE RE-ENTER \n");

}

}

RAM:

if(sev==2)

{

rep=2;

}

for(i=0;i<n1;i++)

{

if((msg[0][i].p!=-1)&&(msg[0][i].e!=-1))

{

if((time[msg[0][i].p][msg[0][i].e].p1)>time[0][i].p1)

{

time[0][i].p1=time[msg[0][i].p][msg[0][i].e].p1;

for(j=i+1;j<n1;j++)

{

time[0][j].p1=time[0][j].p1+time[0][i].p1;

}

}

if((time[msg[0][i].p][msg[0][i].e].p2)>time[0][i].p2)

{

time[0][i].p2=time[msg[0][i].p][msg[0][i].e].p2;

for(j=i+1;j<n1;j++)

{

time[0][j].p2=time[0][j].p2+time[0][i].p2;

}

}

if((time[msg[0][i].p][msg[0][i].e].p3)>time[0][i].p3)

{

time[0][i].p3=time[msg[0][i].p][msg[0][i].e].p3;

for(j=i+1;j<n1;j++)

{

time[0][j].p3=time[0][j].p3+time[0][i].p3;

}

}

}

}

for(i=0;i<n2;i++)

{

if((msg[1][i].p!=-1)&&(msg[1][i].e!=-1))

{

if((time[msg[1][i].p][msg[1][i].e].p2+d)>time[1][i].p2)

{

time[1][i].p2=time[msg[1][i].p][msg[1][i].e].p2+d;

for(j=i+1;j<n2;j++)

{

time[1][j].p2=time[1][j].p2+time[1][i].p2;

}

}

if((time[msg[1][i].p][msg[1][i].e].p1)>time[1][i].p1)

{

time[1][i].p1=time[msg[1][i].p][msg[1][i].e].p1;

for(j=i+1;j<n2;j++)

{

time[1][j].p1=time[1][j].p1+time[1][i].p1;

}

}

if((time[msg[1][i].p][msg[1][i].e].p3)>time[1][i].p3)

{

time[1][i].p3=time[msg[1][i].p][msg[1][i].e].p3;

for(j=i+1;j<n2;j++)

{

time[1][j].p3=time[1][j].p3+time[1][i].p3;

}

}

}

}

for(i=0;i<n3;i++)

{

if((msg[2][i].p!=-1)&&(msg[2][i].e!=-1))

{

if((time[msg[2][i].p][msg[2][i].e].p1)>time[2][i].p1)

{

time[2][i].p1=time[msg[2][i].p][msg[2][i].e].p1;

for(j=i+1;j<n3;j++)

{

time[2][j].p1=time[2][j].p1+time[2][i].p1;

}

}

if((time[msg[2][i].p][msg[2][i].e].p2)>time[2][i].p2)

{

time[2][i].p2=time[msg[2][i].p][msg[2][i].e].p2;

for(j=i+1;j<n3;j++)

{

time[2][j].p2=time[2][j].p2+time[2][i].p2;

}

}

if((time[msg[2][i].p][msg[2][i].e].p3)>time[2][i].p3)

{

time[2][i].p3=time[msg[2][i].p][msg[2][i].e].p3;

for(j=i+1;j<n3;j++)

{

time[2][j].p3=time[2][j].p3+time[2][i].p3;

}

}

}

if(rep==1)

{

sev=2;

goto RAM;

}

}

printf("PROCESS EVENT TIMESTAMP ");

for(i=0;i<n1;i++)

{

printf("\nP%d e%d [%d %d %d]",1,i+1,time[0][i].p1,time[0][i].p2,time[0][i].p3);

}

for(i=0;i<n2;i++)

{

printf("\nP%d e%d [%d %d %d]",2,i+1,time[1][i].p1,time[1][i].p2,time[1][i].p3);

}

for(i=0;i<n3;i++)

{

printf("\nP%d e%d [%d %d %d]",3,i+1,time[2][i].p1,time[2][i].p2,time[2][i].p3);

}

return;

}

***Output :-***





