// Roll No. :- B3003

// Title :- Lamport’s Algorithm

//Program For Clock Synchronisation.

#include<stdio.h>

#include<conio.h>

#include<graphics.h>

#define MAX 15

struct synch

{

int process,clock,send,receive,starttime,recvtime,flag;

int xco,yco,x1co,y1co;

};

void main()

{

int noOfProcess,i,j,ch,loop,tmp,tmp1,tmp2,tmp3;

int gdriver=DETECT,gmode;

int x=0,y=0,x1,y1;

char list[MAX]=" ";

struct synch s[MAX];

printf("\nEnter Number Of Process:");

scanf("%d",&noOfProcess);

printf("\nProcess ID\tTime Slice");

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

{

printf("\n%d\t->\t",i);

scanf("%d",&s[i].clock);

s[i].process=i;

}

ch=1;

i=0;

do

{

printf("\nEnter Sender Process:");

scanf("%d",&s[i].send);

while((s[i].send>noOfProcess-1))

{

printf("\nWrong i/p.\nEnter Sender Process:");

scanf("%d",&s[i].send);

}

printf("\nEnter Receive Process:");

scanf("%d",&s[i].receive);

while((s[i].receive>noOfProcess-1))

{

printf("\nWrong i/p.\nEnter Receive Process:");

scanf("%d",&s[i].receive);

}

printf("\nEnter Event Start time:");

scanf("%d",&s[i].starttime);

while((s[i].starttime % s[s[i].send].clock)!=0)

{

printf("\nWrong i/p.\nEnter Event Start time:");

scanf("%d",&s[i].starttime);

}

s[i].recvtime=(((s[i].starttime/s[s[i].send].clock)+1)\*(s[s[i].receive].clock));

if((s[i].receive)==(s[i].send))

{

s[i].recvtime=s[i].starttime;

}

printf("\nWant to conti..(1/0)");

scanf("%d",&ch);

loop=i;

i++;

}while(ch!=0);

printf("\nSender\tReceive\tStart time\tReceive Time");

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

{

printf("\n%d\t%d\t%d\t\t%d",s[i].send,s[i].receive,s[i].starttime,s[i].recvtime);

s[i].flag=0;

}

getch();

initgraph(&gdriver,&gmode,"c:\\tc\\bgi");

x=60;

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

{

tmp=s[i].clock;

tmp1=tmp;

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

{

itoa(tmp1,list,10);

tmp1=tmp1+tmp;

strcat(list,'\0');

outtextxy(x,y,list);

y+=10;

strcpy(list," ");

}

y=0;

x+=60;

}

//printf("%d",loop);

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

{

x=1;x1=1;y=1;y1=1;

tmp=s[i].starttime;

tmp1=s[i].recvtime;

tmp2=s[i].send;

tmp3=s[i].receive;

//printf("\n%d",tmp1);

x=tmp2\*60;

if(x==0)

x=60;

y=((tmp/s[s[i].send].clock)\*10)-10;

x1=tmp3\*60;

if(x1==0)

x1=60;

y1=((tmp1/s[s[i].receive].clock)\*10)-10;

if(tmp2<tmp3)

{

if(tmp2==0)

{

line(x,y,x1+60,y1);

s[i].xco=x;

s[i].yco=y;

s[i].x1co=x1+60;

s[i].y1co=y1;

}

else

{

line(x+60,y,x1+60,y1);

s[i].xco=x+60;

s[i].yco=y;

s[i].x1co=x1+60;

s[i].y1co=y1;

}

circle(x1+60,y1,2);

}

if(tmp2>tmp3)

{

if(tmp3==0 && tmp2==1)

{

line(x1+60,y,x,y1);

s[i].xco=x;

s[i].yco=y;

s[i].x1co=x1+60;

s[i].y1co=y1;

circle(x,y1,2);

}

else

{

if(tmp3==0 && tmp2!=1)

{

line(x1+120,y,x-60,y1);

s[i].xco=x-60;

s[i].yco=y;

s[i].x1co=x1+120;

s[i].y1co=y1;

circle(x-60,y1,2);

}

else

{

line(x+60,y,x1+60,y1);

s[i].xco=x+60;

s[i].yco=y;

s[i].x1co=x1+60;

s[i].y1co=y1;

circle(x,y1,2);

}

}

}

//printf("\n%d%d%d%d",x,y,x1,y1);

getch();

}

closegraph();

printf("\n\nFinding Problematic Conditions:");

printf("\nSender\tReceive\tStart time\tReceive Time");

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

{

//printf("\n%d\t%d\t%d\t\t%d",s[i].send,s[i].receive,s[i].starttime,s[i].recvtime);

if(s[i].recvtime<s[i].starttime)

{

s[i].flag=1;

printf("\n%d\t%d\t%d\t\t%d",s[i].send,s[i].receive,s[i].starttime,s[i].recvtime);

//printf("\tChange");

}

}

getch();

printf("\n\nAfter Applying Lamport Algorithm.");

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

{

if(s[i].flag==1)

{

while(s[i].recvtime<s[i].starttime)

{

s[i].recvtime=s[i].recvtime+s[s[i].receive].clock;

}

printf("\n%d\t%d\t%d\t\t%d",s[i].send,s[i].receive,s[i].starttime,s[i].recvtime);

printf("\*");

}

else

{

printf("\n%d\t%d\t%d\t\t%d",s[i].send,s[i].receive,s[i].starttime,s[i].recvtime);

}

}

getch();

initgraph(&gdriver,&gmode,"c:\\tc\\bgi");

x=60;

y=0;

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

{

/\*if(s[i].flag!=1)

{

tmp=s[i].starttime+1;//s[s[i].send].clock;

tmp1=tmp;

}

else\*/

{

tmp=s[i].clock;

tmp1=tmp;

}

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

{

itoa(tmp1,list,10);

tmp1=tmp1+tmp;

strcat(list,'\0');

outtextxy(x,y,list);

y+=10;

strcpy(list," ");

}

y=0;

x+=60;

}

getch();

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

{

x=1;x1=1;y=1;y1=1;

tmp=s[i].starttime;

tmp1=s[i].recvtime;

tmp2=s[i].send;

tmp3=s[i].receive;

x=tmp2\*60;

if(x==0)

x=60;

y=((tmp/s[s[i].send].clock)\*10)-10;

x1=tmp3\*60;

if(x1==0)

x1=60;

y1=((tmp1/s[s[i].receive].clock)\*10)-10;

if(tmp2<tmp3)

{

if(tmp2==0)

{

line(x,y,x1+60,y1);

}

else

{

line(x+60,y,x1+60,y1);

}

circle(x1+60,y1,2);

}

if(tmp2>tmp3)

{

if(tmp3==0 && tmp2==1)

{

line(x1+60,y,x,y1);

circle(x,y1,2);

}

else

{

if(tmp3==0 && tmp2!=1)

{

line(x1+120,y,x-60,y1);

circle(x-60,y1,2);

}

else

{

line(x+60,y,x1+60,y1);

circle(x,y1,2);

}

}

}

getch();

}

closegraph();

getch();

}

// Output :-

Enter Number Of Process:3

Process ID Time Slice

0 -> 6

1 -> 8

2 -> 10

Enter Sender Process:0

Enter Receive Process:1

Enter Event Start time:6

Want to conti..(1/0)1

Enter Sender Process:1

Enter Receive Process:2

Enter Event Start time:24

Want to conti..(1/0)1

Enter Sender Process:2

Enter Receive Process:1

Enter Event Start time:50

Want to conti..(1/0)1

Enter Sender Process:1

Enter Receive Process:0

Enter Event Start time:48

Want to conti..(1/0)0

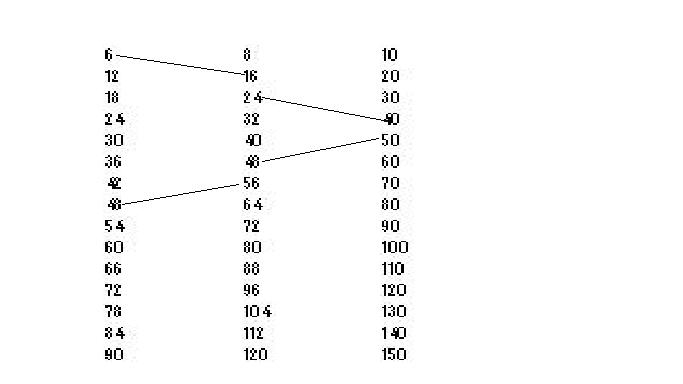
Sender Receive Start time Receive Time

0 1 6 16

1 2 24 40

2 1 50 48

1 0 48 42



Finding Problematic Conditions:

Sender Receive Start time Receive Time

2 1 50 48

1 0 48 42

After Applying Lamport Algorithm.

0 1 6 16

1 2 24 40

2 1 50 56\*

1 0 48 48\*

