SHIVANI MALIK
2003730100017

PROGRAM- 1

Write a program in C to implement Lamports logical clock

```
#include<stdio.h>
 #include <conio.h>
struct process
{
int e;
int ts[10];
}
p[10];
 void main()
 {
 int i,j,n,m,t,e1,e2;
char ch;
clrscr();
 printf("enter the no. of process ");
scanf("%d",&n); for(i=0;i<n;i++)
{
printf("enter the no. of events in process %d",i+1);
scanf("%d",&p[i].e); for(j=0;j<p[i].e;j++) { p[i].ts[j]=j+1;
}
}
for(i=0;i<n;i++)
{
for(j=0;j<p[i].e;j++)
 printf("%d ",p[i].ts[j]); printf("\n");
}
do
{
 printf("enter the process no & event no. from which message is passing (less
than %d)",n);
scanf("%d %d",&m,&e1);
printf("enter the process no & event no. on which msg is passing (less than %d)",n);
scanf("%d %d",&t,&e2);
if((p[m].ts[e1]+1)>p[t].ts[e2])
 {
 p[t].ts[e2]=p[m].ts[e1]+1;
 for(i=e2;i<p[t].e;i++)
{
 p[t].ts[i+1]=p[t].ts[i]+1;
}
}
printf("is there more message(y/n)");
fflush(0);
scanf("%c",&ch);
}
while(ch=='y'&& ch=='Y');
for(i=0;i<n;i++)
{
for(j=0;j<p[i].e;j++)
```

```
printf("%d ",p[i].ts[j]);
printf("\n");
}
getch();
}
```

OUTPUT

```
enter the no. of process 4
enter the no. of events in process 1 6
enter the no. of events in process 1 7
enter the no. of events in process 1 8
enter the no. of events in process 1 9
1 2 3 4 5 6
1 2 3 4 5 6 7
1 2 3 4 5 6 7 8
1 2 3 4 5 6 7 8 9
enter the process no & event no. On which msg is passing(less than 4) 3
is there more message(y/n) n
```

SHIVANI MALIK
2003730100017

EXPERIMENT 2
Write a program for Program for implementing Vector Clock.

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
long *p1(int i,long *comp);
long *p2(int i,long *comp);
long *p3(int i,long *comp);
void main() {
long start[]={0,0,0},*vector;
while(!kbhit()) {
p1(1,&start[0]);
}
printf("\n Process Vector\n");
vector=p1(0,&start[0]);
printf("p1[%ld%ld%ld]\n",*vector,*(vector+1),*(vector+2)); vector=p2(0,&start[0]);
printf("p2[%ld%ld%ld]\n",*vector,*(vector+1),*(vector+2)); vector=p3(0,&start[0]);
printf("p3[%ld%ld%ld]\n",*vector,*(vector+1),*(vector+2));
}
long *p1(int i,long *comp) {
static long a[]={0,0,0};
int next;
if(i==1) {
a[0]++;
if(*(comp+1)>a[1])
a[1]=*(comp+1);
if(*(comp+2)>a[2])
a[2]=*(comp+2);
next=random(2);
if(next==0)
p2(1,&a[0]);
else if(next==1)
p3(1,&a[0]);
return(&a[0]);
}
else
return(&a[0]);
}
long *p2(int i,long *comp)
{
static long b[]={0,0,0};
int next;
if(i==1)
{
b[i]++;
if(*comp>b[0])
b[0]=*(comp);
if(*(comp+2)>b[2])
b[2]=*(comp+2);
```

```
next=random(2);
if(next==0)
p1(1,&b[0]);
else if(next==1)
p3(1,&b[0]);
return &b[0];
}
else
return &b[0];
}
long *p3(int i,long *comp) {
static long c[]={0,0,0}; int next;
if(i==1)
{
c[2]++;
if(*comp>c[0])
c[0]=*(comp);
if(*(comp+1)>c[1])
c[1]=*(comp+1);
next=random(2);
if(next==0)
p1(1,&c[0]);
return &c[0];
}
else
return &c[0];
}
```

OUTPUT

```
      Process Vector
p1[1037269518484778307]
p2[1037269518484778306]
p3[1037269518484778300]
```

SHIVANI MALIK
2003730100017

PROGRAM- 3
Write a program to implement edge chasing distributed deadlock detection algorithm

```
#include<stdio.h>
#include<conio.h>
 void main()
 {
 int temp,process[10][15],
site_count=0,
process_count=0,i,j,k,waiting[15];
int p1,p2,p3;
clrscr();
printf("\n Enter the no. of sites (max 3) \n");
scanf("%d",&site_count);
 for(i=1;i<=site_count;i++)
{
 printf("\n Enter the no. of processes in %d site ( max 4)\n",i);
scanf("%d",&process_count);
for(j=0;j<process_count;j++)
{
 process[i][j]=i+(i*j);
 }
 }
 printf("\n Enter the blocked process \n");
scanf("%d",&k); for(i=1;i<=3;i++)
{
 for(j=0;j<=3;j++)
 {
 if(k==process[i][j])
{
printf("Process %d is at site %d ",k,i);
temp=i;
}
if(k==process[i][j])
 printf("It is a deadlock \n");
if(k==(process[temp][j])&&((process[temp][j])==waiting[process[i][j]])&&(temp!=i))
{
//probe(temp,j,process[i][j]);
if(process[i][j]==waiting[process[temp][j]]);
printf("It is a deadlock\n");
}
}
getch();
}
}
```

SHIVANI MALIK
2003730100017

OUTPUT

```
Enter the no. of sites (max 3)
3
Enter the no. of processes in 1 site (max 4)
2
Enter the no. of processes in 2 site (max 4)
3
Enter the no. of processes in 3 site (max 4)
1
Enter the blocked process
2
Process 2 is at site 1 it is a deadlock
Process 2 is at site 2 it is a deadlock
```

SHIVANI MALIK
2003730100017

PROGRAM- 4

Write a program in C to implement locking algorithm

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a=0;
char b,c;
clrscr();
do
{
printf("if transaction T1 want to lock data object ");
fflush(0);
scanf("%c",&b);
if(a==0 && b=='y')
{a=1; b='n';
}
else
if(a==1)
printf("data object is locked");
printf("if transaction T2 want to lock data object ");
fflush(0);
scanf("%c",&b);
if(a==0 && b=='y')
{
a=1; b='n';
}
else
printf("data object is locked");
printf("\nif transaction want to release data object ");
fflush(0);
scanf("%c",&b);
if(a==1 && b=='y') a=0;
printf("do you want to continue ");
fflush(0);
scanf("%c",&c);
}
while(c=='y');
getch();
}
```

OUTPUT

if transaction T1 want to lock data object x
if transaction T2 want to lock data object y

if transaction want to release data object x
do you want to continue y
if transaction t1 want to lock data object x
data object is locked if transaction t2 want to lock data object y
data object is locked
if transaction want to release data object x
do you want to continue n

SHIVANI MALIK
2003730100017

PROGRAM- 5
Write a program in C to implement non token based algorithm for Distributed multual exclusion

```
#include<stdio.h>
#include<conio.h>
void main()
{
int i,d,p,a,c=0,aa[10],j,n;
char ch='y';
clrscr();
printf("enter no of processes ");
scanf("%d",&n);
i=0;
do
{
printf("enter the process no which want to execute critical section "); scanf("%d",&a);
aa[i]=a;
i++;
c=c+1;
d=i;
printf("some other process want to execute cs? then press (y) "); fflush(0);
scanf("%c",&ch);
}
while(ch=='y');
for(j=1;j<=c;j++)
{
printf("\ncritical section is executing for process %d in queue.....",j);
printf("\ncritical section is finished for process %d",j); printf("\nrelease msg has sent
by process%d",j);
}
getch();
}
```

OUTPUT

```
enter no of processes 3
enter the process no which you want to execute critical section 2
some other process want to execute cs? then press (y) y
enter the process no which you want to execute critical section 1
some other process want to execute cs? then press (y) y
enter the process no which you want to execute critical section 3
some other process want to execute cs? then press (y) y

critical section is executing for process 1 in queue……
critical section is finished for process 1
critical section is executing for process 2 in queue……
critical section is finished for process 2
critical section is executing for process 3 in queue……
critical section is finished for process 3
```

SHIVANI MALIK
2003730100017

PROGRAM- 6

Write a program in C to implement termination detection

```
#include <stdio.h>
#include <stdlib.h>
#include <dos.h>
void main()
{
int i, j, k = 0, n, tw, total = 0, we, ca, w[20];
clrscr();
printf("Enter the number of processes: ");
scanf("%d", &n);

printf("\nAssign a controlling agent: ");
scanf("%d", &ca);

printf("\nEnter the total weight: ");
scanf("%d", &tw);

while (k < n)
{
w[k] = rand() % tw;
tw = tw - w[k];
k++;
}
for (k = 0; k < n; k++)
{
total = total + w[k];
}
printf("%d\n", total);
w[n - 1] = abs(tw - total);

printf("%d\n", w[n - 1]);
printf("\nControlling agent %d %d\n\n", ca, w[ca]);

printf("\nSending computational message to...\n\n");
for (j = 0; j < n; j++)
{
 if (j != (ca - 1))
{
 sound(700);
 delay(2000);
 printf("\tProcess %d %d\n", j + 1, w[j]);
}
}
nosound();
getch();
}
```

OUTPUT

Enter the number of processes: 3

Assign a controlling agent: P

Enter the total weight: 11458
10113

Controlling agent 5268 0

Sending computational message to...

Process 1 346
Process 2 130
Process 3 10113

PROGRAM- 7
Write a c program on Nested Transaction

```
#include <stdio.h>
// Structure to represent a bank account
typedef struct {
int account_number;
float balance;
} Account;
// Function to simulate a simple transaction - Withdraw
void withdraw(Account *account, float amount) {
// Simulating Consistency - Ensure sufficient balance before withdrawal
if (account->balance >= amount) {
// Simulating Atomicity - All or nothing
account->balance -= amount;
printf("Withdrawal successful. New balance: %.2f\n", account->balance);
} else {
printf("Insufficient funds for withdrawal.\n");
}
}
// Function to simulate a simple transaction - Deposit
void deposit(Account *account, float amount) {
// Simulating Atomicity - All or nothing
account->balance += amount;
printf("Deposit successful. New balance: %.2f\n", account->balance);
}

// Function to simulate a nested transaction
void nestedTransaction(Account *account) {
// Transaction 1: Withdraw
withdraw(account, 200.00);

// Transaction 2: Nested transaction - Deposit within the withdrawal transaction
#pragma omp atomic
deposit(account, 100.00);
}

int main() {
// Creating a sample bank account
Account myAccount = {1234, 1000.00};
// Performing a nested transaction
nestedTransaction(&myAccount);

// Simulating Durability - Changes are persistent after the transactions
printf("Final balance after nested transaction: %.2f\n", myAccount.balance);

return 0;
}
```

OUTPUT

Withdrawal successful. New balance: 800.00
Deposit successful. New balance: 900.00
Final balance after nested transaction: 900.00