

Date: 08<sup>th</sup> Oct 2021

Name: Rohan Khurana  
Roll No: 1802910129

## Distributed System Lab LAB - 3

**Aim:** WAP to implement Vector Clock

**Program:**

```
#include<iostream>
#include<conio.h>
#define SIZE 10
using namespace std;
class node {
public:
int data[SIZE];
node *next;
node() {
for(int p=0; p<SIZE; p++) {
data[p] = 0;
}
next = NULL;
}
node(int v[], int n1) {
for(int s = 0; s < n1; s++) {
data[s] = v[s];
}
next = NULL;
}
friend class process;
}*start=NULL;
int main() {
int n, events, sent, receive, sentE, recE, commLines = 0;
node *temp;
node *proc[SIZE];
cout<<"Enter no. of processes: ";
cin>>n;
int vector[n] = {0};
for(int i = 0; i < n; i++) {
for(int v = 0; v < n; v++) {
```

```

        vector[v] = 0;
    }
    cout<<"Enter no. of events in process "<<i+1<<": ";
    cin>>events;
    for(int j = 1; j <= events; j++) {
        vector[i] = j;
        node *newnode = new node(vector,n);
        if(start == NULL) {
            start = newnode;
            temp = start;
        }
        else {
            temp->next = newnode;
            temp = temp->next;
        }
    }
    proc[i] = start;
    start = NULL;
}
cout<<"\nEnter the number of communication lines: ";
cin>>commLines;
node *tempS, *tempR;
for(int i = 0; i < commLines; i++) {
    cout<<"\nEnter the sending process: ";
    cin>>sent;
    cout<<"\nEnter the receiving process: ";
    cin>>receive;
    cout<<"\nEnter the sending event number: ";
    cin>>sentE;
    cout<<"\nEnter the receiving event number: ";
    cin>>recE;
    tempS = proc[sent - 1];
    tempR = proc[receive - 1];
    for(int j = 1; j < sentE; j++)
        tempS = tempS->next;
    for(int j = 1; j < recE; j++)
        tempR = tempR->next;
    for(int j = 0; j < n; j++) {
        tempR->data[j] = (tempR->data[j] < tempS->data[j]) ?
tempS->data[j] : tempR->data[j];
    }
}

```

```

    }
}
cout<<"\nThe resulting vectors are:\n\n";
for(int k = 0; k < n; k++) {
    cout<<"Process "<<k + 1<<": ";
    node *temp1 = proc[k];
    while(temp1) {
        cout<<"(";
        for(int f = 0; f < n - 1; f++)
            cout<<temp1->data[f]<<",";
        cout<<temp1->data[n-1];
        cout<<")";
        temp1 = temp1->next;
    }
    cout<<endl;
}
return 0;
}

```

### Output:

```

Enter no. of processes: 3
Enter no. of events in process 1: 3
Enter no. of events in process 2: 2
Enter no. of events in process 3: 3

Enter the number of communication lines: 2

Enter the sending process: 1
Enter the receiving process: 3
Enter the sending event number: 2
Enter the receiving event number: 3

Enter the sending process: 3
Enter the receiving process: 2
Enter the sending event number: 1
Enter the receiving event number: 2

The resulting vectors are:

Process 1: (1,0,0)(2,0,0)(3,0,0)
Process 2: (0,1,0)(0,2,1)
Process 3: (0,0,1)(0,0,2)(2,0,3)

```