Ex.No: 2	IMPLEMENTATION OF ELECTION ALGORITHMS
Date: 08.01.2024	

1.Bully algorithm:

Code:

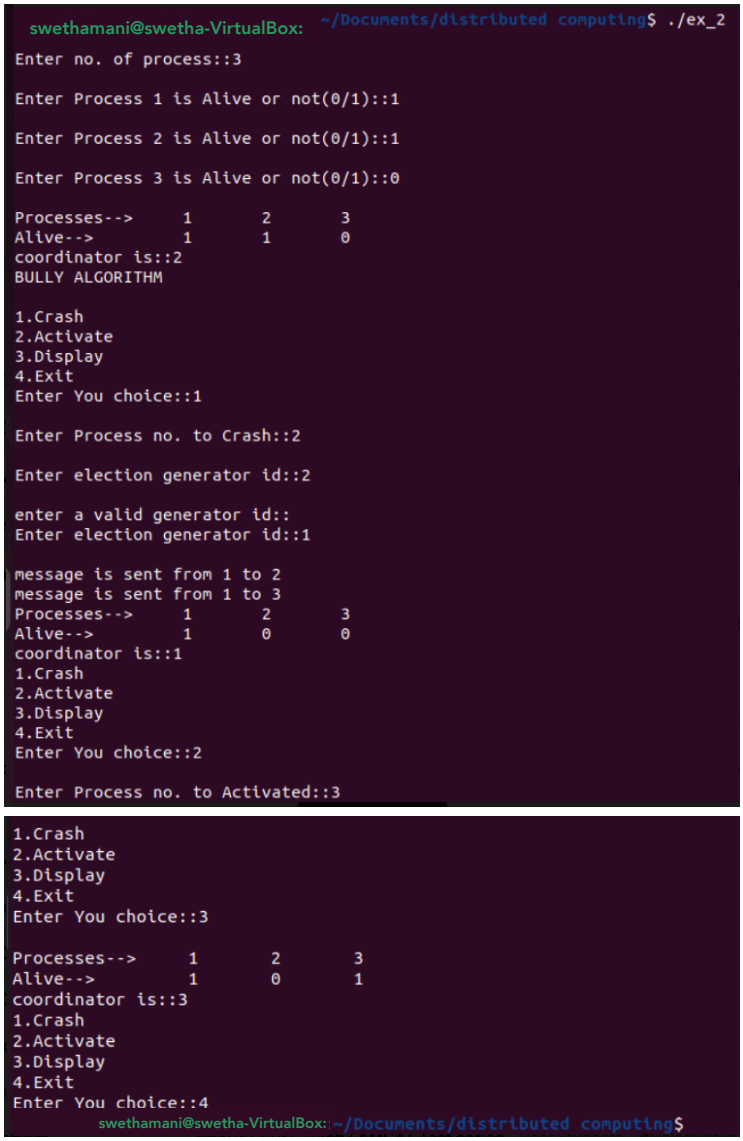
```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
#define MAX 10
int list[MAX],n,c;
void display()
{
int i;
printf("\nProcesses-->");
for(i=1;i<=n;i++)
printf("\t %d",i);
printf("\nAlive-->");
for(i=1;i<=n;i++)
printf("\t %d",list[i]);
printf("\ncoordinator is::%d",c);
}
void bully()
{
int ch,crash,activate,i,gid,flag,subcdr;
do
{printf("\n1.Crash\n2.Activate\n3.Display\n4.Exit\nEnter You choice::");
scanf("%d",&ch);
switch(ch)
```

```
{case 1:
printf("\nEnter Process no. to Crash::");
scanf("%d",&crash);
if(list[crash])
list[crash]=0;
else
{printf("\nProcess is alreaady dead!!");
break;}
do
{printf("\nEnter election generator id::");
scanf("%d",&gid);
if(gid==c)
{printf("\nenter a valid generator id::");
}
}while(gid==crash);
flag=0;
if(crash==c)
{for(i=gid+1;i<=n;i++)
{printf("\nmessage is sent from %d to %d",gid,i);
if(list[i])
{subcdr=i;
printf("Response is sent from %d to %d",i,gid);
flag=1;
}}
if(flag==1)
{c=subcdr;
}
else
{c=gid;
}}
```

```
display();
break;
case 2:
//activate
printf("\nEnter Process no. to Activated::");
scanf("%d",&activate);
if(!list[activate])
list[activate]=1;
else
{printf("\nProcess is alreaady alive!!");
break;
}
if(activate==n)
{
c=n;
break;
}
for(i=activate+1;i<=n;i++)
{printf("\nmessage is sent from %d to %d",activate,i);
if(list[i])
{subcdr=i;
printf("Response is sent from %d to %d",i,activate);
flag=1;
}}
if(flag==1)
{c=subcdr;
}
else
{c=activate;
}
```

```
display();
break;
case 3:
display();
break;
case 4:
break;
}
}while(ch!=4);
}
int main()
{
int i,j;
printf("\nEnter no. of process::");
scanf("%d",&n);
for(i=1;i<=n;i++)
{
printf("\nEnter Process %d is Alive or not(0/1)::",i);
scanf("%d",&list[i]);
if(list[i])
c=i;
}
display();
printf("\nBULLY ALGORITHM\n");
bully();
return 0;
}
```

Output:


swethamani@swetha-VirtualBox: ~/Documents/distributed computing$ ./ex_2
Enter no. of process::3
Enter Process 1 is Alive or not(0/1)::1
Enter Process 2 is Alive or not(0/1)::1
Enter Process 3 is Alive or not(0/1)::0
Processes--> 1 2 3
Alive--> 1 1 0
coordinator is::2
BULLY ALGORITHM
1.Crash
2.Activate
3.Display
4.Exit
Enter You choice::1
Enter Process no. to Crash::2
Enter election generator id::2
enter a valid generator id::
Enter election generator id::1
message is sent from 1 to 2
message is sent from 1 to 3
Processes--> 1 2 3
Alive--> 1 0 0
coordinator is::1
1.Crash
2.Activate
3.Display
4.Exit
Enter You choice::2
Enter Process no. to Activated::3
1.Crash
2.Activate
3.Display
4.Exit
Enter You choice::3
Processes--> 1 2 3
Alive--> 1 0 1
coordinator is::3
1.Crash
2.Activate
3.Display
4.Exit
Enter You choice::4
swethamani@swetha-VirtualBox:~/Documents/distributed computing$

2.Ring algorithm:

Code:

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
#define MAX 10
int list[MAX],n,c;
void display()
{
int i;
printf("\nProcesses-->");
for(i=1;i<=n;i++)
printf("\t %d",i);
printf("\nAlive-->");
for(i=1;i<=n;i++)
printf("\t %d",list[i]);
printf("\ncoordinator is::%d",c);
}
void ring()
{
int msg[20],ring_n,k,i;
int ch,crash,activate,gid,flag,subcdr;
do
{
printf("\n1.Crash\n2.Activate\n3.Display\n4.Exit\nEnter You choice::");
scanf("%d",&ch);
switch(ch)
{case 1:
printf("\nEnter Process no. to Crash::");
scanf("%d",&crash);
```

```
if(list[crash])
list[crash]=0;
else
{printf("\nProcess is alreaady dead!!");
break;
}
do
{
printf("\nEnter election generator id::");
scanf("%d",&gid);
if(gid==c)
{printf("\nenter a valid generator id::");
}
}while(gid==crash);
flag=0;
k=1;
if(crash==c)
{msg[k++]=gid;
for(i=(gid+1)%n;i!=gid;i=(i+1)%n)
{
if(list[i])
{
printf("\nmessage is sent to %d k =%d",i,k);
msg[k++]=i;
// printf("Response is sent from %d to %d",i,gid);
}}
subcdr=0;
for(i=1;i<k;i++)
{printf("\nmsg::%d\n",msg[i]);
if(subcdr<msg[i])
```

```
{subcdr=msg[i];
}}
c=subcdr;
}
display();
break;
case 2:
//activate
printf("\nEnter Process no. to Activated::");
scanf("%d",&activate);
if(!list[activate])
list[activate]=1;
else
{printf("\nProcess is alreaady alive!!");
break;
}
if(activate==n)
{c=n;
break;
}
for(i=activate+1;i<=n;i++)
{printf("\nmessage is sent from %d to %d",activate,i);
if(list[i])
{subcdr=i;
printf("\nResponse is sent from %d to %d",i,activate);
flag=1;
}}
if(flag==1)
{c=subcdr;
}
```

```
else
{c=activate;
}
display();
break;
case 3:
display();
break;
case 4:
break;
}
}while(ch!=4);
}
int main()
{
int i,j;
printf("\nEnter no. of process::");
scanf("%d",&n);
for(i=1;i<=n;i++)
{printf("\nEnter Process %d is Alive or not(0/1)::",i);
scanf("%d",&list[i]);
if(list[i])
c=i;
}
display();
printf("\nRING ALGORITHM\n");
ring();
return 0;
}
```

Output:

```
swethamani@swetha-VirtualBox: /Documents/distributed computing$ gcc -o ex_2_1 ex_2_1.c
swethamani@swetha-VirtualBox: /Documents/distributed computing$ ./ex_2_1

Enter no. of process::3

Enter Process 1 is Alive or not(0/1)::1

Enter Process 2 is Alive or not(0/1)::1

Enter Process 3 is Alive or not(0/1)::1

Processes-->     1       2       3
Alive-->         1       1       1
coordinator is::3
RING ALGORITHM

1.Crash
2.Activate
3.Display
4.Exit
Enter You choice::1

Enter Process no. to Crash::1

Enter election generator id::3

enter a valid generator id::
Processes-->     1       2       3
Alive-->         0       1       1
coordinator is::3
1.Crash
2.Activate
3.Display
4.Exit
Enter You choice::2

Enter Process no. to Activated::1
```

```
message is sent from 1 to 2
Response is sent from 2 to 1
message is sent from 1 to 3
Response is sent from 3 to 1
Processes-->     1       2       3
Alive-->         1       1       1
coordinator is::3
1.Crash
2.Activate
3.Display
4.Exit
Enter You choice::3

Processes-->     1       2       3
Alive-->         1       1       1
coordinator is::3
1.Crash
2.Activate
3.Display
4.Exit
Enter You choice::4
```