# DAY -2

# CSA0465 – OPERATING SYSTEMS FOR HANDLING DEADLOCKS

# LAB EXPERIMENTS – Slot B

## Name :- Aswini .P

## Reg no :- 192011399

**6. Producer and Consumer.**

**Program :-**

```
#include <stdio.h>
#include <stdlib.h>
int mutex = 1;
int full = 0;
int empty = 10, x = 0;
void producer()
{
--mutex;
++full;
--empty;
x++;
printf("\nProducerproduces""item %d",x);
++mutex;
}
void consumer()
{
--mutex;
--full;
++empty;
printf("\nConsumer consumes ""item %d",x);
x--;
```

```
++mutex;
}
int main()
{
int n, i;
printf("\n1. Press 1 for Producer""\n2. Press 2 for Consumer""\n3. Press 3 for Exit");
#pragma omp critical
for (i = 1; i> 0; i++)
{
printf("\nEnter your choice:");
scanf("%d", &n);
switch (n)
{
case 1:
if ((mutex == 1)
&& (empty != 0))
{
producer();
}
else {
printf("Buffer is full!");
}
break;
case 2:
if ((mutex == 1)&& (full != 0))
{
consumer();
}
```

else {

printf("Buffer is empty!");

}

break;

case 3:

exit(0);

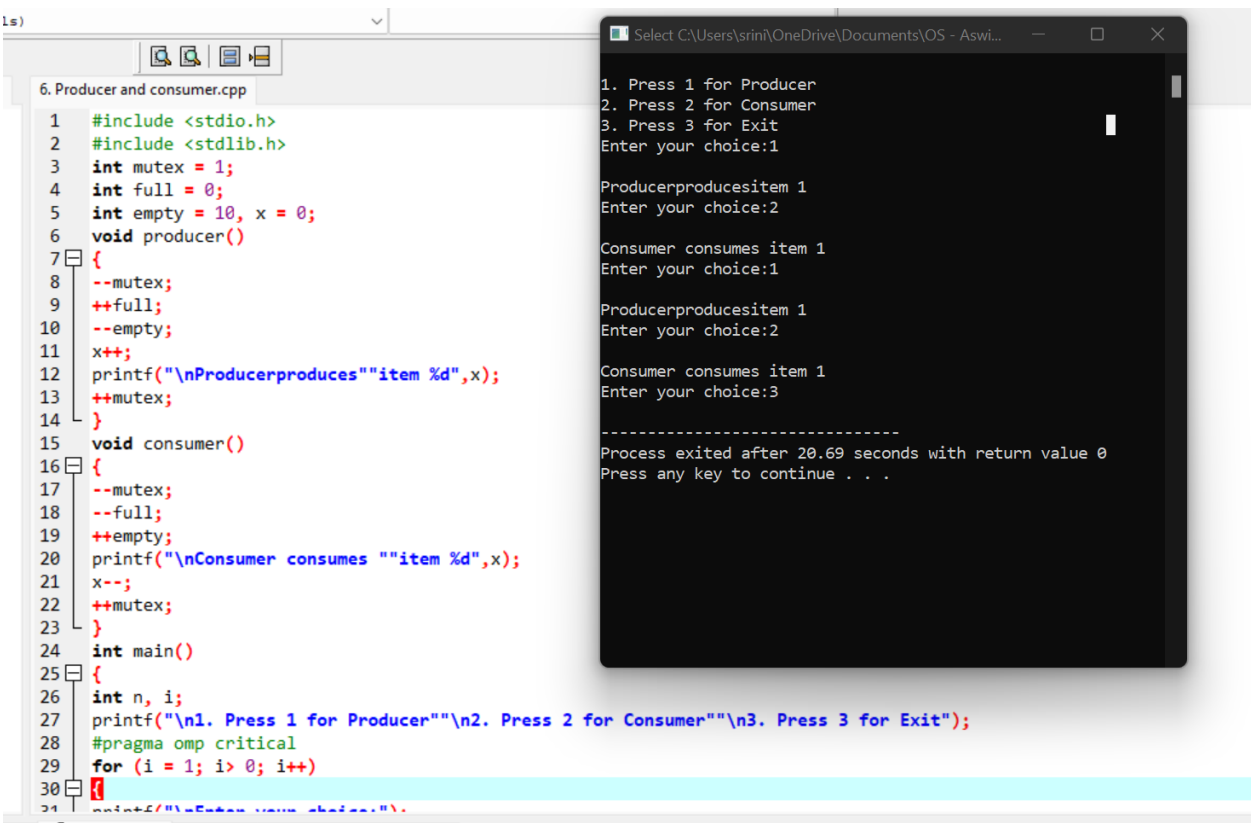
break;

}

}

}

**Output :-**




## 7. Paging – FIFO :-

**Program :-**

#include<stdio.h>

```
int main()
{
int i,j,n,a[50],frame[10],no,k,avail,count=0;
printf("\n ENTER THE NUMBER OF PAGES:\n");
scanf("%d",&n);
printf("\n ENTER THE PAGE NUMBER :\n");
for(i=1;i<=n;i++)
scanf("%d",&a[i]);
printf("\n ENTER THE NUMBER OF FRAMES :");
scanf("%d",&no);
for(i=0;i<no;i++)
frame[i]= -1;
j=0;
printf("\tref string\t page frames\n");
for(i=1;i<=n;i++)
{
printf("%d\t\t",a[i]);
avail=0;
for(k=0;k<no;k++)
if(frame[k]==a[i])
avail=1;
if (avail==0)
{
frame[j]=a[i];
j=(j+1)%no;
count++;
for(k=0;k<no;k++)
printf("%d\t",frame[k]);
```

}

printf("\n");

}

printf("Page Fault Is %d",count);

return 0;

}

**Output :-**

7. Paging FIFO.c

```c
#include<stdio.h>
int main()
{
int i,j,n,a[50],frame[10],no,k,avail,count=0;
printf("\n ENTER THE NUMBER OF PAGES:\n");
scanf("%d",&n);
printf("\n ENTER THE PAGE NUMBER :\n");
for(i=1;i<=n;i++)
scanf("%d",&a[i]);
printf("\n ENTER THE NUMBER OF FRAMES :");
scanf("%d",&no);
for(i=0;i<no;i++)
frame[i]= -1;
j=0;
printf("\tref string\t page frames\n");
for(i=1;i<=n;i++)
{
printf("%d\t\t\t",a[i]);
avail=0;
for(k=0;k<no;k++)
if(frame[k]==a[i])
avail=1;
if (avail==0)
{
frame[j]=a[i];
j=(j+1)%no;
count++;
for(k=0;k<no;k++)
printf("%d\t",frame[k]);
}
printf("\n");
}
printf("Page Fault Is %d",count);
return 0;
}
```

"C:\Users\pavan\OneDrive\Documents\OS\7. Pa...

```
 ENTER THE NUMBER OF PAGES:
9

 ENTER THE PAGE NUMBER :
1
2
3
5
6
3
2
4
2

 ENTER THE NUMBER OF FRAMES :3
        ref string       page frames
1               1       -1      -1
2               1       2       -1
3               1       2       3
5               5       2       3
6               5       6       3
3
2               5       6       2
4               4       6       2
2
Page Fault Is 7
Process returned 0 (0x0)   execution time : 19.512 s
Press any key to continue.
```

**8. Paging – LRU :-**

**Program :-**

#include<stdio.h>

int main()

{

int q[20],p[50],c=0,c1,d,f,i,j,k=0,n,r,t,b[20],c2[20];

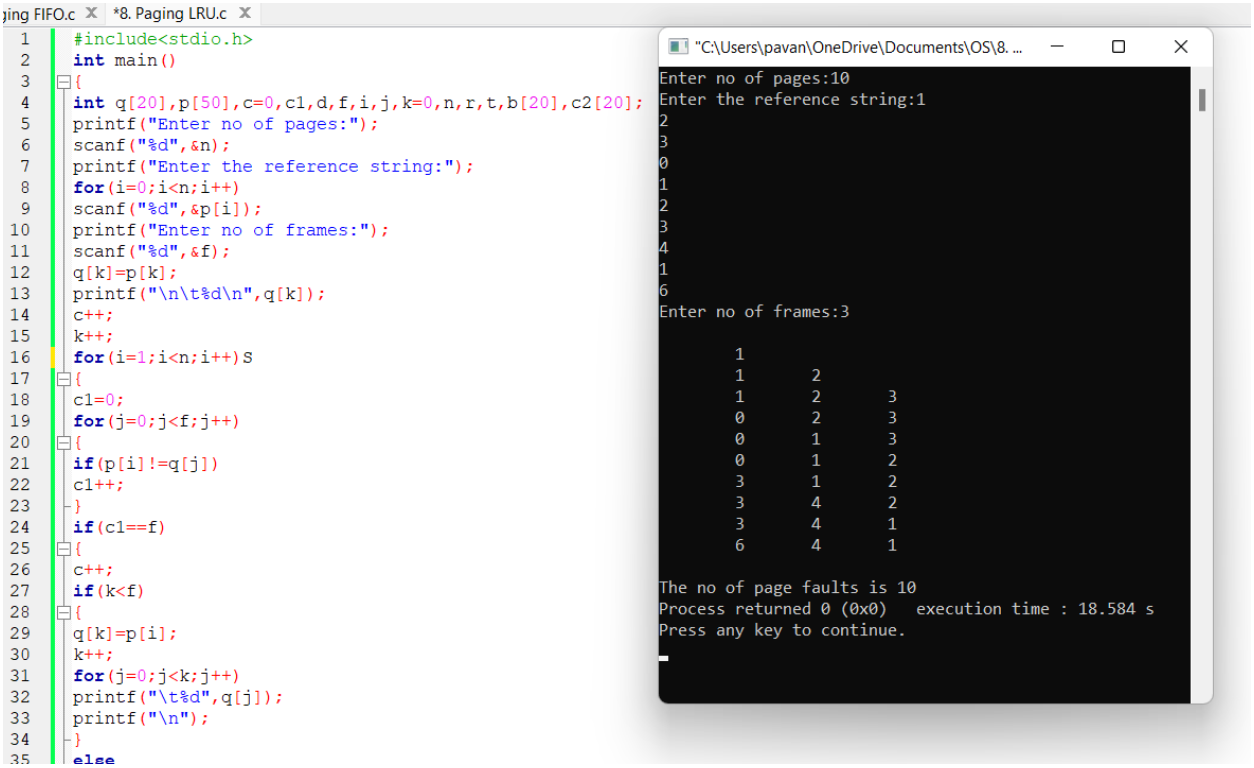
printf("Enter no of pages:");

```
scanf("%d",&n);
printf("Enter the reference string:");
for(i=0;i<n;i++)
scanf("%d",&p[i]);
printf("Enter no of frames:");
scanf("%d",&f);
q[k]=p[k];
printf("\n\t%d\n",q[k]);
c++;
k++;
for(i=1;i<n;i++)S
{
c1=0;
for(j=0;j<f;j++)
{
if(p[i]!=q[j])
c1++;
}
if(c1==f)
{
c++;
if(k<f)
{
q[k]=p[i];
k++;
for(j=0;j<k;j++)
printf("\t%d",q[j]);
printf("\n");
```

```
}
else
{
for(r=0;r<f;r++)
{
c2[r]=0;
for(j=i-1;j<n;j--)
{
if(q[r]!=p[j])
c2[r]++;
else
break;
}
}
for(r=0;r<f;r++)
b[r]=c2[r];
for(r=0;r<f;r++)
{
for(j=r;j<f;j++)
{
if(b[r]<b[j])
{
t=b[r];
b[r]=b[j];
b[j]=t;
}
}
}
```

```
for(r=0;r<f;r++)
{
if(c2[r]==b[0])
q[r]=p[i];
printf("\t%d",q[r]);
}
printf("\n");
}
}
}
printf("\nThe no of page faults is %d",c);
}
```

**Output :-**



**9. Paging – Optimal :-**

**Program :-**

```
#include<stdio.h>
int main()
{
int no_of_frames, no_of_pages, frames[10], pages[30], temp[10], flag1, flag2, flag3, i, j, k, pos,
max, faults = 0;
printf("Enter number of frames: ");
scanf("%d", &no_of_frames);
printf("Enter number of pages: ");
scanf("%d", &no_of_pages);
printf("Enter page reference string: ");
for(i = 0; i < no_of_pages; ++i)
{
scanf("%d", &pages[i]);
}
for(i = 0; i < no_of_frames; ++i)
{
frames[i] = -1;
}
for(i = 0; i < no_of_pages; ++i)
{
flag1 = flag2 = 0;
for(j = 0; j < no_of_frames; ++j)
{
if(frames[j] == pages[i])
{
flag1 = flag2 = 1;
```

```
break;
}
}
if(flag1 == 0)
{
for(j = 0; j < no_of_frames; ++j)
{
if(frames[j] == -1)
{
faults++;
frames[j] = pages[i];
flag2 = 1;
break;
}
}
}
if(flag2 == 0)
{
flag3 =0;
for(j = 0; j < no_of_frames; ++j)
{
temp[j] = -1;
for(k = i + 1; k < no_of_pages; ++k)
{
if(frames[j] == pages[k])
{
temp[j] = k;
break;
```

```
}
}
}
for(j = 0; j < no_of_frames; ++j)
{
if(temp[j] == -1)
{
pos = j;
flag3 = 1;
break;
}
}
if(flag3 ==0)
{
max = temp[0];
pos = 0;
for(j = 1; j < no_of_frames; ++j)
{
if(temp[j] > max)
{
max = temp[j];
pos = j;
}
}
}
frames[pos] = pages[i];
faults++;
}
```

```
printf("\n");
for(j = 0; j < no_of_frames; ++j)
{
printf("%d\t", frames[j]);
}
}
printf("\n\nTotal Page Faults = %d", faults);
return 0;
}
```

**Output :-**

FIFO.c | *8. Paging LRU.c | Paging - Optimal.c

```c
#include<stdio.h>
int main()
{
int no_of_frames, no_of_pages, frames[10], pages[30], temp[10], flag1, flag2, flag3, i, j, k, pos,
max, faults = 0;
printf("Enter number of frames: ");
scanf("%d", &no_of_frames);
printf("Enter number of pages: ");
scanf("%d", &no_of_pages);
printf("Enter page reference string: ");
for(i = 0; i < no_of_pages; ++i)
{
scanf("%d", &pages[i]);
}
for(i = 0; i < no_of_frames; ++i)
{
frames[i] = -1;
}
for(i = 0; i < no_of_pages; ++i)
{
flag1 = flag2 = 0;
for(j = 0; j < no_of_frames; ++j)
{
if(frames[j] == pages[i])
{
flag1 = flag2 = 1;
break;
}
}
if(flag1 == 0)
{
for(j = 0; j < no_of_frames; ++j)
{
if(frames[j] == -1)
{
faults++;
frames[j] = pages[i];
```

"C:\Users\pavan\OneDrive\Documents\OS\Paging...

```
Enter number of frames: 3
Enter number of pages: 8
Enter page reference string: 1
0
2
3
0
4
5
6

1       -1      -1
1       0       -1
1       0       2
3       0       2
3       0       2
4       0       2
5       0       2
6       0       2

Total Page Faults = 7
Process returned 0 (0x0)   execution time : 14.211 s
Press any key to continue.
```

**10 . Sequential file allocation :-**

**Program :-**

```
#include <stdio.h>
typedef struct
{
int usn;
char name[25];
int m1,m2,m3;
}
STD;
STD s;
void display(FILE *);
int search(FILE *,int);
void main()
{
int i,n,usn_key,opn;
FILE *fp;
printf(" How many Records ? ");
scanf("%d",&n);
fp=fopen("stud.dat","w");
for (i=0;i<n;i++)
{
printf("Read the Info for Student: %d (usn,name,m1,m2,m3) \n",i+1);
scanf("%d%s%d%d%d",&s.usn,s.name,&s.m1,&s.m2,&s.m3);
fwrite(&s,sizeof(s),1,fp);
}
fclose(fp);
fp=fopen("stud.dat","r");
```

```
do
{
printf("Press 1- Display\t 2- Search\t 3- Exit\t Your Option?");
scanf("%d",&opn);
switch(opn)
{
case 1: printf("\n Student Records in the File \n");
display(fp);
break;
case 2: printf(" Read the USN of the student to be searched ?");
scanf("%d",&usn_key);
if(search(fp,usn_key))
{
printf("Success ! Record found in the file\n");
printf("%d\t%s\t%d\t%d\t%d\n",s.usn,s.name,s.m1,s.m2,s.m3);
}
else
printf(" Failure!! Record with USN %d not found\n",usn_key);
break;
case 3: printf(" Exit!! Press a key . . .");
break;
default: printf(" Invalid Option!!! Try again !!!\n");
break;
}
}
while(opn != 3);
fclose(fp);
}
```

```
/* End of main() */
void display(FILE *fp)
{
rewind(fp);
while(fread(&s,sizeof(s),1,fp))
printf("%d\t%s\t%d\t%d\t%d\n",s.usn,s.name,s.m1,s.m2,s.m3);
}
int search(FILE *fp, int usn_key)
{
rewind(fp);
while(fread(&s,sizeof(s),1,fp))
if( s.usn == usn_key) return 1;
return 0;
}
```

**Output :-**