**PROGRAM:**

#include<stdio.h>

#include<sys/sem.h>

#define BUF 5

inti,full=0, empty=BUF, x=0,front=-1,rear=-1,np=0,nc=0;

struct elements

{

int item;

intval;

}q[100];

void producer();

void consumer();

int wait(int);

int signal(int);

int main()

{

intn,sem\_set\_id,ch;

sem\_set\_id=semget(IPC\_PRIVATE,1,0600);

printf("SEMAPHORE CREATED....SEMAPHORE ID...%d\n",sem\_set\_id);

printf("\n The buffer size is %d ",BUF);

do

{

printf("\n Enter 1) for active Producer ");

printf("\n Enter 2) for active consumer ");

printf("\n Enter 3) to quit \n ");

scanf("%d",&ch);

switch(ch)

{

case 1 : producer();

break;

case 2 : consumer();

break;

case 3 : break;

/\* case 4 : printf("Front = %d",front) ;

printf("rear = %d",rear);

printf("full = %d",full);

printf("empty = %d",empty);

\*/

}

}while(ch!=3);

}

int wait(int s)

{

return (--s);

}

int signal(int a)

{

return(++a);

}

void move()

{

inti;

front=wait(front);

//printf("%d -> front ",front);

for(i=0;i<=rear-1;i++)

{

q[i]=q[i+1];

}

rear=wait(rear);

// printf("%d - > rear ",rear);

// printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

// for(i=0;i<=rear;i++)

// printf("%d item, %d val \n",q[i].item,q[i].val);

}

void producer()

{

int e;

if(full!=BUF)

{

x++;

full=signal(full);

empty=wait(empty);

printf("\n Enter the element ");

scanf("%d",&e);

if(front==-1)

{

front=0;

rear=0;

}

else rear+=1;

q[rear].val=e;

q[rear].item=x;

printf("\n producer produce the process item %d => Elements %d \t",x,e);

printf("\n");

}

elseprintf("\n The Buffer is Full");

}

void consumer()

{

int e;

if(empty!=BUF)

{

full=wait(full);

empty=signal(empty);

printf("\n");

// printf("\*\* %d \*\*",front);

e=q[front].val;

if(front==rear)

{

printf("\n consumer consumes the process item %d => Element %d",q[front].item,e);

front=-1;

rear=-1;

return;

}

else front+=1;

// if(front==-1)

// printf("\n consumer consumes the process item %d => Element %d",front+2,e);

printf("\n consumer consumes the process item %d => Element %d",q[front-1].item,e);

move();

}

else printf("\n The buffer is empty");

}