NAME: *Hariprasad K K*

REG NO: *19BCE7079*

COURSE : *Operating System*

DATE : *11-11-2021*

CODE:

FIL11.C

```
#include<stdio.h>
#include<stdlib.h>

int mutex=1,full=0,empty=3,x=0;

int main()
{
    int n;
    void producer();
    void consumer();
    int wait(int);
    int signal(int);
    printf("\n1.Producer\n2.Consumer\n3.Exit");
    while(1)
    {
        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;
        }
    }
```

```
    return 0;
}

int wait(int s)
{
    return (--s);
}

int signal(int s)
{
    return(++s);
}

void producer()
{
    mutex=wait(mutex);
    full=signal(full);
    empty=wait(empty);
    x++;
    printf("\nProducer produces the item %d",x);
    mutex=signal(mutex);
}
void consumer()
{
    mutex=wait(mutex);
    full=wait(full);
    empty=signal(empty);
    printf("\nConsumer consumes item %d",x);
    x--;
    mutex=signal(mutex);
}
```

FIL15.C

```
#include<stdio.h>


#define max 25

void main()

{

int frag[max],b[max],f[max],i,j,nb,nf,temp;

static int bf[max],ff[max];
```

```
printf("\nEnter the number of blocks:");
scanf("%d",&nb);
printf("Enter the number of files:");
scanf("%d",&nf);
printf("\nEnter the size of the blocks:-\n");
for(i=1;i<=nb;i++)
{
printf("Block %d:",i);
scanf("%d",&b[i]);
}
printf("Enter the size of the files:-\n");
for(i=1;i<=nf;i++)
{
printf("File %d:",i);
scanf("%d",&f[i]);
}
for(i=1;i<=nf;i++)
{
for(j=1;j<=nb;j++)
{
if(bf[j]!=1)
{
temp=b[j]-f[i];
if(temp>=0)
{
ff[i]=j;
break;
}
```

```
}
}
frag[i]=temp;
bf[ff[i]]=1;
}
printf("\nFile_no:\tFile_size :\tBlock_no:\tBlock_size:\tFragment");
for(i=1;i<=nf;i++)
printf("\n%d\t\t%d\t\t%d\t\t%d\t\t%d",i,f[i],ff[i],b[ff[i]],frag[i])
;

}
```

FIL16.C

```
#include<stdio.h>

#define max 25
void main()
{
int frag[max],b[max],f[max],i,j,nb,nf,temp,lowest=10000;
static int bf[max],ff[max];

printf("\nEnter the number of blocks:");
scanf("%d",&nb);
printf("Enter the number of files:");
scanf("%d",&nf);
printf("\nEnter the size of the blocks:-\n");
for(i=1;i<=nb;i++)
{
```

```
printf("Block %d:",i);
scanf("%d",&b[i]);
}
printf("Enter the size of the files:-\n");
for(i=1;i<=nf;i++)
{
printf("File %d:",i);
scanf("%d",&f[i]);
}
for(i=1;i<=nf;i++)
{
for(j=1;j<=nb;j++)
{
if(bf[j]!=1)
{
temp=b[j]-f[i];
if(temp>=0)
if(lowest>temp)
{
ff[i]=j;
lowest=temp;
}
}
}
frag[i]=lowest;
bf[ff[i]]=1;
lowest=10000;
}
```

```
printf("\nFile_no  \tFile_size  \tBlock_no  \tBlock_size
\tFragment");

for(i=1;i<=nf && ff[i]!=0;i++)

printf("\n%d\t\t%d\t\t%d\t\t%d\t\t%d",i,f[i],ff[i],b[ff[i]],frag[i])
;


}
```

FIL14.C

```
    #include<stdio.h>

    void main() {
      int
k=0,output[10],d=0,t=0,ins[5],i,avail[5],allocated[10][5],need[10][5
],MAX[10][5],pno,P[10],j,rz, count=0;

      printf("\n Enter the number of resources : ");
      scanf("%d", &rz);
      printf("\n enter the max instances of each resources\n");
      for (i=0;i<rz;i++) {
            avail[i]=0;
            printf("%c= ",(i+97));
            scanf("%d",&ins[i]);
      }
      printf("\n Enter the number of processes : ");
      scanf("%d", &pno);
      printf("\n Enter the allocation matrix \n     ");
      for (i=0;i<rz;i++)
      printf(" %c",(i+97));
      printf("\n");
      for (i=0;i <pno;i++) {
            P[i]=i;
            printf("P[%d]  ",P[i]);
            for (j=0;j<rz;j++) {
                  scanf("%d",&allocated[i][j]);
                  avail[j]+=allocated[i][j];
            }
      }
      printf("\nEnter the MAX matrix \n     ");
      for (i=0;i<rz;i++) {
            printf(" %c",(i+97));
            avail[i]=ins[i]-avail[i];
      }
      printf("\n");
      for (i=0;i <pno;i++) {
            printf("P[%d]  ",i);
            for (j=0;j<rz;j++)
             scanf("%d", &MAX[i][j]);
      }
```

```
      printf("\n");
      A: d=-1;
      for (i=0;i <pno;i++) {
            count=0;
            t=P[i];
            for (j=0;j<rz;j++) {
                  need[t][j] = MAX[t][j]-allocated[t][j];
                  if(need[t][j]<=avail[j])
                   count++;
            }
            if(count==rz) {
                  output[k++]=P[i];
                  for (j=0;j<rz;j++)
                  avail[j]+=allocated[t][j];
            } else
             P[++d]=P[i];
      }
      if(d!=-1) {
            pno=d+1;
            goto A;
      }
      printf("\t <");
      for (i=0;i<k;i++)
      printf(" P[%d] ",output[i]);
      printf(">");

    }
```

FIL18.C

```
#include<stdio.h>


#define max 25

void main()

{

int frag[max],b[max],f[max],i,j,nb,nf,temp,highest=0;

static int bf[max],ff[max];



printf("\nEnter the number of blocks:");

scanf("%d",&nb);

printf("Enter the number of files:");

scanf("%d",&nf);
```

```
printf("\nEnter the size of the blocks:-\n");
for(i=1;i<=nb;i++)
{
printf("Block %d:",i);
scanf("%d",&b[i]);
}
printf("Enter the size of the files:-\n");
for(i=1;i<=nf;i++)
{
printf("File %d:",i);
scanf("%d",&f[i]);
}
for(i=1;i<=nf;i++)
{
for(j=1;j<=nb;j++)
{
if(bf[j]!=1) //if bf[j] is not allocated
{
temp=b[j]-f[i];
if(temp>=0)
if(highest<temp)
{
ff[i]=j;
highest=temp;
}
}
}
frag[i]=highest;
```

```
bf[ff[i]]=1;

highest=0;

}

printf("\nFile_no  \tFile_size  \tBlock_no  \tBlock_size
\tFragment");

for(i=1;i<=nf;i++)

printf("\n%d\t\t%d\t\t%d\t\t%d\t\t%d",i,f[i],ff[i],b[ff[i]],frag[i])
;



}
```

SCREENSHOTS :