EX NO:9 **FILE ALLOCATION STRATEGY**

DATE:

AIM:

To write C program that implements concepts of file allocation strategy using sequence,indexed and linked allocation methods

SEQUENTIAL FILE ALLOCATION:

ALGORITHM:

Step 1: start

Step 2: Get the number of memory partition and their sizes.

Step 3: Get the number of processes and values of block size for each process.

Step 4: First fit algorithm searches all the entire memory block until a hole which is big enough is encountered. It allocates that memory block for the requesting process.

Step 5: Best-fit algorithm searches the memory blocks for the smallest hole which can be allocated to requesting process and allocates it.

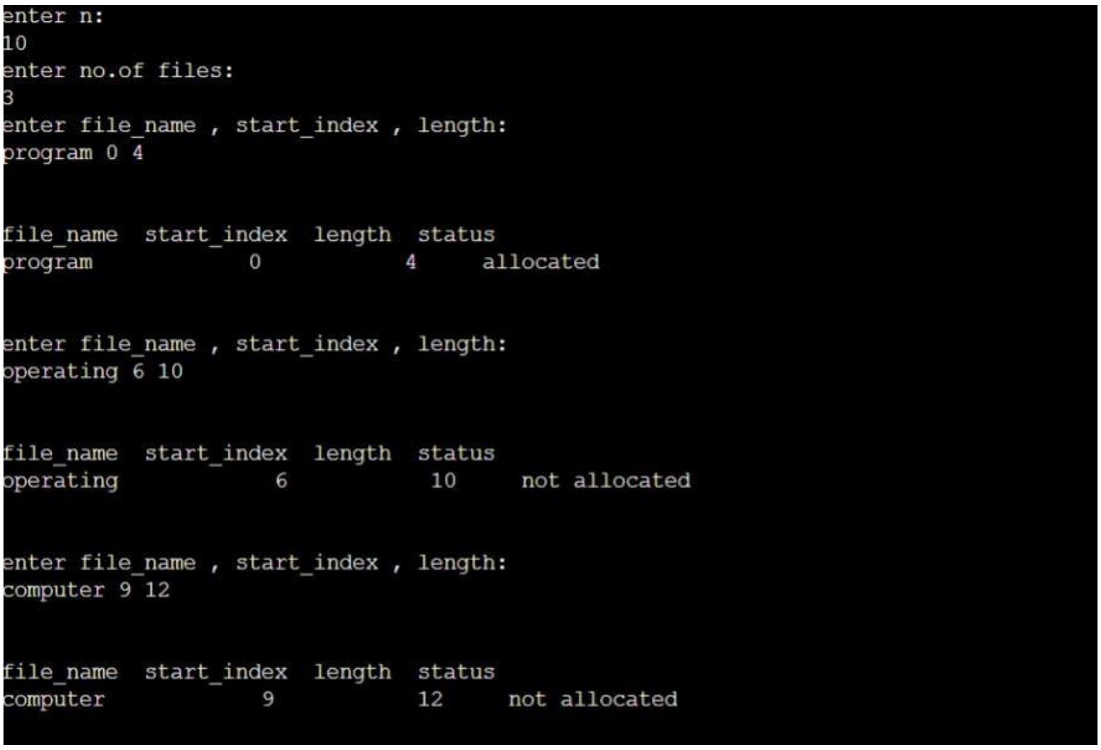
Step 6: Worst fit algorithm searches the memory blocks for the largest hole and allocates it to the process. Step 7: Analyses all the three memory management techniques and display the best algorithm which utilizes the memory resources effectively and efficiently.

Step 8:stop.

PROGRAM:

```
#include<stdio.h> #include<conio.h> int main()

{

int n,f,j,m,sts=0; printf("enter n:\n"); scanf("%d",&n); int buffer[n],si[100],l[100]; for(int
i=0;i<n;i++)

{

buffer[i]=0;

}

printf("enter no.of files:\n");

scanf("%d",&f); char str[f][100];

for(int i=0;i<f;i++)

{

printf("enter file_name , start_index , length:\n"); scanf("%s%d%d",str[i],&si[i],&l[i]);
printf("\n\n");

printf("file_name start_index length status\n");
printf("%s      %d      %d      ",str[i],si[i],l[i]); for(j=si[i];j<si[i]+l[i];j++)
{
if(buffer[j]==0)
{
buffer[j]=1;
}
else
{
printf("not allocated\n");
printf("\n\n");
break;
}
}
if(j==si[i]+l[i])
{
printf("allocated\n");
printf("\n\n");
sts=1;
}
}
getch();
}
```

OUTPUT:



```
enter n:
10
enter no.of files:
3
enter file_name , start_index , length:
program 0 4


file_name  start_index  length  status
program             0           4     allocated


enter file_name , start_index , length:
operating 6 10


file_name  start_index  length  status
operating             6           10     not allocated


enter file_name , start_index , length:
computer 9 12


file_name  start_index  length  status
computer            9           12     not allocated
```

INDEXED FILE ALLOCATION:

ALGORITHM:

Step 1: Start.

Step 2: Let n be the size of the buffer

Step 3: check if there are any producer

Step 4: if yes check whether the buffer is full

Step 5: If no the producer item is stored in the buffer

Step 6: If the buffer is full the producer has to wait

Step 7: Check there is any consumer.If yes check whether the buffer is empty

Step 8: If no the consumer consumes them from the buffer

Step 9: If the buffer is empty, the consumer has to wait.

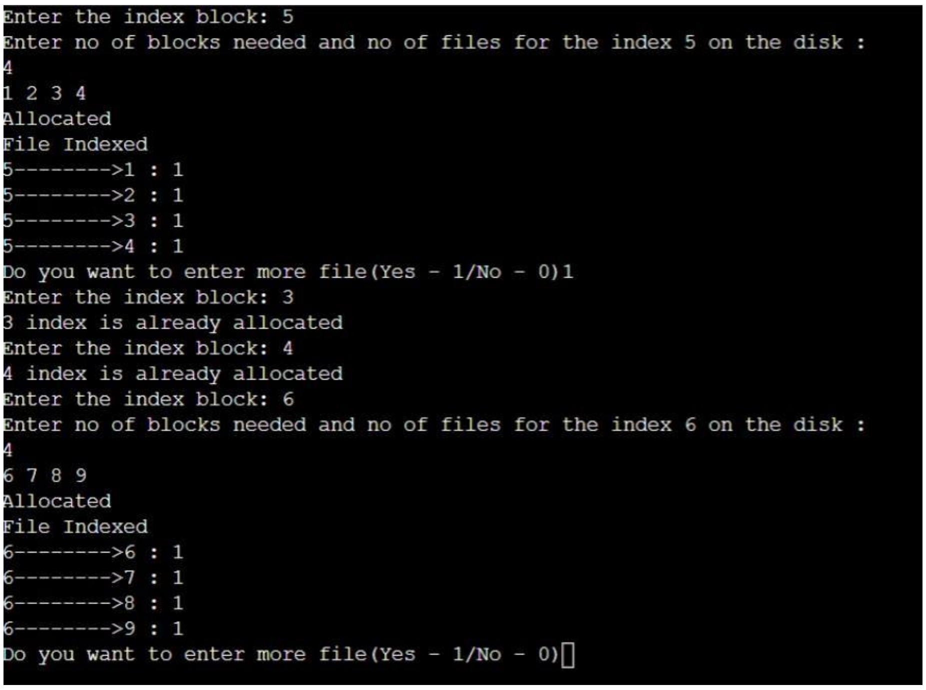
Step 10: stop

PROGRAM:

```
#include<conio.h>

#include<stdlib.h>

#include<stdio.h>

void main()
```

```
{
int f[50], index[50],i, n, st, len, j, c, k, ind,count=0;
for(i=0;i<50;i++)
f[i]=0;
x:printf("Enter the index block: ");
scanf("%d",&ind);
if(f[ind]!=1)
{
printf("Enter no of blocks needed and no of files for the index %d on the disk : \n", ind);
scanf("%d",&n);
}
else
{
printf("%d index is already allocated \n",ind);
goto x;
}
y: count=0;
for(i=0;i<n;i++)
{
scanf("%d", &index[i]);
if(f[index[i]]==0)
count++;
}
if(count==n)
{
for(j=0;j<n;j++)
f[index[j]]=1;
printf("Allocated\n");
printf("File Indexed\n");
for(k=0;k<n;k++)
printf("%d-------->%d : %d\n",ind,index[k],f[index[k]]);
}
else
{
printf("File in the index is already allocated \n");
printf("Enter another file indexed");
goto y;
}
printf("Do you want to enter more file(Yes - 1/No - 0)");
scanf("%d", &c);
if(c==1)
goto x;
else
exit(0);
getch();
}
```

OUTPUT:



```
Enter the index block: 5
Enter no of blocks needed and no of files for the index 5 on the disk :
4
1 2 3 4
Allocated
File Indexed
5-------->1 : 1
5-------->2 : 1
5-------->3 : 1
5-------->4 : 1
Do you want to enter more file(Yes - 1/No - 0)1
Enter the index block: 3
3 index is already allocated
Enter the index block: 4
4 index is already allocated
Enter the index block: 6
Enter no of blocks needed and no of files for the index 6 on the disk :
4
6 7 8 9
Allocated
File Indexed
6-------->6 : 1
6-------->7 : 1
6-------->8 : 1
6-------->9 : 1
Do you want to enter more file(Yes - 1/No - 0)
```

LINKED FILE ALLOCATION:

ALGORITHM:

Step 1: start

Step 2: When the page is required replace the page at the head of the queue

Step 3: Now the new page is inserted at the tail of the queue

Step 4: Create a stack

Step 5: When the page fault occurs replace page present at the bottom of the stack

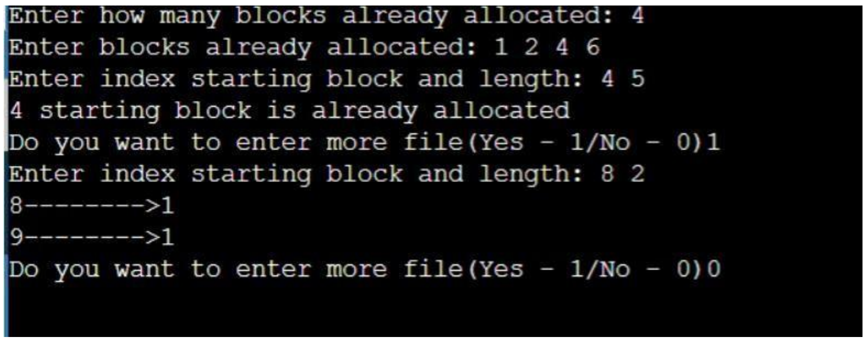
Step 6: Stop.

PROGRAM:

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
void main()
{
int f[50], p,i, st, len, j, c, k, a;
for(i=0;i<50;i++)
f[i]=0;
printf("Enter how many blocks already allocated: ");
scanf("%d",&p);
printf("Enter blocks already allocated: ");
for(i=0;i<p;i++)
{
scanf("%d",&a);
f[a]=1;
}
x: printf("Enter index starting block and length: ");
scanf("%d%d", &st,&len);
k=len;
if(f[st]==0)
{
for(j=st;j<(st+k);j++)
{
```

```
if(f[j]==0)
{ f[j]=
1;
printf("%d-------->%d\n",j,f[j]);
}
else
{
printf("%d Block is already allocated \n",j);
k++;
}
}
}
else
Name:Iswaryapriyadharshni Rollno:21CSE072 Pageno:
printf("%d starting block is already allocated \n",st);
printf("Do you want to enter more file(Yes - 1/No - 0)");
scanf("%d", &c);
if(c==1)
goto x;
else
exit(0);
getch();
}
```

OUTPUT:



```
Enter how many blocks already allocated: 4
Enter blocks already allocated: 1 2 4 6
Enter index starting block and length: 4 5
4 starting block is already allocated
Do you want to enter more file(Yes - 1/No - 0)1
Enter index starting block and length: 8 2
8-------->1
9-------->1
Do you want to enter more file(Yes - 1/No - 0)0
```

observation(20)	
record(5)	
total(25)	

RESULT:

Thus the program for file allocation strategy is executed and the output is verified