EXPERIMENT NO:-8

AIM: - Write a program to implement dynamic partitioning placement algorithms i.e Best Fit, First –Fit and Worst –Fit.

THEORY:-

One of the simplest methods for memory allocation is to divide memory into several fixed-sized partitions. Each partition may contain exactly one process. In this multiple- partition method, when a partition is free, a process is selected from the input queue and is loaded into the free partition. When the process terminates, the partition becomes available for another process. The operating system keeps a table indicating which parts of memory are available and which are occupied. Finally, when a process arrives and needs memory, a memory section large enough for this process is provided. When it is time to load or swap a process into main memory, and if

there is more than one free block of memory of sufficient size, then the operating system must decide which free block to allocate. Best-fit strategy chooses the block that is closest in size to the request. First- fit chooses the first available block that is large enough. Worst-fit chooses the largest available block.

/*Program to implement BEST-FIT dynamic partitioning placement algorithms*/

```
#include<stdio.h>
#include<conio.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];
clrscr();
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]);
getch();
}
```

OUTPUT:

```
user@user-H81M-S:~$ ./exp8_1

        Memory Mannagement scheme-Best Fit
Enter the number of blocks:3
Enter the number of files:3

Enter the size of the blocks:-
Block 1:43
Block 2:22
Block 3:53
Enter the size of the files :-
File 1:33
File 2:43
File 3:22

File No File Size       Block No        Block Size      Fragment
1               33              2               22              -11
```

/*Program to implement Worst-Fit dynamic partitioning placement algorithm*/

```
#include<stdio.h>
#include<conio.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];
clrscr();
printf("\n\tMemory Management Scheme - Worst Fit");
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:\tFragement");
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]);
getch();
}
```

OUTPUT:

```
user@user-H81M-S:~$ gcc -o exp8-2 exp8-2.c
user@user-H81M-S:~$ ./exp8-2

        Memory Management Scheme - Worst Fit
Enter the number of blocks:4
Enter the number of files:4

Enter the size of the blocks:-
Block 1:23
Block 2:45
Block 3:33
Block 4:18
Enter the size of the files :-
File 1:54
File 2:16
File 3:25
File 4:12

File_no:        File_size :     Block_no:       Block_size:     Fragement
1               54              0               0               0
2               16              2               45              29
3               25              3               33              8
4               12              1               23              11
```

/*Program to implement First-Fit dynamic partitioning placement algorithm*/

```
#include<stdio.h>
#include<conio.h>
#define max 25
void main()
{
int frag[max],b[max],f[max],i,j,nb,nf,temp;
static int bf[max],ff[max];
clrscr();
printf("\n\tMemory Management Scheme - First Fit");
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:\tFragement");
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]);
getch();
}
```

OUTPUT:

```
user@user-H81M-S:~$ gcc -o exp8_3 exp8_3.c
user@user-H81M-S:~$ ./exp8_3

        Memory Management Scheme - First Fit
Enter the number of blocks:5
Enter the number of files:5

Enter the size of the blocks:-
Block 1:45
Block 2:18
Block 3:23
Block 4:56
Block 5:52
Enter the size of the files :-
File 1:58
File 2:12
File 3:30
File 4:33
File 5:15

File_no:        File_size :     Block_no:       Block_size:     Fragement
1               58              0               0               -6
2               12              1               45              33
3               30              4               56              26
4               33              5               52              19
5               15              2               18              3
```

CONCLUSION:-Thus we have studied and implemented dynamic partitioning placementalgorithms Best Fit, First –Fit and Worst –Fit.

SIGN AND REMARK

R1 (3 Marks)	R2 (3 Marks)	R3 (3 Marks)	R4 (3 Mark)	R5 (3 Mark)	Total (15 Marks)	Signature