\

| | |
|---|---|
| **EX NO: 12** | **FILE ALLOCATION STRATEGIES** |
| **DATE:** | |

## A. SEQUENTIAL FILE ALLOCATION AIM:

To Write a C Program to implement Sequential File Allocation method.

**ALGORITHM:**

1: Start the program.

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

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

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

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

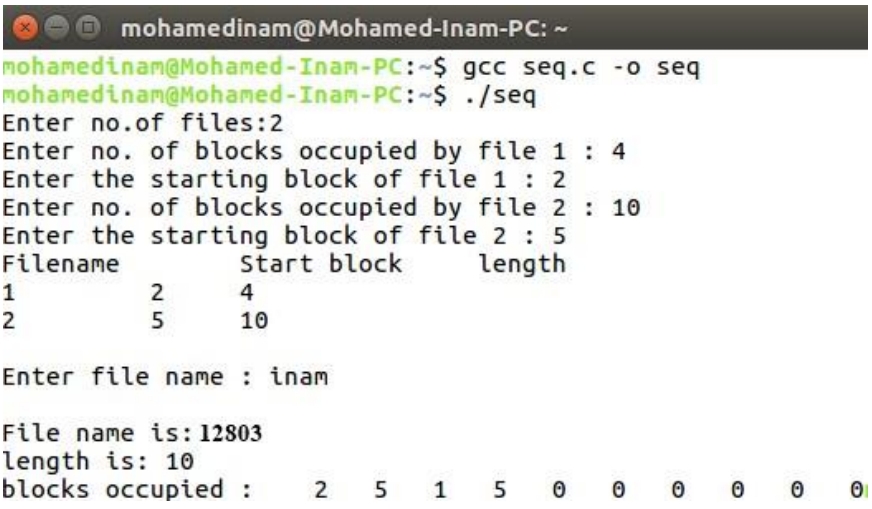
6: Worst fit algorithm searches the memory blocks for the largest hole and allocates it to the process.

7: Analyses all the three memory management techniques and display the best algorithm which utilizes the memory resources effectively and efficiently. 8: Stop the program.

**PROGRAM:**

```
#include<stdio.h>
main()
{
 int n,i,j,b[20],sb[20],t[20],x,c[20][20];
clrscr();
printf("Enter no.of files:");
 scanf("%d",&n);
 for(i=0;i<n;i++)
 {
        printf("Enter no. of blocks occupied by file%d",i+1);
        scanf("%d",&b[i]);
        printf("Enter the starting block of file%d",i+1);
        scanf("%d",&sb[i]);
        t[i]=sb[i];
        for(j=0;j<b[i];j++)
                c[i][j]=sb[i]++;
        }
printf("Filename\tStart block\tlength\n");
 for(i=0;i<n;i++)
        printf("%d\t  %d \t%d\n",i+1,t[i],b[i]);
printf("Enter file name:");
 scanf("%d",&x);
printf("File name is:%d",x);
printf("length is:%d",b[x-1]);
printf("blocks occupied:");
 for(i=0;i<b[x-1];i++)
        printf("%4d",c[x-1][i]);
}
```

**OUTPUT:**






**RESULT:**

Thus the SEQUENTIAL file allocation method is implemented successfully.