

Aim:

Aim: Implementation of First-fit allocation technique.

Description: 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 First-fit chooses the first available block that is large enough.

Source Code:

Firstfit.c

```
#include <stdio.h>
#include <conio.h>
#define max 25

void main()
{
    int frag[max], b[max], f[max], bf[max], ff[max];
    int i, j, nb, nf, temp;

    printf("\tMemory Management Scheme - First Fit\n");
    printf("Enter the number of blocks: ");
    scanf("%d", &nb);
    printf("Enter the number of files: ");
    scanf("%d", &nf);

    printf("Enter 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]);
    }

    // Initialize bf[] to mark all blocks as free initially
    for (i = 1; i <= nb; i++)
        bf[i] = 0;

    // Allocation logic
    for (i = 1; i <= nf; i++)
    {
        for (j = 1; j <= nb; j++)
        {
```

```

ficient
    {
        ff[i] = j;           // Allocate file i to block j
        frag[i] = b[j] - f[i]; // Calculate fragmentation
        bf[j] = 1;           // Mark block as allocated
        break;
    }
}
if (j > nb) // If no suitable block is found
{
    ff[i] = 0; // File i is not allocated
    frag[i] = 0; // No fragmentation
}
}

// Display results
printf("File_no:\tFile_size:\tBlock_no:\tBlock_size:\tFragment\n");
for (i = 1; i <= nf; i++)
{
    if (ff[i] != 0)
    {
        printf("%d\t%d\t%d\t%d\t%d\n", i, f[i], ff[i], b[ff[i]], frag
[i]);
    }
    else
    {
        printf("%d\t%d\t\t\t\tNot Allocated\n", i, f[i]);
    }
}
}

```

Execution Results - All test cases have succeeded!

Test Case - 1				
User Output				
Memory Management Scheme - First Fit 4				
Enter the number of blocks: 4				
Enter the number of files: 4				
Enter the size of the blocks: 10				
Block 1: 10				
Block 2: 20				
Block 3: 30				
Block 4: 40				
Enter the size of the files: 11				
File 1: 11				
File 2: 12				
File 3: 13				
File 4: 14				
File_no:	File_size:	Block_no:	Block_size:	Fragment
1	11	2	20	9
2	12	3	30	18
3	13	4	40	27
4	14	Not Allocated		