

Ex. No.: 10b

FIRST FIT

Date: 2.4.2025

Aim:

To write a C program for implementation of memory allocation methods for fixed partition using First Fit.

Algorithm:

1. Define the maximum limit as #define max 25.
 2. Declare variables: frag[max], b[max], f[max], i, j, nb, nf, temp, bf[max], ff[max].
 3. Input the number of blocks (nb) and files (nf).
 4. Input the size of each block and file using loops.
 5. For each file, search for the first block that is free and large enough to accommodate it.
 6. If found, allocate that block to the file and calculate internal fragmentation.
 7. Mark the block as used.
 8. Print the allocated block and fragmentation details.
-

Program Code (first_fit.c):

```
#include <stdio.h>

#define max 25

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

    printf("Enter number of blocks: ");
    scanf("%d", &nb);

    printf("Enter number of files: ");
    scanf("%d", &nf);
```

}

}

}

}

```
printf("%d\t%d\t\t%d\t\t%d\t\t%d\n", i+1, f[i], ff[i]+1, b[ff[i]], frag[i]);
```

```
return 0;
```

}

Sample Output:

Enter number of blocks: 5

Enter number of files: 4

Enter size of each block:

Block 1: 100

Block 2: 500

Block 3: 200

Block 4: 300

Block 5: 600

Enter size of each file:

File 1: 212

File 2: 417

File 3: 112

File 4: 426

File No	File Size	Block No	Block Size	Fragment
1	212	2	500	288
2	417	5	600	183
3	112	3	200	88
4	426	0	0	0 <-- Not allocated

Result:

Thus, the First Fit memory allocation technique for fixed partitioning was implemented successfully in C.