

# OS PRACTICAL EXAM

Name: Sakshi Jain

Adm No: U19CS048

---

1. Write a program to implement and simulate the Memory Management with Fixed Partition algorithm (MFT). Consider the scenario where the memory is divided in two parts and the process is fit into it. The process which is best suited will be placed in the particular memory where it suits. In MFT, the memory is partitioned into fixed size partitions and each job is assigned to a partition. The memory assigned to a partition does not change. In MVT, each job gets just the amount of memory it needs. That is, the partitioning of memory is dynamic and changes as jobs enter and leave the system. MVT is a more "efficient" user of resources. MFT suffers with the problem of internal fragmentation and MVT suffers with external fragmentation.

**Source Code:**

```
#include<stdio.h>

int main()
{
    int ms, os, n, ps, t, curr;
    printf("Enter The Total Size of Main Memory: ");
    scanf("%d",&ms);
    printf("Enter The Size of Memory required by OS: ");
    scanf("%d",&os);
    ms=ms-os;
    printf("Enter The No of Partitions of Main Memory: ");
    scanf("%d",&n);
    int used[n], int_frag[n];
    int ext_frag_start, ext_frag_end;
    ext_frag_start=0;
    ext_frag_end=ms-1;

    for(int i=0; i<n; ++i)
    {
        used[i]=0;
        int_frag[i]=-1;
    }

    ps=ms/n;
    printf("\n-----\n");
    curr=0;
    int size, id;

    do
    {
        printf("Enter The Process ID: ");
        scanf("%d", &id);
        printf("Enter The Size of Process: ");
```

```

scanf("%d", &size);
if(size<=ps)
{
    used[curr]=id;
    int_frag[curr]=ps-size;
    curr++;
    ext_frag_start=curr*ps;
}
else
{
    printf("The Size of process is more than the Partition Size, so Memeoory can't
be allocated to The Process!!!\n");
}

if(curr==n)
{
    printf("\nAll The Partioned are now occupied rmeove some process to enter new
process!!!\n");
    ext_frag_start=-1;
    ext_frag_end=-1;
    break;
}
printf("Do you want to continue? 0 for NO & 1 for YES\n");
int ch;
scanf("%d", &ch);

if(ch==0)
{
    break;
}
}

while(1);
printf("\n-----\n");
printf("Total Main Main Memory Size: %d\n", ms+os);
printf("Total Size required by OS: %d\n", os);
printf("Total Main Main Memory Size available for Process: %d\n", ms);
printf("Total No. of Partions: %d\n", n);
printf("Main Memory State: ");

for(int i=0; i<n; ++i)
{
    printf("%d, ", used[i]);
}
printf("\nInternal Fragmentation: ");

for(int i=0; i<n; ++i)
{
    printf("%d, ", int_frag[i]);
}

printf("\nExternal Fragmentation: ");
printf("\nStart of External Fragmentation: %d\nEnd of External Fragmentation: %d\n",
ext_frag_start, ext_frag_end);

```

```
    return 0;
}
```

## Output:

```
input
Enter The Total Size of Main Memory: 10
Enter The Size of Memory required by OS: 2
Enter The No of Partitions of Main Memory: 2

-----
Enter The Process ID: 1
Enter The Size of Process: 3
Do you want to continue? 0 for NO & 1 for YES
1
Enter The Process ID: 2
Enter The Size of Process: 1

All The Partioned are now occupied rmeove some process to enter new process!!!

-----
Total Main Main Memory Size: 10
Total Size required by OS: 2
Total Main Main Memory Size available for Process: 8
Total No. of Partions: 2
Main Memory State: 1, 2,
Internal Fragmentation: 1, 3,
External Fragmentation:
Start of External Fragmentation: -1
End of External Fragmentation: -1

...Program finished with exit code 0
Press ENTER to exit console.
```

```
input
Enter The Total Size of Main Memory: 12
Enter The Size of Memory required by OS: 2
Enter The No of Partitions of Main Memory: 2

-----
Enter The Process ID: 1
Enter The Size of Process: 4
Do you want to continue? 0 for NO & 1 for YES
1
Enter The Process ID: 2
Enter The Size of Process: 6
The Size of process is more than the Partition Size, so Memeoory can't be allocated to The Process!!!
Do you want to continue? 0 for NO & 1 for YES
0

-----
Total Main Main Memory Size: 12
Total Size required by OS: 2
Total Main Main Memory Size available for Process: 10
Total No. of Partions: 2
Main Memory State: 1, 0,
Internal Fragmentation: 1, -1,
External Fragmentation:
Start of External Fragmentation: 5
End of External Fragmentation: 9

...Program finished with exit code 0
Press ENTER to exit console.
```