



**VIT<sup>®</sup>**  
**Vellore Institute of Technology**  
(Deemed to be University under section 3 of UGC Act, 1956)

**OPERATING SYSTEMS LAB 18.05.2021**

**MEMORY MANAGEMENT**

**-V.MAHEYSH**

**19BCE1120**

**Aim:**

To apply various memory management algorithms.

## INPUT CONSIDERED:

### Memory Partition

| Partition   | Memory Size Available |
|-------------|-----------------------|
| Partition 1 | 100 KB                |
| Partition 2 | 500 KB                |
| Partition 3 | 200 KB                |
| Partition 4 | 300 KB                |
| Partition 5 | 600 KB                |

### Process Requires

| Process   | Memory Required |
|-----------|-----------------|
| Process 1 | 212 KB          |
| Process 2 | 417 KB          |
| Process 3 | 112 KB          |
| Process 4 | 426 KB          |

## FIRST FIT ALGORITHM:

```
1 #include<stdio.h>
2
3
4 void FirstFit(int bs[50],int m,int ps[50],int n)
5 {
6     int ab[50];
7     for(int i=0;i<n;i++)
8     {
9
10         ab[i]=-1;
11         for(int j=0;j<m;j++)
12         {
13
14             if(ps[i]≤bs[j])
15             {
16                 bs[j]-=ps[i];
17                 ab[i]=j;
18                 break;
19             }
20         }
21     }
22     printf("\nMemory block assigned:-\n ");
23     printf("    Process\tSize\tBlock\n");
24     for(int i=0;i<n;i++)
25     {
26
27         printf("\t%d\t%d\t",i+1,ps[i]);
28         if(ab[i]≠-1)
29             printf("%d\n",ab[i]+1);
30         else
31             printf("%s\n","Not allocated");
32     }
33 }
34
35
36 int main()
37 {
38     int m,bs[50],n,ps[50];
39     printf("FIRST FIT ALGORITHM:\n");
40     printf("\nEnter no of blocks: ");
41     scanf("%d",&m);
42     printf("Enter each block size: ");
43     for(int i=0;i<m;i++)
44         scanf("%d",&bs[i]);
45     printf("Enter no of processes: ");
46
47     scanf("%d",&n);
48     printf("Enter each process size: ");
49     for(int i=0;i<n;i++)
50         scanf("%d",&ps[i]);
51     FirstFit(bs,m,ps,n);
52 }
```

## OUTPUT:

```
root@kali:~/maheysh/18_05_21LAB# gcc FIRSTFIT_ALGO.c
root@kali:~/maheysh/18_05_21LAB# ./a.out
FIRST FIT ALGORITHM:
```

```
Enter no of blocks: 5
Enter each block size: 100 500 200 300 600
Enter no of processes: 4
Enter each process size: 212 417 112 426
```

Memory block assigned:-

| Process | Size | Block         |
|---------|------|---------------|
| 1       | 212  | 2             |
| 2       | 417  | 5             |
| 3       | 112  | 2             |
| 4       | 426  | Not allocated |

## NEXT FIT ALGORITHM:

```
1 #include<stdio.h>
2
3
4 void NextFit(int bs[50],int m,int ps[50],int n)
5 {
6     int ab[50],c,j;
7     for(int i=0;i<n;i++)
8     {
9
10         ab[i]=-1;
11         c=0;
12         j=0;
13         while(c<m)
14         {
15
16             if(ps[i] ≤ bs[j])
17             {
18                 bs[j]-=ps[i];
19                 ab[i]=j;
20                 break;
21             }
22             j=(j+1)%m;
23             c++;
24         }
25     }
26     printf("\nMemory block assigned:-\n ");
27     printf("    Process\tSize\tBlock\n");
28     for(int i=0;i<n;i++)
29     {
30
31         printf("\t%d\t%d\t",i+1,ps[i]);
32         if(ab[i]≠-1)
33             printf("%d\n",ab[i]+1);
34         else
35             printf("%s\n","Not allocated");
36     }
37 }
38 }
39
40 int main()
41 {
42     int m,bs[50],n,ps[50];
43     printf("NEXT FIT ALGORITHM:\n");
44     printf("\nEnter no of blocks: ");
45     scanf("%d",&m);
```

```

46     printf("Enter each block size: ");
47     for(int i=0;i<m;i++)
48         scanf("%d",&bs[i]);
49     printf("Enter no of processes: ");
50     scanf("%d",&n);
51     printf("Enter each process size: ");
52     for(int i=0;i<n;i++)
53         scanf("%d",&ps[i]);
54     NextFit(bs,m,ps,n);
55 }
56

```

## OUTPUT:

```

root@kali:~/maheysh/18_05_21LAB# gcc NEXTFIT_ALGO.c
root@kali:~/maheysh/18_05_21LAB# ./a.out
NEXT FIT ALGORITHM:

Enter no of blocks: 5
Enter each block size: 100 500 200 300 600
Enter no of processes: 4
Enter each process size: 212 417 112 426

Memory block assigned:-
  Process   Size   Block
    1       212     2
    2       417     5
    3       112     2
    4       426  Not allocated
root@kali:~/maheysh/18_05_21LAB#

```

## BEST FIT ALGORITHM:

```
1 #include<stdio.h>
2
3
4 void BestFit(int bs[50],int m,int ps[50],int n)
5 {
6     int ab[50],a;
7     for(int i=0;i<n;i++)
8     {
9
10         ab[i]=-1;
11         for(int j=0;j<m;j++)
12             if(ps[i] ≤ bs[j])
13             {
14                 a=bs[j]-ps[i];
15                 if(ab[i]==-1)
16                     ab[i]=j;
17                 else if(bs[ab[i]]-ps[i]>a)
18                     ab[i]=j;
19             }
20         if(ab[i]≠-1)
21             bs[ab[i]]-=ps[i];
22     }
23     printf("\nMemory block assigned:-\n ");
24     printf("    Process\tSize\tBlock\n");
25     for(int i=0;i<n;i++)
26     {
27
28         printf("\t%d\t%d\t",i+1,ps[i]);
29         if(ab[i]≠-1)
30             printf("%d\n",ab[i]+1);
31         else
32             printf("%s\n","Not allocated");
33     }
34 }
35 }
36
37 int main()
38 {
39     int m,bs[50],n,ps[50];
40     printf("BEST FIT ALGORITHM:\n");
41     printf("\nEnter no of blocks: ");
42     scanf("%d",&m);
43     printf("Enter each block size: ");
44     for(int i=0;i<m;i++)
45         scanf("%d",&bs[i]);
46
47     printf("Enter no of processes: ");
48     scanf("%d",&n);
49     printf("Enter each process size: ");
50     for(int i=0;i<n;i++)
51         scanf("%d",&ps[i]);
52     BestFit(bs,m,ps,n);
53 }
```

## OUTPUT:

```
root@kali:~/maheysh/18_05_21LAB# gcc BESTFIT_ALGO.c
root@kali:~/maheysh/18_05_21LAB# ./a.out
BEST FIT ALGORITHM:

Enter no of blocks: 5
Enter each block size: 100 500 200 300 600
Enter no of processes: 4
Enter each process size: 212 417 112 426

Memory block assigned:-
  Process      Size      Block
    1           212         4
    2           417         2
    3           112         3
    4           426         5
root@kali:~/maheysh/18_05_21LAB#
```

## WORST FIT ALGORITHM:

```
1 #include<stdio.h>
2
3
4 void WorstFit(int bs[50],int m,int ps[50],int n)
5 {
6     int ab[50],a;
7     for(int i=0;i<n;i++)
8     {
9
10         ab[i]=-1;
11         for(int j=0;j<m;j++)
12         {
13
14             if(ps[i]≤bs[j])
15             {
16                 a=bs[j]-ps[i];
17                 if(ab[i]==-1)
18                     ab[i]=j;
19                 else
20                 {
21                     if(bs[ab[i]]-ps[i]<a)
22                     {
23                         ab[i]=j;
24                     }
25                 }
26             }
27         }
28         if(ab[i]≠-1)
29             bs[ab[i]]-=ps[i];
30     }
31     printf("\nMemory block assigned:-\n ");
32     printf("    Process\tSize\tBlock\n");
33     for(int i=0;i<n;i++)
34     {
35
36         printf("\t%d\t%d\t",i+1,ps[i]);
37         if(ab[i]≠-1)
38             printf("\t%d\n",ab[i]+1);
39         else
40             printf("\t%s\n","Not allocated");
41     }
42 }
43 }
44
```

```

45 int main()
46 {
47     int m,bs[50],n,ps[50];
48     printf("WORST FIT ALGORITHM:\n");
49     printf("\nEnter no of blocks: ");
50     scanf("%d",&m);
51     printf("Enter each block size: ");
52     for(int i=0;i<m;i++)
53         scanf("%d",&bs[i]);
54     printf("Enter no of processes: ");
55     scanf("%d",&n);
56     printf("Enter each process size: ");
57     for(int i=0;i<n;i++)
58         scanf("%d",&ps[i]);
59     WorstFit(bs,m,ps,n);
60 }
61

```

## OUTPUT:

```

root@kali:~/maheysh/18_05_21LAB# gcc WORSTFIT_ALGO.c
root@kali:~/maheysh/18_05_21LAB# ./a.out
WORST FIT ALGORITHM:

Enter no of blocks: 5
Enter each block size: 100 500 200 300 600
Enter no of processes: 4
Enter each process size: 212 417 112 426

Memory block assigned:-
    Process    Size    Block
        1      212      5
        2      417      2
        3      112      5
        4      426    Not allocated
root@kali:~/maheysh/18_05_21LAB#

```

## Result:

Various memory management algorithms were applied. Best fit algorithm was found out to be the most efficient.