



DEPARTMENT OF
COMPUTER SCIENCE AND ENGINEERING

Title: Contiguous Memory Allocation Techniques

OPERATING SYSTEM LAB
CSE 310



GREEN UNIVERSITY OF BANGLADESH

1 Objective(s)

- To gather knowledge of different types of contiguous memory allocation techniques.
- To implement worst fit, first fit, best fit contiguous memory allocation technique.

2 Problem analysis

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, a memory section large enough for this process is provided. When it is time to load or swap a process into main memory, and if there is more than one free block of memory of sufficient size, then the operating system must decide which free block to allocate.

2.1 Worst Fit Contiguous Memory Allocation Technique

This method requires to scan the entire list of free blocks. The largest block among the free blocks is selected. This strategy leaves the largest fragment which may be useful for the other process.

3 Implementation in C

```
1  #include<stdio.h>
2
3  #define max 25
4
5  void main()
6  {
7      int frag[max],b[max],f[max],i,j,nb,nf,temp,highest=0;
8      int bf[max],ff[max];
9
10     for(i=0; i<max; i++){
11         b[i] = 0;
12         f[i] = 0;
13         frag[i] = 0;
14         bf[i] = 0;
15         ff[i] = 0;
16     }
17
18     printf("\nEnter the number of blocks:");
19     scanf("%d",&nb);
20
21     printf("Enter the number of files:");
22     scanf("%d",&nf);
23
24     printf("\nEnter the size of the blocks:-\n");
25     for(i=1; i<=nb; i++)
26     {
27         printf("Block %d:",i);
28         scanf("%d",&b[i]);
29     }
30
31     printf("Enter the size of the files:-\n");
32     for(i=1; i<=nf; i++)
33     {
```

```

34     printf("File %d:",i);
35     scanf("%d",&f[i]);
36 }
37
38 for(i=1; i<=nf; i++)
39 {
40     for(j=1; j<=nb; j++)
41     {
42         if(bf[j] !=1)
43         {
44             temp=b[j]-f[i];
45             if(temp>=0)
46                 if(highest<temp)
47                 {
48                     ff[i]=j;
49                     highest=temp;
50                 }
51         }
52     }
53     frag[i]=highest;
54     bf[ff[i]]=1;
55     highest=0;
56 }
57
58 printf("\nFile_no   \tFile_size   \tBlock_no   \tBlock_size   \tFragment");
59 for(i=1; i<=nf; i++)
60     printf("\n%d\t\t%d\t\t%d\t\t%d\t\t%d",i,f[i],ff[i],b[ff[i]],frag[i]);
61
62 }

```

4 Input/Output

4.1 Input

Input of the program is given below.

```

Enter the number of blocks: 4
Enter the number of files: 3

Enter the size of the blocks:-
Block 1: 5
Block 2: 8
Block 3: 4
Block 4: 10
Enter the size of the files:-
File 1: 1
File 2: 4
File 3: 7

```

4.2 Output

Output of the program is given below.

File_no	File_size	Block_no	Block_size	Fragment
1	1	4	10	9
2	4	2	8	4
3	7	0	0	0

5 Discussion & Conclusion

Based on the focused objective(s) to understand about worst fit contiguous memory allocation algorithm, the additional lab exercise made me more confident towards the fulfilment of the objectives(s).

6 Lab Task (Please implement yourself and show the output to the instructor)

1. Implement best fit contiguous memory allocation algorithm.

6.1 Problem analysis

This method needs the list of free blocks to be sorted according to their size. Then the smallest block that is large enough for the process to accommodate is selected from the list of free blocks. This strategy reduces the wastage of memory as it does not allocate a block of larger size which leaves some amount of memory even after the process accommodates the space.

6.2 Input

Input of the program is given below.

```
Enter the number of blocks: 4
Enter the number of files: 3

Enter the size of the blocks:-
Block 1: 5
Block 2: 8
Block 3: 4
Block 4: 10
Enter the size of the files:-
File 1: 1
File 2: 4
File 3: 7
```

6.3 Output

Output of the program is given below.

File_no	File_size	Block_no	Block_size	Fragment
1	1	3	4	3
2	4	1	5	1
3	7	2	8	1

7 Lab Exercise (Submit as a report)

- Implement first fit contiguous memory allocation algorithm.

7.1 Input

Input of the program is given below.

```
Enter the number of blocks: 4
Enter the number of files: 3

Enter the size of the blocks:-
Block 1: 5
Block 2: 8
Block 3: 4
Block 4: 10
Enter the size of the files:-
File 1: 1
File 2: 4
File 3: 7
```

7.2 Output

Output of the program is given below.

File_no:	File_size :	Block_no:	Block_size:	Fragment
1	1	1	5	4
2	4	2	8	4
3	7	4	10	3

8 Policy

Copying from internet, classmate, seniors, or from any other source is strongly prohibited. 100% marks will be *deducted* if any such copying is detected.