****

**Green University of Bangladesh**

**Department of Computer Science and Engineering (CSE)**

**Semester: (Spring, Year:2024), B.Sc. in CSE (Day)**

**Lab Report NO #04**

**Course Title:** Operating System Lab

**Course Code:** CSE 310  **Section:** 221 D6

**Lab Experiment Name:** Contiguous Memory Allocation Techniques.

**Student Details**

| **Name** | | **ID** |
| --- | --- | --- |
| **1.** | Tanvir Ahmed | 221002461 |

**Lab Date : 6/5/2024**

**Submission Date : 13/5/2024**

**Course Teacher’s Name : Jarin Tasnim Tonvi**

| **Lab Report Status**  **Marks: ………………………………… Signature:.....................**  **Comments:.............................................. Date:..............................** |
| --- |

**1. TITLE OF THE LAB REPORT EXPERIMENT**

Contiguous memory allocation techniques.

**2. OBJECTIVES/AIM**

* To gather knowledge of contiguous memory allocation techniques.
* To implement the best fit contiguous memory allocation technique.

**3. PROCEDURE**

* First, take the number of blocks and files from the user.
* Take the size of blocks and files.
* Now run two loops, the outer loop for files and the inner loop for blocks.
* In the loops first check if the block is already used or not.
* If the block is not used, find the block with the lowest fragmentation.
* Assign the fragmentation to an array and set the block as used.
* Finally, print all the information.

**4. IMPLEMENTATION**

Source Code:

1. #**include**<stdio.h>
2. #**include** <limits.h>
3. void **main**()
4. {
5. int frag[100],b[100],f[100],i,j,nb,nf,temp,lowest=INT\_MAX,bf[100],ff[100];
6. printf("\nEnter the number of blocks: ");
7. scanf("%d",&nb);
8. printf("Enter the number of files: ");
9. scanf("%d",&nf);
10. printf("\nEnter the size of the blocks: \n");
11. **for**(i=1; i<=nb; i++)
12. {
13. printf("Block %d:",i);
14. scanf("%d",&b[i]);
15. }
16. printf("Enter the size of the files: \n");
17. **for**(i=1; i<=nf; i++)
18. {
19. printf("File %d:",i);
20. scanf("%d",&f[i]);
21. }
22. **for**(i=1; i<=nf; i++)
23. {
24. **for**(j=1; j<=nb; j++)
25. {
26. **if**(bf[j]!=1)
27. {
28. temp=b[j]-f[i];
29. **if**(temp>=0)
30. **if**(lowest>temp)
31. {
32. ff[i]=j;
33. lowest=temp;
34. }
35. }
36. }
37. frag[i]=lowest;
38. bf[ff[i]]=1;
39. lowest=INT\_MAX;
40. }
41. printf("\nFile\_no \tFile\_size \tBlock\_no \tBlock\_size \tFragment");
42. **for**(i=1; i<=nf; i++)
43. 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]);
44. }

**5. TEST RESULT / OUTPUT**

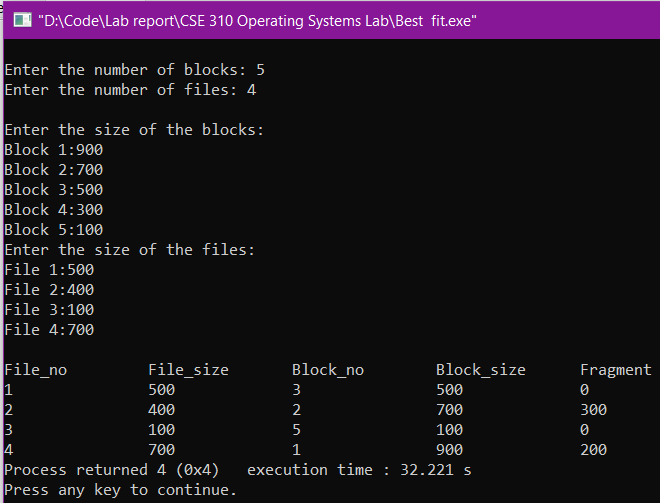
****

Fig1. Sample input and output for the problem.

**6. ANALYSIS AND DISCUSSION**

* In this exercise, I implemented the best fit contiguous memory allocation algorithm.
* In the previous lab class, I learned and implemented the worst fit, best fit, and first fit contiguous memory allocation algorithm.
* I have the experience to implement this same task (best fit algorithm) in the previous class and I just implemented that same code here in this exercise.
* First, I wrote the code for the worst fit algorithm that I found from the lab menial then I modified it to the best fit algorithm by changing some conditions. That is, I convert the highest fragmentation value to the lowest in the condition checking based on the best fit algorithm criteria.
* Finally, I succeeded in implementing both the worst fit and best fit algorithms.

**7. SUMMARY**

This lab exercise demonstrates the successful implementation of contiguous memory allocation techniques based on the knowledge that I learned from the previous lab class. In the previous class, I already implemented the worst fit, best fit, and first fit algorithms. I just repeated the worst fit first and then modified it to the best fit algorithm. I merged all of them in this exercise and solved the problem. I completed the problem without any major difficulties, I faced a syntax error and I was able to solve this instantly.