OS MINI-PROJECT

FILE ALLOCATION

METHODS

MOHIT KURSIJA 75

PAYAL KADAM 64

**TITLE:**To implement different file allocation techniques using GUI.

**DESCRIPTION:**

This project is made using JAVA Swing.Swing is a GUI widget toolkit for [Java](https://en.wikipedia.org/wiki/Java_(programming_language)). It is an [API](https://en.wikipedia.org/wiki/Application_programming_interface) for providing a [graphical user interface](https://en.wikipedia.org/wiki/Graphical_user_interface) (GUI) for Java programs.

In this project, we basically represent the allocation of files in disk space by using different allocation methods.The allocation methods define how the files are stored in the disk blocks. Two allocation techniques as discussed below are represented.

The allocationis shown in code with the use of array for disk space. The logic is different for different methods, according to which the block-related information is stored in this array. The allocation is then represented by changing the colours of blocks.

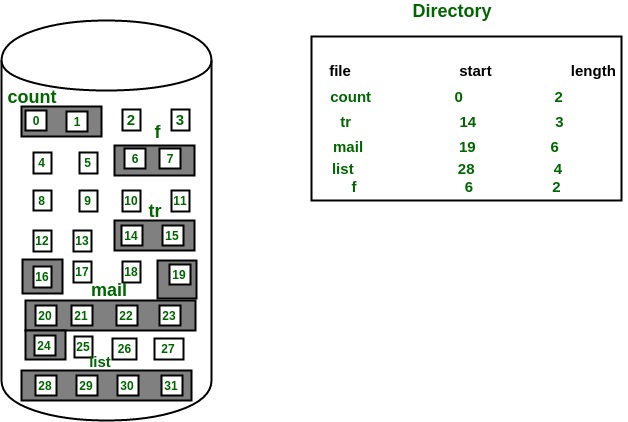
**File Allocation Methods:**

**1. Contiguous Allocation:**

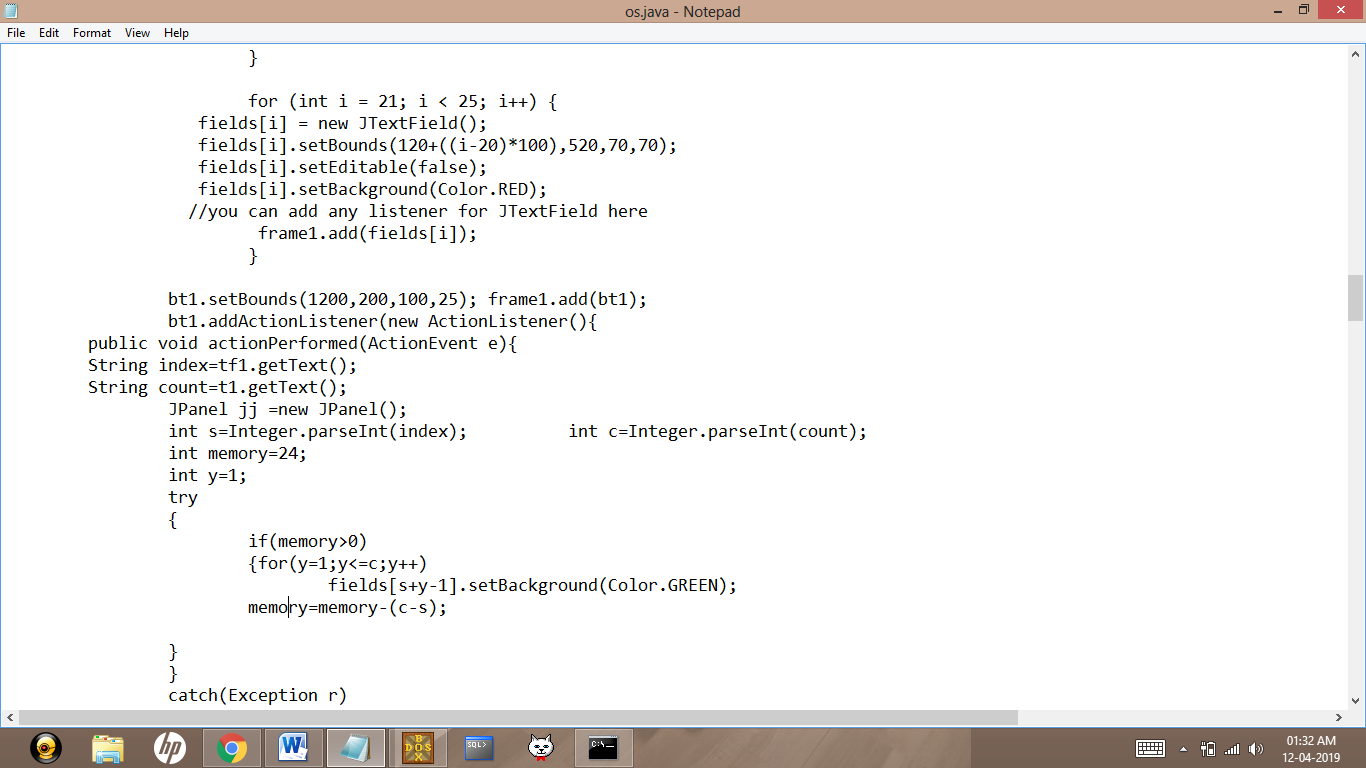
In this scheme, each file occupies a contiguous set of blocks on the disk. For example, if a file requires n blocks and is given a block b as the starting location, then the blocks assigned to the file will be:*b, b+1, b+2,……b+n-1.* This means that given the starting block address and the length of the file (in terms of blocks required), we can determine the blocks occupied by the file.  
The directory entry for a file with contiguous allocation contains

* Address of starting block
* Length of the allocated portion.

The*file ‘mail’* in the following figure starts from the block 19 with length = 6 blocks. Therefore, it occupies *19, 20, 21, 22, 23, 24* blocks.



In the code, it takes the start block and length of file for allocation from the user, selects the blocks to be allocated and shows the result.



**2.Linked List Allocation Using an Index:**

We can eliminate the problem of wasting space by having to store a pointer and the problem of not having random access by taking the pointer from each block and storing them in an index or table and placing this in memory. Below is a diagram showing how the files (A and B) would be represented using a linked list allocation with an index.

|  |
| --- |
| http://www.cs.nott.ac.uk/~pszgxk/courses/g53ops/images/linkedlistindex.jpg |

If we consider file B, it occupies blocks 11, 2, 14 and 8, but as the pointers are in memory random access is much faster as a given offset can be located by using only memory accesses until the correct block has been reached.

In the code, it takes the pointer information that each block needs to store, from the user, selects the blocks to be allocated and shows the result.