# File System Due , 18-Nov

In this assignment, you are to implement a basic file system using a File Allocation Table.

### Data Structures

Organize you disk partition into 10 blocks with 10 characters per block. You only need to store String data in this file system. You will need to design a file table to store file information and a file allocation table to keep track of which blocks (if any) belong to which files and to track free blocks.

### Methods

int createFile(String inFileName, boolean readOnly)  
Creates an entry in the file table with the given file name and opens a file. If the file is read-only then the readOnly field is true.  
This method returns a file handle if successful or -1 if not (e.g., if there is no more room in the file table).

int deleteFile(String inFileName)  
Frees all blocks assigned to the file and removes the entry from the file table.  
Returns 1 if successful, 0 if not (for example, the file was not in the file table).

String readFile(int fileHandle)  
Returns the data in the file for the given file handle as one continuous String (even if the file occupies more than one disk block).

int writeFile(int fileHandle, String data)  
Writes the data to the file with the given fileHandle.  
Returns 1 if the write was successful or 0 if not (e.g., there were not enough free blocks to hold the file or if the file is read-only).

### Test Scenario

The program FileSystemTest.java is available on MUOnline. It implements the following test scenario:

1. Create file 1 and write a string of 25 characters to file 1 (3 blocks used)
2. Create file 2 and write a string of 45 characters to file 2 (5 blocks for a total of 8). Note that block allocations may not be contiguous after this step.
3. Read file 1 and display it on the console
4. Read file 2 and display it on the console
5. Delete file 1 (5 blocks now in use)
6. Create file 3 and write a string of 35 characters to file 3 (4 blocks for a total of 9)
7. Read file 3 and display it on the console
8. Delete file 2 (4 blocks now in use)
9. Create file 4 and write string of 18 characters to file 4 (2 blocks for a total of 6)
10. Read file 4; the remnants of prior strings should not appear.
11. Create file 5 and write a string of 55 characters (6 blocks are required so this should fail)
12. Create file 6 and write a string of 32 characters (4 blocks are required for a total of 10)
13. Read file 6

### Program Design

I strongly urge you to create a program design before diving into the code. This design can be in whatever form suits you best: pseudo-code, flowchart, UML diagram, etc.

### Submitting Your Assignment

Upload your report and your Java source in MUOnline. Feel free to talk with classmates about coding methods and techniques however, the code you submit must be written by you.