Design and Implementation of an In-memory File System

It is required to design and implement a UNIX-like file system. Such a file system typically consists of a sequence of blocks, organized in four different regions: boot block, super block, inode list, and data blocks. Although a file system is usually implemented on some secondary storage medium like disk, CDROM or flash memory, in this assignment it is required to implement the same "in-memory". A block of main memory will be set aside and considered as the medium where the file system is going to reside.

The parameters of the in-memory file system will be as follows:

- The total size of the file system will be 16 Mbytes. Space for the file system can be allocated using the "malloc" system call, and accessed for implementing all the various subsystems.
- The block size will be 1024 bytes. In other words, the total number of blocks will be 16K. The block address is represented as a 16-bit integer.
- The total number of files can be 128; that is, there will be 128 entries in the inode list.
- In the inode, the file data can be kept track of in the usual way using 10 direct pointers, 1 singly indirect, 1 double indirect, and 1 triple indirect pointer. Of course, in this case the size of the file system is small and the indirect pointers will not be used. Assume that the maximum size of a file is 10 Kbytes.

Some of the parts of the assignment has to be carried out individually, while some parts will require discussion and collaboration. One objective of this assignment is also to promote group activity.

6. Write functions for the following:

    a. Merge two files and store it as a new file in the in-memory file system.

    b. Open, close, read and write functions that can be called from a C program.

    c. Write a C program to read data from a file in the in-memory file system.

    d. Write a C program to create a new file in the in-memory file system and write some data into it.

Hint: The following points may be noted:

- Write a function that will take the inodes of the two given files and that of the new file as inputs, and copy the files one after the other block-by-block.
- Write C library functions “open”, “close”, “read”, “write” using which you can write a C program that uses the in-memory file system. You can choose the parameters of these functions as per your convenience.
- Using the library functions as implemented, write a C program to read the contents of a given file block-by-block, and display it on the screen.
- Using the library functions as implemented, write a C program to create a new file, and write few lines of text into it.