I will try to explain what is behind the construction of the system - I hope I was correct in my understanding

First of all, hours of reading materials on the Internet and in books and thinking deeply about how to plan and implement such a system

**The system is built like this:**

In the virtualFileSystem file that is saved on the hard disk, I saved a binary tree type data structure in a binary file that contains objects of the file information

At the beginning of use - the program reads from the memory into the tree structure all the data of the files that were saved in the file - then all operations that require reading or writing and updating the files are done only through the tree structure

In this way I saved all the runs to memory for reading files that have already been read once

and also all the runs to memory for writing to files except for the final writing

(That is, if a user wrote 3 times to a certain file, only when the close method is called, the data will be saved and updated)

**In this way I saved many runs for the hard disk memory and the ram memory**

Of course I planned and implemented all the additional requirements

It was really interesting and challenging to understand the construction of the system but the implementation was quite enjoyable

**Assumptions I made during the writing:**

The vfs system works according to the operating system, so double use of the system is not possible

The files saved on the computer are text files only