# IT-308   Project Report.

Topic : Virtual -File-System.

Features: Fast System-calls.
Journaling.
In-memory file systems.

**Team members**: Harshil ,201301410
Samarth,201301456

## Project Description:

Our File-System is divided in to two layers :

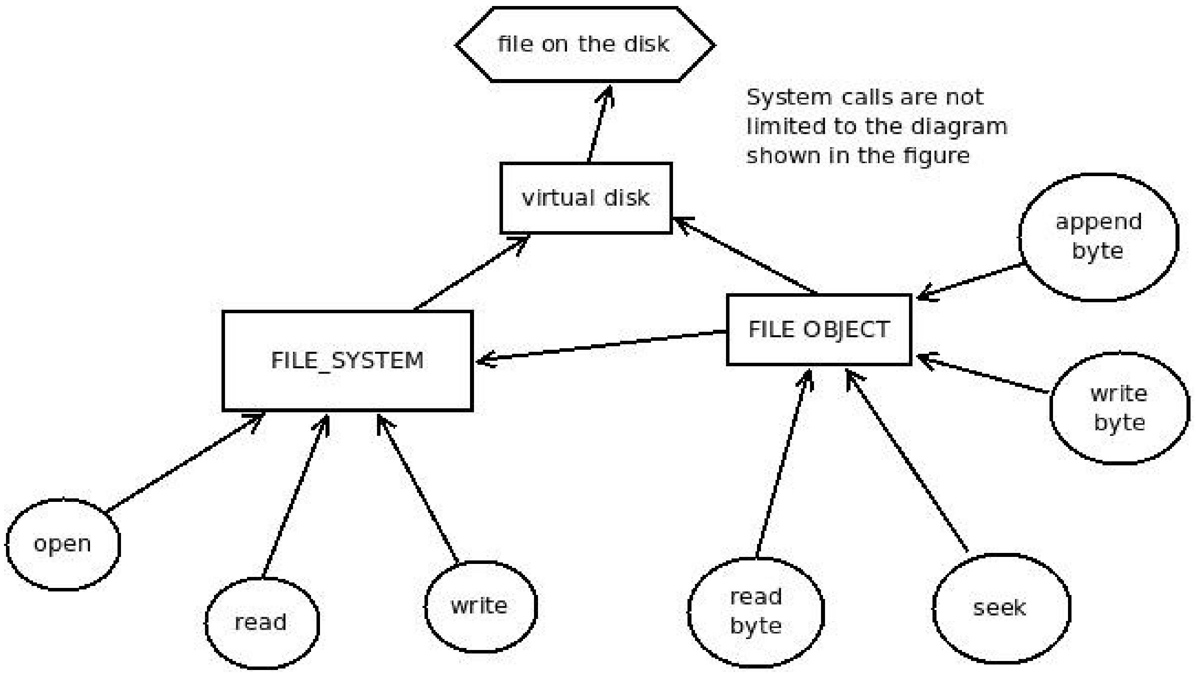
**1) Layer 1**

It contains file-system calls like open,file,read...etc.

**2)Layer 2**

It contains calls that access the virtual-disk(or file)

**Diagram:**



We have designed our file system in an object oriented way So there are three objects FILE_SYSTEM,FILE_OBJECT, VIRTUAL DISK

- The size of virtual disk is 20MB.
- The virtual disk is divided in to two files "FS.data" and "FS.metadata". FS.data contains the actual blocks that are allocated to the files. FS.metadata contains information about the  disk like disk size, blocksize,no of blocks etc.To see how the meta-data has been organized FS_template.metadata .
- FILE_SYSTEM has a FAT table which keeps track of non-allocated blocks that are not used by any file.
- Allocated block list for individual files is maintained by the FILE object itself.

Our File-System support following system calls.

- create() , to create a file.
- open()  , to open the created file (it will return file-descriptor on successful call).
- close(), to close the file.
- read(), to read form the file.
- write(), to write to the file.
- append(), to append to the file.
- seek(), to seek a particular position in the file.
- delete(), to delete a file

- rename(), to rename a file.
- size() ,to get size of the file.
- fileinfo(), to print whole information of the file.
- normalexit(), to exit normally.

**Working:**

First the blocks on the disk is read in to the main memory in the ram if the file system is large then then paging can be done to support “virtual block managing”.In this case the size of the file-system is 20MB, so it can be safely accommodated in to the memory.
so first as the file system is started the disk is loaded in to memory for fast access. During virtual paging if there is block which is not in the main memory then the block is loaded back in to main memory and some other block is written back in to the disk.

working of the system calls is shown in the code in comments.

**Future Scope:**
The file system is loaded in to memory so it will not be persistent when interrupted during working so a log is written which contains the block that need to be written with the modified value. So if a file system is crash it can recover from

the crash when it starts again. Future scope of this project is to make the journaling system better resistant to crash. Also the journaling of meta data should  be included.

**Application:**

The file system is very robust as it can create 100000 files in 482 ms. So it can be used in file-systems where speed is a great concern.