

Virtual File System

Summary:

Assume that you have a virtual file system with a root directory called "root" all the files and folders will be stored under it. The disk size consists of **N** blocks and each block size is **1 KB**.

This program simulates the **allocation** and **de-allocation** of files and folders using different allocation techniques.

1. Contiguous Allocation (Using worst Fit allocation)
2. Linked Allocation
3. Indexed Allocation

After running the application, the user can interact with your virtual file system through a series of commands illustrated in the table below:

Commands:

Command	Summary
CreateFile root/file.txt 100	This command used to create a file named "file.txt" with 100 KB size under the path "root" Prerequisites: 1. Path already exists 2. No file with the same name is already created under this path 3. Enough space exists
CreateFolder root/folder1	This command is used to create a new folder named "folder1" under the path "root". Prerequisites: 1. Path already exists 2. No folder with the same name is already created under this path.
DeleteFile root/folder1/file.txt	This command is used to delete a file named "file.txt" from the path "root/folder1". Any blocks allocated by this file should be de-allocated. Prerequisites: 1. The file already exists under the path specified
DeleteFolder root/folder1	This command is used to delete a folder named "folder1" from the path "root". All files and subdirectories of this folder will also be deleted. Prerequisites: 1. The folder already exists under the path specified

DisplayDiskStatus	This command is used to display the status of your Driver the status should contain the following information: 1. Empty space 2. Allocated space 3. Empty Blocks in the Disk 4. Allocated Blocks in the Disk
DisplayDiskStructure	This command will display the files and folders in your system file in a tree structure.

Saving and loading Virtual File System:

In this program I'm not creating actual files and folder, just **simulating** having a **series of blocks** and these blocks will be allocated to files when created and will be deallocated when these files are deleted.

The allocated blocks are saved on a file on the hard disk to be able to load it in the next time you run the application. So, when the application starts, the system automatically loads the disk structure form the **Virtual File System file** say named "c:\Ctgs.vfs". Then the user will start to enter commands which will be executed on the loaded data in memory, and before the application terminates, the data in memory is written into the file again.

The following information is stored in Ctgs.vfs file:

1. Files and Folders Directory Structure.
2. The Empty blocks of the virtual DISK.
3. The allocated blocks in your virtual DISK and which files/folders are take these places.

Empty Blocks (Free Space Manager):

The free space manager component is used to **know which blocks are free**, to **search** for free contiguous blocks, to **de-allocate blocks** when a file is deleted and to **allocate blocks** when a file is created and so on.