CSS430 Prog 5 Report
Alex Xiong
Elygh Thao

Specification on assumptions and limitations

Due to incorrect execution, most of our files cannot be considered functionally correct. Thus, our program is limited and cannot be fully operational. Directory, inode, superblock and other file system files should follow assignment instructions to the best of their abilities.

Internal Design

Directory:

- Directory object is meant to be the entry point for accessing the multiple inodes associated with the filesystem object.
- Directory object directly interacts with the disk using a string referring to the filename to allocate, deallocate, and read bytes to help keep track of which inode numbers are in use and which ones are available to use
- Has a useful search function to help find where a filename and its associated inumber

File Table:

- File Table object is used by the File System to handle the user thread commands
- File Table uses File Table Entry class to create entries it its table
- File Table makes requests to allocate information, find information, or deallocate information in the disk using the private directory object it instantiates
- File Table keeps track of file table entry's being accessed in a vector to remember which files have been added
- File Table overall finds inodes and returns the FileTableEntry
- Returns a request (boolean) to remove a FileTableEntry
- Returns a whether or not (boolean) a table is empty

File Table Entry:

- Used by File Table to keep hold data referencing an Inode's inumber and associated String, used to make the File Table object keep better track of which inodes are being added/removed

Inode:

- Inodes describes a file in the disk by means of pointers
- Inodes are kept track of in the single Directory object instantiated from the File System class
- All of their variables except iNodeSize and directSize are public so that they can be accessed from other objects such as File System or Directory

Super Block:

- Is used only on disk block 0

- Very basic block used to instantiate the number of disk blocks, number of inodes, and the block number of the head block of the free list
- Super block used in File System object to help instantiate the directory and inode system it practices

File System:

- Instantiates a superblock, directory, and filetable to keep track of file system
- Superblock is used to initialize number of inodes
- Directory is used to register root in directory entry 0
- File table is created, and used to store directory in the file table
- File system acts as the manager system to access and write data in the form of FileTableEntry objects being fed in as input in addition to byte arrays
- FileSystem() constructor initializes everything
- sync() syncs the current superblock with the correct strings by using open(), write(), and close()
- format() formats the superblock creating a free list, new directory, and new filetable
- open() attempts to return a new table entry of the strings passed in
- close() decrements the count of the file table entry as well as returns true or false whether or not its free
- fsize() casts synchronize to get size of file in bytes and returns length of inode length
- read() reads disk until eof file is reached, once finding target block, read the data by copying its information into a buffer and return the offset of amount that buffer places on the disk
- write() write to the disk by first synchronizing, then finding the target block. This is done by an inode method. After finding targetblock, if it's -1 or -2 or -3, return -1, method ends. If the target is not any of these numbers, proceed with write(). Find where to write (offset) and use a buffer to write the data into the block. Before ending write it to the disk and return the number of bytes written
- deallocAllBlocks() deallocs all of the blocks by looping through each inode and calling the returnBlock method with ftEnt's direct[i] as well as setting ftEnt.inode.direct[] to -1. Before ending, write to disk and return true, this method will only return false if there is only one block or ftEnt value passed in is null.
- seek() method updates seek pointer for the filetable entry object passed in depending on the block location

CSS430 Prog 5 Report
Alex Xiong
Elygh Thao

Execution performance and functionality

Incorrect Output

When we tried running our code, we got a SysLib error. It was looking for a format method that was not defined in the SysLib.java file provided.