CS3008 Assignment
Martin Minovski, ID: 51121641

REPORT

1. **The submission consists of the source files filesys.h, filesys.c and shell.c, Makefile and submission virtual disk exports for CAS 20.**
2. **All steps for achieving CAS 20 have been followed in shell file except for threading and concurrency. Internal copy/move and import/export to/from real HDD have been implemented.**
3. **Source code and functions have been documented with comments.**
4. **Functions from API explained:**
    1. void format()
        1. Sets all bytes to NIL
        2. Prepares reserved block, FAT table and root dir.
    2. MyFILE * myfopen (const char * filename, const car * mode) ;
        1. Calls a chain of functions for each step of the process. First tokenizes the path variable, then checks if file exists. If it doesn't exist and mode is “w”, it creates the file automatically and returns the stream pointer. File creation explained:
            1. Find new free block in FAT
            2. Create new directory entry and fill in the name
            3. Set all flags (unused, file length, isdir, etc) to 0
            4. Update FAT and virtual disk
    3. void myfclose (MyFILE * stream)
        1. Writes out any pending bytes (in case of mode=”w”)
        2. Writes the modified time for the file (in case of mode=”w”)
        3. Copies the FAT table into the virtual disk FAT block
        4. Frees the memory allocation for the stream.
    4. int myfgetc (MyFILE * stream)
        1. Returns EOF if end of file is reached. EOF detection achieved by using file length and current position. Therefore can store binary files.
        2. Changes the block if the position is on last byte of block, but there is more data to read from other blocks in the FAT chain for the file entry.
    5. void myfputc (int b, MyFILE * stream)
        1. Checks if stream mode is “w”, otherwise throws error and returns.
        2. Shifts stream to next block in FAT chain, if required.
        3. Writes byte to stream
        4. Updates position and file length of stream
    6. void mymkdir (const char * path)
        1. Tokenizes the path variable if required
        2. While tokenizing (using int pathaction(const char * path, int action)), creates all directories stated in path variable. Creation of a folder explained:
            1. Find new free block in FAT table
            2. Create a new dir entry in current directory
            3. Fill in the name, set isdir to 1
            4. Create self and parent links (./ and ../)
            5. Update FAT table and virtual disk
    7. void myrmdir (const char * path)
        1. Tokenizes the path.
        2. Checks if the path to the directory to be deleted (last token) exists and throws error if not.

3. Deletes the directory:
    1. Recursively checks for non-empty inner directories and calls the same function on them. Base case for recursion: no non-empty inner directories, returns.
    2. Treats empty directories as files. Calls removeEntry(char*name, int isdir) on them.
    3. Removes itself and changes currentDirIndex to parent directory.
8. void mychdir (const char * path)
    1. Tokenizes path.
    2. Changes directory for every token. If dir doesn't exist, throws error.
9. void myremove (const char * path)
    1. Tokenizes path and scans for the entry to remove. Throws error if not found.
    2. Calls removeEntry(char * name, int isdir)
        1. NOTE: Deleted directory entries (for files or dirs) persist – their name is set to "*DELETED*" and the "unused" flag is set to 1. That way it won't be considered by other functions. Blocks in FAT chain are emptied and set to UNUSED in FAT so that space can be used in future.
10. char ** mylistdir (const char * path)
    1. Creates an array of strings containing the contents of the curent directory

5. More interesting functions:

1. char * mylistdisk()
    1. Recursively lists all the directories and their contents in the virtual disk. Base case for recursion: no directories in the directory.
    2. Shifts dir block if necessary.
2. void myreadstringfile(char * name)
    1. Prints to console contents of a file as string using loop and file length.
3. void mywritestringfile(char * name,char*data)
    1. Creates new file and writes contents of *data to it.
4. void mycopytorealdisk (char * filename, char * destname)
    1. Creates a copy of a file to the real hard drive.
5. void mycopyfromrealdisk (char * filename)
    1. Reads a file from real hard drive and creates a copy in virtual disk.
6. void mycopyfile(char * pathsrc, char * pathdest)
    1. Tokenizes path and copies file in virtual disk.
7. void mymovefile(char * pathsrc, char * pathdest)
    1. Tokenizes path and moves file in virtual disk.
8. int pathaction(const char * path, int action)
    1. Tokenizes path and calls required action. Used by API functions requiring tokenizing. Action numbers:
        1. Change dir
        2. Make dir
        3. Remove dir
        4. Remove file
        5. Make dir and change dir for file open

6. How to use the file system (out of shell.c):

1. Import filesys.c
2. Use functions from API – first format, then the rest.