Project 4 Specification
FAT32 File System Utility Extension

Assigned: April 25, 2018
Due: May 9, 2018

Language Restrictions: C/C++, Python, Java (Note: C/C++ code MUST run on Ubuntu Linux 16.04)
Additional Restrictions: system() and exec*() system calls may not be used

Purpose

In Project 3, you wrote a FAT32 file system utility that could read data from a FAT32 file system, but could not change the FAT32 file system. In this project you will extend that work to create new files, write into them, and delete files.

Problem Statement

Your program will be a continuation of your code from Project 3 with the same teams. The amount of work requires is not that much, but you must understand free list management in FAT32, since creating files (which possibly means extending a directory), writing into files, and deleting files affects several lists within the FAT table.

Project Tasks

Here are the additional commands you are expected to implement

- `freelist` (do this one first)

  Description: print out the indexes of the first three free clusters and total number of free clusters on FAT structure (for your own edification and debugging).

- `newfile <FILE_NAME> <SIZE>`

  Description: In the current directory, create a new file with the given name (8+3 name will be specified. Also initialize it to the given size by assigning it clusters that are currently free on the FAT. Make sure the directory entry reflects the assigned size. The contents of your newly created file should be the string "New File.\r\n" repeated over and over. (\r\n stands for the Microsoft newline character sequence).

- `delete <FILE_NAME>`

  Description: mark the designated file as deleted in the current directory and all its contents as free in the FAT. `ls` should no longer list the file, though its directory entry should still be visible in hexedit, marked as deleted.

If `newfile` and `delete` work correctly, the changes should be checkable using `freelist`.

Allowed Assumptions (also see Assignment 3)

- It is not sufficient if your delete ONLY works when deleting the most recently created file.
- If you are going to trash your filesystem image (likely at some point), keep an extra copy around.

Update the README for Assignment 3.

Update the following sections of your README text file as needed:

- Instructions for compiling your programs
- Instructions for running your programs/scripts
- Any challenges you encountered along the way
- Any sources you used to help you write your programs/scripts

Submission Procedure

You must zip up your README file, your source code, and anything that is needed to compile it (such as a Makefile), and submit the entire zip to Canvas by the due date.