Lab05

Two weeks ago we learned about data structures used in FAT filesystems, and you created a tool to browse FAT-like disk images, based on the "Disk" library I provided for reading/writing blocks.

During the next few weeks I'd like you to build a similar tool for browsing Ext-like disk images! It should be built on top of the "Disk" library just like before and should provide a nearly identical look/feel to your current tool. You're welcome to re-use any code from your previous assignment to bootstrap the process; you will just need to replace the FAT-parsing parts with Ext-parsing code. 😊

Attached to the drop box is another "disk.img" file, this time in Ext-like format. Again, it contains several files and directories, and uses a block size of 512.

When run, your new tool should print the disk label, and then let the user do any of the following...

- dir List contents of current directory. Print type, size (for files), and name.
- cd <dir> Change directory ("cd .." should go to the parent directory)
- read <file> Read and print the contents of a file.
- pwd Print the current working directory.
- stat <file> Print the inode information for this file. ← NEW!

As you work, test your program against the provided disk image. Ultimately you should be able to browse the directories in the image, and read any files you find.

Style and presentation counts! It doesn't have to look like mine, but it should be easy to read and understand what information is being displayed.

Deliverables:

Please turn in…

- Your code, ***as plaintext file(s)*** (.txt files).
- A screenshot of it running on the provided disk image, showing that each command ***fully*** works.

Please do not zip the files, upload them individually.

Stretch goals:

Love programming? Having fun? Want to take it a step further? Here are a few other commands you might enjoy trying to implement. To be clear, these are totally 100% optional,

only for fun, only if you want to. You can earn full credit without doing any of these, don't feel obligated. 😊

- write <file> Make a new file called <file> and enter data
- mkdir <dir> Make a new directory called <dir>
- del <file> Delete a file called <file>
- rmdir <dir> Delete the whole directory <dir>
- link <name> <target> Link <name> to <target>