Description:

fsc.c

This file contains the low level fileio functions. These functions act as an obfuscation layer to the Filesystem api.

Search\_Directory will search through the Directory structure until it finds a file matching the supplied filename, when it finds this file it returns its inode numerical value.

Add\_to\_Directory will add a file to the Directory\_Structure that file will have the inode number supplied by the argument.

Get\_file \_in\_Directory: this allows for the collection of a files name given its position on the DS

Inode\_Read: will return an Inode from somewhere on the Inode\_List based on the argument given

Inode\_Write: will write an inode to a certain position in the Inode\_List

Block\_Read: Will read the data of a block to a character pointer given as an argument. The block it reads and how much it reads depends on the arguments as well. This allows for the collection of the raw data from the filesystem

Block\_Write: will write the data contained in a pointer supplied as an argument. the block it writes to and how much it writes varies based on the supplied arguments. This allows for data to be stored in the filesystem.

Superblock\_Read: returns the superblock struct

Superblock\_Write: will allow a function to change the superblock to another superblock supplied as an argument.

fsi.c

Initialize\_Filesystem: this initializes all necessary values for the filesystem to function for the first file. It also established the file that will be used for logging. Sets a superblock, allocates space for the Disk\_Blocks, and labels the inodes with their appropriate numbers

Create\_File: this reads a file’s binary data into a buffer. This buffer is then sent to the Write\_File command which writes the file to a place in the filesystem. It also collects all necessary information for that file’s assigned Inode

Open\_file: Sets the open flag for a file

Read\_File: Reads a file in the filesystem by calling Block\_Read

Write\_File: Will write to a file in the filesystem this is used by create file to save the binary data of a file to the filesystem.

Close\_File: sets the open flag for a file to closed if its open.

fsa.c

write\_into\_filesystem: this function uses a very similar method to those used in the previous assignment to get a list of files. In this case on each file it calls the function Create\_File.

make\_filesystem\_summary: this function will generate a file that gives a general summary of what is in the file system, and the details of its location. This is done by iterating throught the Directory structure using functions from fsc.c.

read\_images\_from\_filesystem\_and\_write\_to\_output\_directory: will take all the files in the filesystem with an extension of .jpg and will write their data into new .jpg files in an output directory

generate\_html\_file: this contains helper functions that help strip the filename of its extension and helps get the extension. By collecting all files in the outputdir one can then run the convert command on them with a little bit of string manipulation to create thumbnails for them.