­­­­­ Justin Block

Joe Deleeuw

COP4610 Assignment 3

Design of the Program

­­

This program simulates a FAT32 system. It takes multiple commands as per the instructions of the assignment and based on the command and parameters the user enters specific actions occur. The program consists of a main file called fatmod.cpp this handles all the commands and delegates them to filesystem.h the filesystem is the main brawns behind the program and takes care of most of the parsing, cd, ls and other commands main functionality.We have a couple different structs/classes to help organize the data that is stored such as our directory file.

The program takes the file directory and based on the FAT32 specs we parse the information and store it in the data structures we have. When a user wants to do something like rm a file we flag it as per the FAT32 specs from our data structure (vector) and also from the filesystem itself. This ensure changes propogate fully and allows for easy access. For the most part we use our files vector as reference and we continuously add to it as the user changes down to deeper directories.

Overall this program should be extensible for future use and with more time we would refactor it as much as possible to extend the capabilities further. We utilize a lot of filesystem parsing and have worked very hard on it.

Development Journal

GIT LOG is in documentation folder. It is called gitlog.txt

Division of Labor

For the most part we paired programmed and either worked using c9 (a way to code in the same file at the same time) or we worked both at the majors lab. This allowed us to bounce ideas off each other and perform “rubber ducky debugging” or explaining what bugs we were having and then helping each other find the root cause. This worked very well for us and allowed us to get things done in an efficient manner.

Missing Functionality

* MKDIR
* UNDELETE

Bugs/Issues

* Issue: CD.. goes back to the main directory instead of the parent. This is because we need to improve parsing of things.
* Issue: LS.. was causing segfault so it has been disabled (and rm ..) along with it. We dissected the issue to be caused by calling parseInteger with an out of range value.
* CAN cd or LS to a child that is not within the current – caused because our parsing for our data structure needs more edge case checking.
* Rm and rmdir can display a value after removal.

Suggestions for Assignment Write-up

**Justin:** I found the directions to be a bit hard to follow at the beginning. It would have been nice to show a small demo of how to go parse just one bit of code. I found the program to be out of the ordinary which is good and it was interesting when things all started falling into place. I would just try and add more visuals to the slides as for me at least I find visuals very helpful.

**Joe:** I wish there was a bit more direct guidance from the professor regardint the assignment itself, I am not sure if that is something that needs to be fixed however. The assignment itself was good in that it helped us learn a lot about how filesystems work in general.