Avik Bag
William Fligor

Mike Lui
ECEC353 Project 4
9/2/2017

Report

For this report, the main goal was to implement a file mounting system that is found in FUSE systems. The bbfs system was used. Essentially with this, particular directory can be mimicked by the root directory.

The first step of this project was to ensure that it could print to the log file after running an echo command, which cleared out the file. The way to counter this was to change the fopen open flag ‘a’. This allows any character array to be appended to the file.

The next two deals multiple file systems. The script already allows for multiple file system mounts. The logging system also allows for multiple file system logging but the issue here is that the log messages are interweaved. The way to counter that is by implementing a lock around where the log_msg() method in log.c, which carried out a vfptintf() to dumb onto the log file. This is the method that is responsible for logging to the log file. There are multiple methods that are calling the log_msg() method. These methods needs a lock before the log_msg() call, and after the final struct call. This ensures that the entire message is printed rather than having it print one line from one file mount and then another from another, causing the data to be interleaved.

One issue however is that we can’t account for the order of the information being written to the log file. We first tried to implement a queue in shared memory but that proved to be too difficult. We tried implementing a linked list which didn’t work. But regardless, the data being written is atomic, which means that only one log message from one mount can be written onto the log file at any given time.