CSCI 4730 – Project 4

Gregory Woolsey

This project acts as a file system simulator that has numerous commands that the user can perform on files or directories. For this project, I was tasked with creating the methods for cat, cp, rm, and ln for file manipulation, and mkdir, rmdir, and cd for directory manipulation.

Starting with file manipulation, cat functions by getting the inode of the file, then allocates memory to a str to be used to print the file contents. The disk memory of the inode is then read and sent to str, which is then called in a print statement to print the file's contents. The remove function finds the user-defined file to remove, sets the memory blocks of the inode to 0, and lowers the numEntry of the current directory by 1 at the location of the file, shifting every other file in the array 1 place to the left, then the superblock and inode of the file are freed. Link acts by taking the inode of the specified file, and writes a new inode of that file to a specified destination. Copy acts by finding the specified file and creating a whole new file with its own inode and block memory space, which is then added to the end of the current directory.

Next with directory manipulation, mkdir creates a new directory entry and adds it to the end of the current directory. Then, two new directories are created, root and parent, and are added to a new dentry to act as the newly created directory. The rmdir function acts very similar to the remove function of the file manipulation, reducing numEntry size and freeing the block memory of the specified inode. The cd function functions by writing the block of the current node of the current directory to memory, then reading the block memory of the specified inode into the current directory, changing the current directory in the simulator.