Dominic Taraska
Tug76525
Project 4

This project contains a simple file system implementation done in C. For my implementation, I decided to use an Inode design. mmap() calls are used to map the necessary data structures to the disk.
The partition of the virtual disk file is:

[SUPERBLOCK] [Inode bit-map] [data block bit-map] [inode blocks] [DATA BLOCKS]

Data blocks comprise half the disk, the other half belonging to the meta information.
Directory entries are stored inside inodes, which is its own data structure.

The directory structure itself is as such:

- **Boolean to indicate wether it is in use**
- **Boolean to indicate wether it is a directory**
- **int to indicate the parent directory**
- **name (maximum length of 12)**
- **extension (max length of 4)**
- **size of the file**
- **creation date, last modified**
- **pointers to file data**

When a new entry is created, the current directory that is open is kept track of with a variable. The variable always represents the directory that is currently open, and so any new sub directories will be created in that location.

Available data blocks as well as available inodes are kept track of by using their own respective bitmaps (one for inode blocks and one for data blocks).