LFS Design Details
Yunhao Xu
Yifeng Li

Log Layer: Contains the representation of a log that is stored in the flash.
-Log begins with the superblock (implemented in superblock.cc)
-Log is composed of segments (implemented in segment.cc)
-Segments are composed of blocks (block.cc)
-A cleaner (to be implemented)

Superblock contains system metadata, such as the size of the flash, size of a file block, size of a segment, which segment in the log programs can perform the next write, segment usage data, wear limit, ifile—which contains all inodes. These metadatas are stored within the checkpoints created by the superblock.

Segment is essentially a list of file blocks. The first few blocks of a segment are allocated for metadata (segment summary). The segment summary contains metadata such as usage numbers, block sizes, free blocks remaining, and segment id—which identifies itself from all other segments. The rest of the blocks are used for actual file storage. A segment from the log layer is cached in memory for read/write efficiency.

Blocks are the basic storage elements of segments. They contain block id, which makes all blocks within a segment unique. They also contain inode number and inode version, which refers to which inode/file they belong to and the version of the inode.

The cleaner will check the inode id and version. If the same inode in the ifile has a newer version or is pointing to a block in a different block (in different segment, or different block in same segment), then the block is no longer in use and ready to be cleaned.

File Layer: Contains references to files blocks that are stored on the log.
-File layer has ifile (ifile.cc).
-Ifile keeps track of all inodes (inode.cc), which refer to file blocks stored in the log.

Inode contains 4 direct block references, which directs to the proper block of a segment in the log. Inode also contains indirect blocks, which points to a block that contains more references to direct blocks.

The implementation has both classes and structs. E.g. when a segment is written to the log, the class representation is converted into a struct of equivalent meaning for easy storage. When reading, the struct gets converted into a class, for ease of usage.

Directory Layer: A file that contains hierarchical representation of the directory.
-The directory layer is implemented within a file, which contains the hierarchical representation of the directory. The directory class should parse what is written in the file and keeps record enforces the logistics of the directory tree (what is possible, what is not possible).

The class should also parse its internal representation of the directory into a struct and write it into the special directory file.