Lydia Parsa

- filesystem.c uses the framework provided by softwaredisk.c to create an inode based block allocation system for my filesystem.

- The software disk has 4096 blocks, each with 4096 bytes.

- The software disk partitioning:

  - Block 0 is a data bitmap to track free disk blocks
    - 0 means the block is free
    - 1 means it is used
  - Block 1 is a inode bitmap to track free inode blocks
    - 0 means the corresponding inode is free
    - 1 means it is used
    - Each inode has a corresponding index, the position of a bit in the inode bitmap corresponds to the index of that inode.
  - Blocks 2-5 store the inodes
    - 128 inodes per block
    - 4 blocks
    - 128*4 = 512 inodes, thus 512 files
  - Blocks 6 – 69 are directory entry blocks
    - 8 entries per block
    - 64 blocks
    - 64*8 = 512 entries, thus maximum allotted files is 512
  - The remaining blocks are dedicated to storing file data

- The functions allocate_bit, free_bit, and used_bit track and allocate the space in the bitmaps.

- Design Limitations
  - File names are limited at 256 characters.
  - Can only support up to 512 files.
  - A file name cannot be composed of null characters.