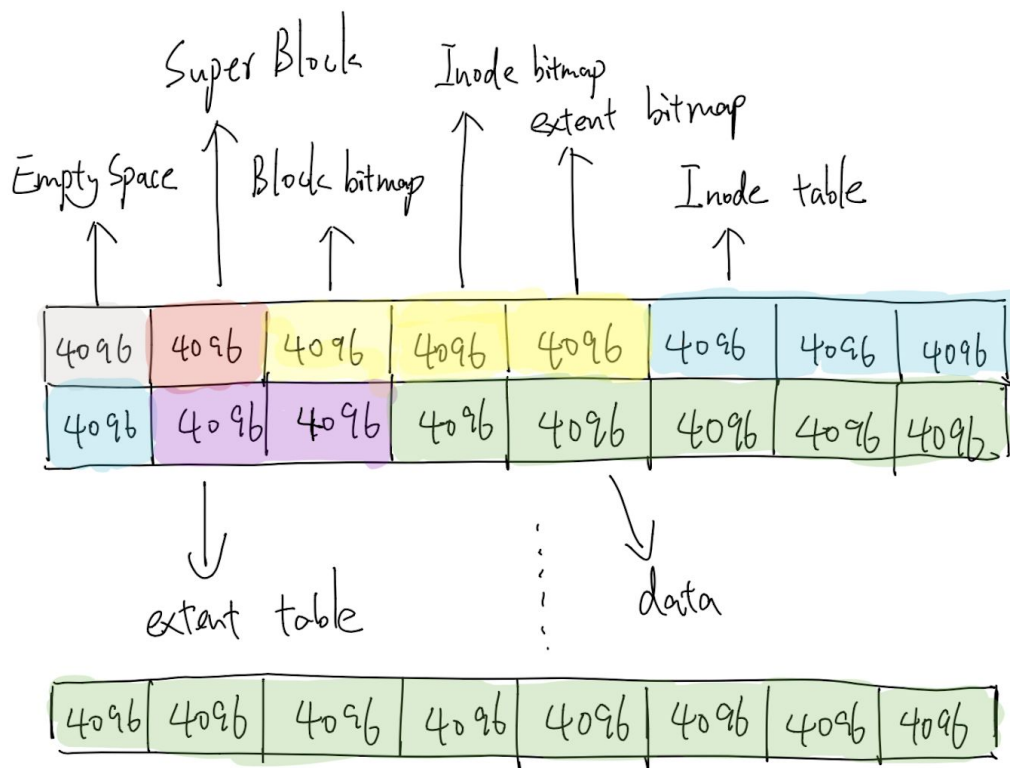


CSC369 Assignment 1 Proposal

Ruijie Sun, Xin Luo

Q1 Disk image diagram.



Q2 Space partitioning

The file system splits the HD into blocks of 4096 bytes. The first block is reserved as empty space. The 2nd to 11th blocks are Metadata blocks and all remaining blocks are for real data. In more detail, the 3rd block is for Block bitmap which enables us to keep track of free data blocks. The 4th block is for Inode bitmap which enables us to keep track of free inodes. The 5th block is for Extent bitmap which enables us to keep track of extents. The 6th, 7th, and 8th, and 9th blocks are for the Inode table. The 10th and 11th blocks are for extent table. All other blocks are data blocks.

Q3 A description of how you will store the information about the extents that belong to a file.

The corresponding file inode will store the pointers to extent structs(each extent struct is stored in extent table). For each extent struct, it stores the starting address of data block and the number of consecutive blocks belonging to the extent. It will write extent related information into the data block unexpectedly if it run out of space to store extents. And this is the reason we need to keep track of the free extent number.

Q4 Describe how to allocate disk blocks to a file (In other words, how do you identify an extent and allocate it to a file? What are the steps involved?)

When we have a new file to allocate disk blocks, the following steps will be executed:

Step 1: Based on inode bitmap, set up a new inode(Create inode and modify inode bitmap).

Step 2: According to the size of the input file, the file system looks for the contiguous empty blocks in the blocks_bitmap. If one set of contiguous empty blocks is enough then the file system constructs one extent. Otherwise, keep looking for the next contiguous blocks, this will come up with several extents.

Step 3: Modify extents_bitmap and block_bitmap based on used extents and used blocks.

Q5 Describe how to free extents.

Change the value on the corresponding extent bitmap from 1 to 0 (from occupied to empty) without changing inode bitmap information.

Q6 Describe the algorithm to seek to a specific byte in a file.

Iterate all extent struct pointers in the inode to accumulate the used data byte to match the specific byte number.

Q7 Describe how to allocate and free inodes.

Allocate: Change the value on corresponding inode bitmap index from 0 to 1, construct a new inode then stored into inodes_table;

Free: In the inodes_bitmap, change the value on the corresponding index to 0;

Q8 Describe how to identify a free directory entry.

The corresponding directory entry's **inode index** in inode-bitmap is 1, but all related **extent index** in extent bitmap 0.

Q9 Describe how to search a directory for a particular entry name.

Look through all the entries in that directory, check if there is a match. If so, follow the matched name and it's inode, we can get the data of that entry.