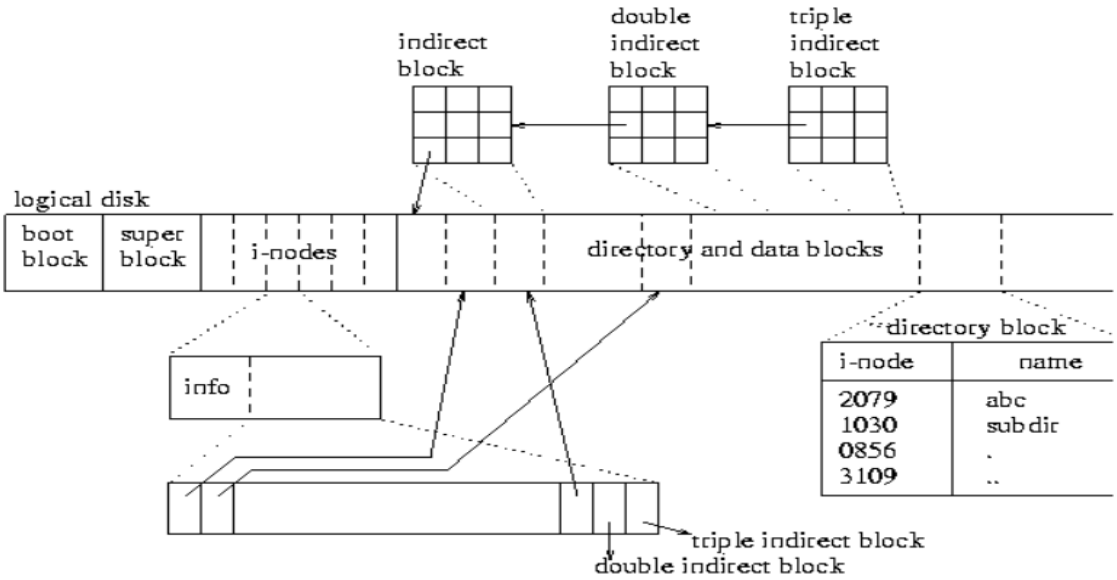
PART 1

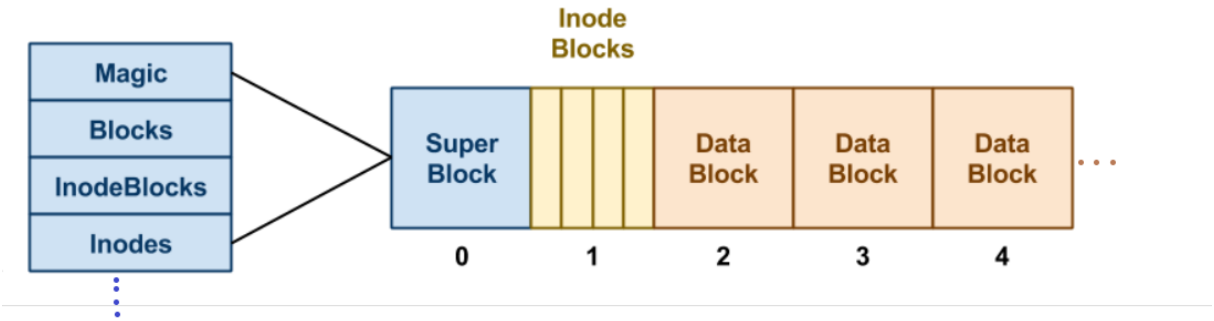
My File System is like below Figure.



Superblock Structure

A *superblock* is a record of the characteristics of a file system, including file system magic number, number of blocks in file system , number of blocks reserved for inodes , free inodes, number of free inodes ,the *block* size, the empty and the filled blocks and their respective counts, the size and location of the inode tables.

```
struct superblock{
        uint64_t MagicNumber;
        uint64_t Blocks;
        uint64_t InodeBlocks;
        uint16_t freeInode;
        uint16_t freeBlockNumber;
        uint32_t **freeBlocks;
        uint32_t **dataBlocks;
        struct inode *iNode;
        struct inode *freeINodes;
        int blockSize;
};
```



I-Node Structure

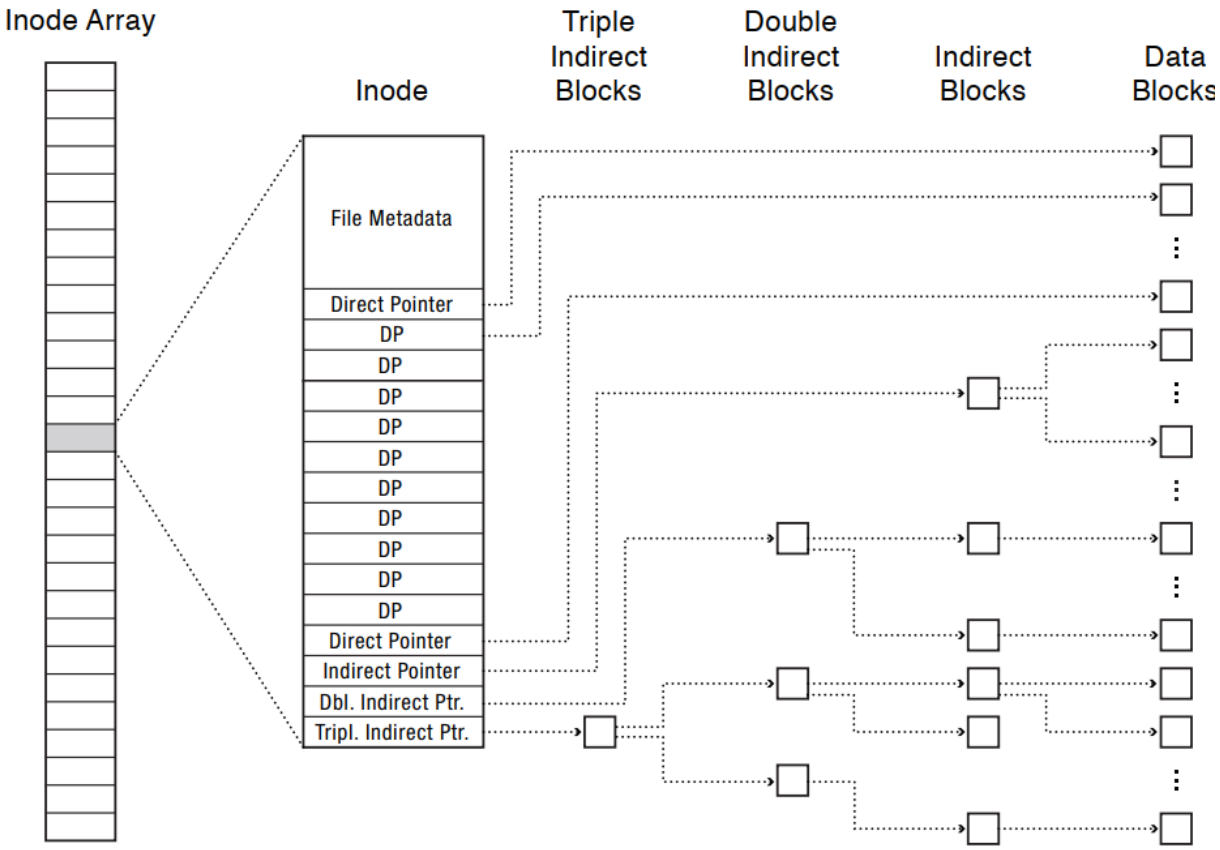
An inode is an entry in inode table, containing information (the metadata) about a regular file and directory.

My inode structure designed like this.

```
struct inode{
        uint64_t inodeNumber;
        uint64_t  size;
        char name[90];
        char time[102];
        uint64_t *direct[10];
        struct SingleBlock singleIndirect;
        struct DoubleBlock doubleIndirect;
        struct TripleBlock tripleIndirect;
};
```

A direct pointer points a data blocks.Single Indirect Block points data blocks. Double Indirect Block points 2 single indirect blocks.Triple Indirect Block points 2 double indirect blocks.

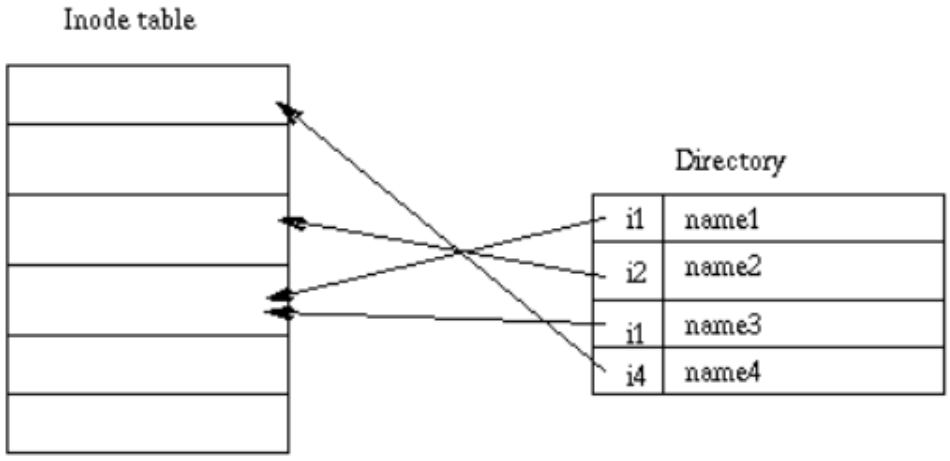
Metadata includes inode number,size,last modification date and time,name of file.



Directory Structure and Directory Entries

```
struct dir{
    struct inode *iNode;
    char filename[60];
};
```

Each file has own inode. Directory entries are kept a with struct pointer array and its size has been adjusted according to the amount of byte required for the file to be 1mb.



Free Blocks and Free I-Nodes

Free blocks and free i-nodes are kept in super block.Size of each block is set to 1024*parameter1. For mySystem.dat file to be 1mb, block numbers and directory numbers have been adjusted according to the required byte amount.