GIT Department of Computer Engineering CSE 312/504 - Spring 2022
Homework #3
Report

Sefa Çiçek
1801042657

1. Directory Structure and Directory Entries

Directory and file have same properties such as name and I-node. One structure is enough for them. The size of I-node and file/directory name in the “File” structure are determined according to Figure 4.32.

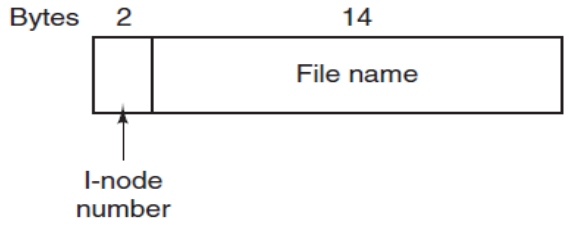


Figure 4-32. A UNIX V7 directory entry.

```
struct File {
    uint16_t iNode;  // 2 byte
    char name[NAME_LENGTH];  // 14 byte
} __attribute__((__packed__));
```

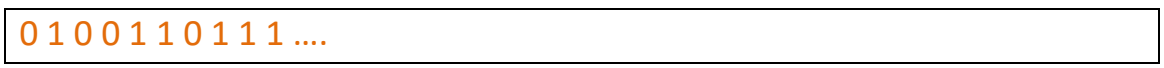
Files have I-nodes to store their attributes like size, type and creation time.

* An I-node initially contains 2 file -> current (.) and parent (..) that’s why size of an I-node equals to 2 file size.
* Type is either sfile or directory.
* Creation time is found by the ‘time’ function in time.h library.
* To determine whether I-node is free or not, we use ‘connected’ property.
* I-node structure also has direct and indirect arrays to store block addresses.

```
struct Inode {
    uint32_t size;
    uint8_t  type;
    int32_t  creationTime;
    int32_t  modificationTime;
    uint8_t connected;
    uint16_t direct[DIRECT];         // 10
    uint16_t singleInd[INDIRECT1];   // 1
    uint16_t doubleInd[INDIRECT1];   // 1
    uint16_t tripleInd[INDIRECT1];   // 1
}__attribute__ ((__packed__));
```

2. Keeping Empty Blocks

We can easily access the empty blocks thanks to the bitmap array.



* The bitmap array is initially filled with zero, and the responsible index returns to 1 when a new block is created. Like blocks, I-nodes have a property ‘connected’ to distinguish free and attached I-nodes.

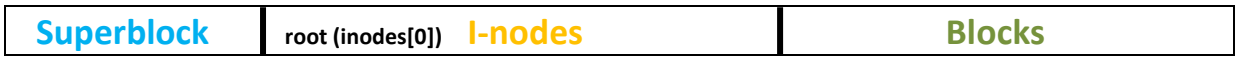
3. Superblock – Crucial Informations

Superblock structure has lots of informations about file system. Generall, a superblock keeps block size, block and inode offset etc. Addititionally, keeping 'last used inode index and adress' parameters makes things even easier.

```
struct SuperBlock {
    uint32_t lastUsedInodeIndex;
    uint32_t lastUsedInodeAddress; // initially inode[0]
    uint16_t blockSize;  // user-defined
    uint16_t blocksNum;
    uint16_t inodesNum; // 256 - hardcoded
    uint32_t blockOffset;
    uint32_t inodeOffset; // inception of root
    uint8_t bitMap[1024];
} __attribute__ ((__packed__));
```

* A superblock has block size propert which is taken by user. (4 KB etc.)
* Blocks and inodes num is basically used for calculating offsets or empty place.
* Block and I-node offset is necessary to find a new place for block or inode.
* Last used I-node address and index are kept to find where we last stayed on part of I-node structure.
* Bitmap is used to find the free blocks.

4. Disk Partition



0 4 MB

Disk fragmentation is determined by needs and easy accessibility. For example, root is not a separate part, it is kept on I-nodes to makes root operations easier.

* The reason of keeping superblock in the first place is to make it indicate to the same point as the file system pointer. In this way, properties of superblock can be easily accessed.