Department of Computer Science and Engineering

The Chinese University of Hong Kong

CSCI3150/ESTR3102 Introduction to Operating Systems

Tutorial 2: Inode – Define file metadata

Objectives:

(1) Understand the role of inode in file systems.
(2) Learn how to use an inode to describe a file or directory. Learn how to utilize the directory entry and how to traverse the multi-level directory tree.

Note: In the project, you will be asked to implement a simple file system, and this lab is the preliminary knowledge.

1. Background

1.1 Inode

Inode is a record of the metadata of a file, for example, its type (regular file or directory), its size, its data block number, the indexes of those data blocks, etc. Once we obtain the inode of a file, we can obtain all of its data via data blocks accordingly.

An example structure of inode can be found below:

```
struct inode /* The structure of inode, each file has only one inode */
{
        int i_number; /* The inode number */
        time_t i_mtime; /* Creation time of inode*/
        int i_type; /* Regular file for 0, directory file for 1 */
        int i_size; /* The size of file */
        int i_blocks; /* The total numbers of data blocks */
        int direct_blk[2]; /*Two direct data block pointers */
        int indirect_blk; /*One indirect data block pointer */
        int file_num; /* Number of files under a directory (0 for regular file)*/
};
```

1.2 directory mapping

It is easy to imagine how a regular file is stored in the file system – first it has its inode, which contains metadata. Then it has some data blocks whose pointers are stored in the inode.

How about directory? In the lecture we have learned that the directory is a special type of file. It also has the inode and data blocks. The only difference is that the data blocks do not store user data, instead they store a list of special data structure called ***dir_mapping***, which is the mapping of file name and inode number.

An example structure can be found below:

```
struct dir_mapping /* Record file information in directory file */
{
        char dir[20]; /* The file name in current directory */
        int inode_number; /* The corresponding inode number */
};
```

Thus, when we look for a specific file under a directory, we are traversing these ***dir_mappings*** one by one by comparing the ***dir*** with the file name. Once the ***dir*** equals the file name, the corresponding ***inode_number*** is found and can be returned.

2. A simple *inode* Program

Next, we will provide a simple program to show the structure of inode and how to use it.

Copy the "HD" file into your current directory, which is the hard disk simulation file and has been initialized properly.

There are five files under the "code" directory:

- **HD**: It is the hard disk simulation file.
- **superblock.h**: It contains the structure of superblock and the parameters of SFS.
- **inode.h:** It contains the structure of inode and the structure of dir_mapping.
- **inode.c**: It contains the basic read function about inode. There are also print functions to show inode information and inode region on hard disk. ***print_dir_mappings*** will show the dir_mappings under a specific directory.
- **Inode-test.c**: It contains two test cases. The first case is about how to show the inode information and the inode region on the hard disk file. The second case is about how to show all the dir_mappings under the root directory.

2.1 Compile and Run

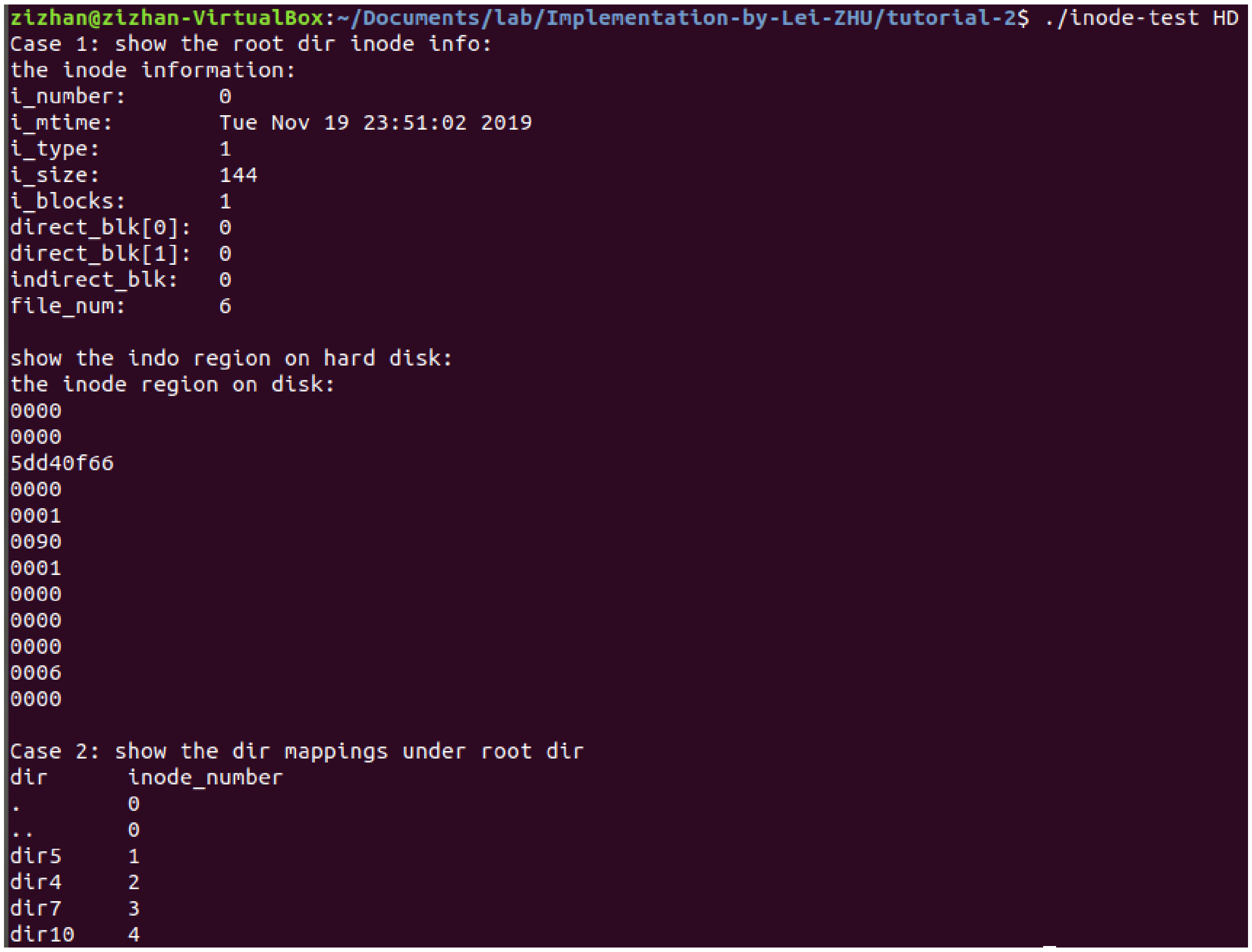
Compile:

```
gcc –o inode-test inode-test.c inode.c
```

Run:

```
./inode-test ./HD
```

You will see two cases on your screen:



```
zizhan@zizhan-VirtualBox:~/Documents/lab/Implementation-by-Lei-ZHU/tutorial-2$ ./inode-test HD
Case 1: show the root dir inode info:
the inode information:
i_number:       0
i_mtime:        Tue Nov 19 23:51:02 2019
i_type:         1
i_size:         144
i_blocks:       1
direct_blk[0]:  0
direct_blk[1]:  0
indirect_blk:   0
file_num:       6

show the indo region on hard disk:
the inode region on disk:
0000
0000
5dd40f66
0000
0001
0090
0001
0000
0000
0000
0006
0000

Case 2: show the dir mappings under root dir
dir      inode_number
.        0
..       0
dir5     1
dir4     2
dir7     3
dir10    4
```

The first case prints out the inode information of the root directory and show its inode region on hard disk. Please see the detailed implementation in code **inode.c.**

The second case prints out all the dir_mappings under the root directory. You can use this function to show other directories.

2.2 Implementation

The core functions of ***inode.c*** are listed below (they should be useful for the project and can be used it):

- ***read_inode()***: initiate an inode structure pointer and read an inode from HD.
- ***print_dir_mappings():*** first read the inode corresponding to current ***i_number***. Then initiate a **DIR_NODE** type pointer ***p_block***, which will point to the entry list. Read the block into memory according to the ***direct_blk[0]*** of inode (SFS only support 100 inodes, 100 * sizeof(DIR_NODE) < 4096, thus only **direct_blk[0]** will be used), whose pointer is ***p_block***. Then iterate through the enrty list and print out all the entries.

More detailed implementation can be found in ***inode.c*** and ***inode-test.c***.