Operating Systems
CSCI 5806

Spring Semester 2025 — CRN 22968

Term Project — Step 4 — Inode Access

Target completion date: Friday, March 21, 2025

Goals

- Provide functions to provide access to inodes contained in an ext2 file system.

Details

In this step, we access inodes, the data structures containing all metadata for files — except the name. Accessing inodes also involves the inode bitmaps, which indicate whether or not a given inode is in use or is free.

To begin, you will need two functions:

- **`int32_t fetchInode(struct Ext2File *f,uint32_t iNum, struct Inode *buf)`**
  Read the inode whose index number is **`iNum`** and store the information in the buffer.
- **`int32_t writeInode(struct Ext2File *f,uint32_t iNum, struct Inode *buf)`**
  Write the inode whose index number is **`iNum`** from the given buffer.

Finding an inode is not a difficult process, but is not a trivial process either. Essentially, inodes are stored in an array on the disk. However, there are two quirks:

1. The first inode number is `1`, not `0`.
2. The array is sliced into strips, with one strip stored in each block group.

Dealing with a `1`-based array is easy, just subtract `1` from the inode number at the start of processing.

Handling a distributed array is also not difficult; it is basically a three-step process:

1. Find the block group containing the desired inode. The number of inodes in each block group is stored in the superblock.
2. Find the block containing the desired inode; treat that block like an array of inodes. The first block containing inodes is stored in the block group descriptor table. Within one block group, all blocks containing inodes are stored consecutively. The size of an inode is stored in the superblock.
3. Access the inode from the block containing the inode.

For writing of files, you will also need to work with the inode bitmaps to know which inodes are currently in use. In general, there are three functions to deal with the inode bitmaps:

- **`int32_t inodeInUse(struct Ext2File *f,uint32_t iNum)`**
  Returns true if the inode is marked as allocated, false if it is not marked.
- **`uint32_t allocateInode(struct Ext2File *f,int32_t group)`**
  Select any unmarked inode in the given block group, mark it as allocated and return its inode number. If **`group`** is `-1`, then select any available inode from any group.
- **`int32_t freeInode(struct Ext2File *f,uint32_t iNum)`**
  Mark the given inode as free.

Other Functions

You will probably want a function to display an inode with the fields labeled and in text form. The following example illustrates my version of these.

Example

This is the output from my step 4 program, on the fixed VDI file with 1KB blocks. It shows the root inode — inode 2 — and the file system's **`lost+found`** directory — inode 11 — in readable form.

```
Inode 2:
Offset: 0x0
   00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f    0...4...8...c...
  +-----------------------------------------------+  +----------------+
00|ed 41 e8 03 00 04 00 00 d3 ea bc 56 a8 bf ba 56|00| A          V   V|
10|a8 bf ba 56 00 00 00 00 e8 03 04 00 02 00 00 00|10|   V             |
20|00 00 00 00 03 00 00 00 03 02 00 00 00 00 00 00|20|                 |
30|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|30|                 |
40|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|40|                 |
50|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|50|                 |
60|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|60|                 |
70|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|70|                 |
80|                                               |80|                 |
90|                                               |90|                 |
a0|                                               |a0|                 |
b0|                                               |b0|                 |
c0|                                               |c0|                 |
d0|                                               |d0|                 |
e0|                                               |e0|                 |
f0|                                               |f0|                 |
  +-----------------------------------------------+  +----------------+

                Mode: 40755 -d-----rwxr-xr-x
                Size: 1024
              Blocks: 2
           UID / GID: 1000 / 1000
               Links: 4
             Created: Tue Feb  9 23:42:16 2016
         Last access: Thu Feb 11 15:10:59 2016
   Last modification: Tue Feb  9 23:42:16 2016
             Deleted: Wed Dec 31 19:00:00 1969
               Flags: 00000000
        File version: 0
           ACL block: 0
       Direct blocks:
                 0-3:          515           0           0           0
                 4-7:            0           0           0           0
                8-11:            0           0           0           0
Single indirect block: 0
Double indirect block: 0
Triple indirect block: 0
```

```
Inode 11:
Offset: 0x0
   00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f    0...4...8...c...
  +-----------------------------------------------+  +----------------+
00|c0 41 00 00 00 30 00 00 88 bb ba 56 88 bb ba 56|00| A   0     V   V|
10|88 bb ba 56 00 00 00 00 00 00 02 00 18 00 00 00|10|   V            |
20|00 00 00 00 00 00 00 00 04 02 00 00 05 02 00 00|20|                |
30|06 02 00 00 07 02 00 00 08 02 00 00 09 02 00 00|30|                |
40|0a 02 00 00 0b 02 00 00 0c 02 00 00 0d 02 00 00|40|                |
50|0e 02 00 00 0f 02 00 00 00 00 00 00 00 00 00 00|50|                |
60|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|60|                |
70|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|70|                |
80|                                               |80|                |
90|                                               |90|                |
a0|                                               |a0|                |
b0|                                               |b0|                |
c0|                                               |c0|                |
d0|                                               |d0|                |
e0|                                               |e0|                |
f0|                                               |f0|                |
  +-----------------------------------------------+  +----------------+

                Mode: 40700 -d-----rwx------
                Size: 12288
              Blocks: 24
           UID / GID: 0 / 0
               Links: 2
             Created: Tue Feb  9 23:24:40 2016
         Last access: Tue Feb  9 23:24:40 2016
   Last modification: Tue Feb  9 23:24:40 2016
             Deleted: Wed Dec 31 19:00:00 1969
               Flags: 00000000
        File version: 0
           ACL block: 0
       Direct blocks:
                 0-3:          516         517         518         519
                 4-7:          520         521         522         523
                8-11:          524         525         526         527
Single indirect block: 0
Double indirect block: 0
Triple indirect block: 0
```