VE482 Lab Report

Lab 11 - Fall 2020

Boming Zhang
Chujie Ni
Qinhang Wu
Zhimin Sun

Goals of the lab

- Understand filesystems
- Work with a filesystem in userspace
- Learn the basics of FUSE

VE482 Lab 11

What is a filesystem?

File system is a system that controls how data is stored and retrieved. By separating the data into pieces and giving each piece a name, the data is easily isolated and identified. Taking its name from the way paper-based data management system is named, each group of data is called a "file." The structure and logic rules used to manage the groups of data and their names is called a "file system."

How is the Linux VFS working?

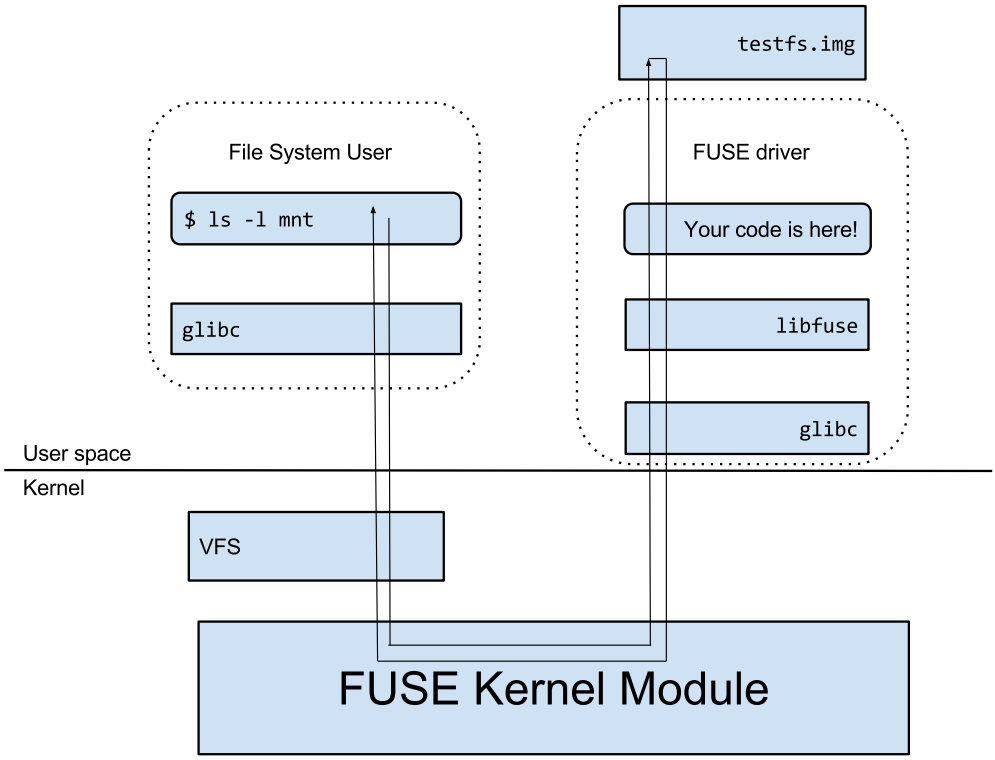
VFS is the software layer in the kernel that provides the filesystem interface to userspace programs. It also provides an abstraction within the kernelwhich allows different filesystem implementations to coexist.

The key point is that VFS is a layer of code which implements generic filesystem actions and vectors requests to the correct specific code to handle the request. A VFS specifies an interface between the kernel and a concrete file system. Therefore, it is easy to add support for new file system types to the kernel simply by fulfilling the API. Therefore, we can only focus on the API provided by VFS instead of all the details associated with the concrete file system.

What is FUSE, and how does it interact with the VFS? Can you sketch it quickly to make it clearer?

FUSE for Filesystem in User Space. An open source framework for implementing filesystem in user land.

FUSE has a daemon process that deal with the file operations. For every file operations, it take these operations in user space and convert it into kernel space commands and finish the job, twice each operation, one for request and the other for response.



Implementation

We first try to recover the file. After download `del.tar.bz2`, run:

```
# cd .del
# vim -r lemondbfs.c
```

And we will get the recovered version of `lemondbfs.c`.

Then we implement the lemonDBfs in `./lemondbfs/`.

Task done

1. for each .tbl file show a dir in mount_dir
   eg. for file mTable10.tbl show dir mtable10
2. inside each dir show the corresponding table in file orig.tbl
   eg. for file mTable10.tbl show mtable10/orig.tbl
3. inside each dir show an empty file called .query
   eg. for mTable10.tbl show the empty file mTable10/.query
4. when a query is written to a .query file, apply it in the orig.tbl and create the file res.tbl containing the result of the query
   eg. upon a write on mTable10/.query, run the query (error on wrong query) and create mTable10/res.tbl with the result of the query
5. lemonDBfs becomes more stable (not allow to run as root)
6. Allow more than one query in the .query file
7. When many queries are input allow temporary result tables to be created.
8. Allow the deletion of .tbl files: when a dir is deleted in mount_dir the corresponding .tbl file should disappear

Demo

```
$ cd /lemondbfs/src
$ make
(so many outputs)
$ ls ../db_dir/
mTable0.tbl  mTable1.tbl  mTable3.tbl
$ ./lemondbfs ../db_dir/ ../mount_dir/
about to call fuse_main
lm_data: 0x563217b8deb0
$ ls ../mount_dir/
mTable0  mTable1  mTable3
$ ls -lha ../mount_dir/mTable0/
total 0
drwxr-xr-x 1 root root   0 Jan  1  1970 .
drwxr-xr-x 2 root root   0 Jan  1  1970 ..
-rwxr-xr-x 1 root root 65K Jan  1  1970 orig.tbl
-rwxr-xr-x 1 root root   0 Jan  1  1970 .query
$ cat ../mount_dir/mTable0/orig.tbl
(so many outputs)
```

```
$ echo "UPDATE ( c0 100 ) FROM mTable0 WHERE ( c2 > 0 );" > ../mount_dir/mTable0/.query
$ ls -lha ../mount_dir/mTable0/
total 0
drwxr-xr-x 1 root root    0 Jan  1  1970 .
drwxr-xr-x 2 root root    0 Jan  1  1970 ..
-rwxr-xr-x 1 root root  65K Jan  1  1970 orig.tbl
-rwxr-xr-x 1 root root    0 Jan  1  1970 .query
-rwxr-xr-x 1 root root 104K Jan  1  1970 res.
$ rmdir ../mount_dir/mTable3/
$ ls ../mount_dir/
mTable0  mTable1
$ ls ../db_dir/
mTable0.tbl  mTable1.tbl
```

Reference

- https://en.wikipedia.org/wiki/File_system
- https://en.wikipedia.org/wiki/Virtual_file_system
- https://ic.unicamp.br/~islene/2s2013-mo806/vfs/andre-zhen.pdf
- https://sites.cs.ucsb.edu/~trinabh/classes/w19/labs/lab6.html