VE482 — Introduction to Operating Systems

Lab 11

Manuel — UM-JI (Fall 2020)

Goals of the lab

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

1 Introduction

You have already been working at CoolZone IT solutions for about two weeks and you like it very much. You are leaning many new things and times flies so fast in such a nice environment; there is no comparison with Lemonion Inc. where you could not even go to the restroom without Mr. Frown complaining you were deserting your cubicle for too long or too often.

However you now fully get the measure of your father's words: "The problem is not hard, it simply requires patience". You are indeed spending so much time rebooting and fixing bugs, and at times you definitely feel overwhelmed by frustration. You would definitely like to work more on dadfs and implement file deletion, but facing the long 'Edit-Compile-Run-Crash-Reboot-FindFile' loop on your free time, no, you do not want it. So as you are on your way to have lunch with your parents you think of a nice way to present things to your dad and explain him that you want to rest a bit before going ahead with his filesystem.

All along the meal you redirect the conversation on other topics and avoid any word that could be connected to filesystem, you do not even pronounce the word "dad". Unfortunately, at the end of the lunch your father says "Dadfs is waiting, let's go!". At first you pretend you did not hear him and move to the kitchen to help your mum washing the dishes, but she quickly tells you to join your dad who is waiting for you. As you can neither escape it nor delay it any longer you decide to face you father and tell him that today you will not be able to work on dadfs.

Your father patiently listen to your arguments and replies. "I understand your frustration, this also happened to me. But don't be so negative, in fact you're learning a great lesson that can help you becoming a better programmer: now that you understand that the issue does not lie in the problem but in its perception, you can overcome many common obstacles. For instance you can defy the seduction of 'dynamically typed' languages, and favor 'statically typed' ones, even if it means spending 30 minutes debugging type mismatches in 50 lines of code... Look at the bright side, you're leaning patience, long-suffering, and perseverance, those are all very important assets if you want a successful career."

2 Tasks

Seeing that you are half-convinced by his motivating words you dad carries on. "You know what, I have the perfect solution for you: instead of working on the module, in kernel mode, let's work in userspace? Everything will be much easier and you can work with dadfs in the same way as with any other regular C program. You just need to use FUSE, this is a great project that acts as an interface to export a filesystem to the kernel. No need for functions to share buffers between the userspace and the kernel

space, this is FUSE's job, not yours. You simply type your code and treat FUSE as a dependency/library to build your program."

At hearing that their is no risk to crash the kernel you feel relieved and want to try this "magical library". As you ask your dad how to get started, he tells you that he has already prepared some materials. Your job for today is to port dadfs to FUSE and implement file deletion. With that idea in mind he shares with you some resources.

Since he wants to go soon for his nap he quickly shows you a couple of FUSE links: the FUSE documentation, and the libfuse source code.

As you seat down at the desk your dad understands that you are ready to start. He then leaves the room and goes for his nap. Although the temptation is high to play video games instead of porting dadfs to FUSE, you decide to start working on this new short project. After a few minutes you fully understand the global idea behind FUSE and how it interacts with Linux Virtual File System (VFS).

You are ready to jump into dadfs when your mum enters the room and asks you: "So what's wrong with the electricity, I thought you were supposed to work on dadfs?" As you look perplex your mother continues: "Yes, I heard something was wrong a fuse, is it fixed, can I what TV or their is a risk of blackout?" When you are done laughing your mum asks you a few questions:

1. What is a filesystem?
2. How is the Linux VFS working?
3. What is FUSE, and how does it interact with the VFS? Can you sketch it quickly to make it clearer?

As you patiently explain her the high level ideas without going into any technical detail, you realise that porting dadfs to FUSE would still be a pretty tough task. You do not really want to give up, but rather want some easier training, and maybe in a few weeks or months when things calm down at work you can come back to this task. In the meantime you try to find something easier to complete with FUSE. While you are lost in your thoughts you do not even realise that you are manipulating the mouse. Only when you see a window popping up you understand that you have clicked on an old icon. And here your are, connected to Lemonium Inc.!??!

It seems they forgot to shutdown your account and revoke your key... While you have nothing harmful in mind you navigate in the git tree and discover the existence of a hidden folder called `.del`. Surprisingly it belongs to Mr. Frown and is dated from the night of the crash! Has Mr. Frown anything to do with that crash? You remember that it was officially due to a high number of transactions, but this looks highly suspicious...

As you investigate the folder content you see three files: `Makefile`, `README`, and `.lemondbfs.c.swp`. The most recent one is the `README`, while the two others roughly date from the project creation. After having a quick look at the two main files you understand that Mr. Frown was working on a FUSE filesystem to access the lemonDB database, and this probably went wrong. Being in panic he tried to cover up his traces but obviously failed to completely erase them. You are so excited, you can have your revenge on Mr. Frown! All you need to do is to keep those evidences and recover the `lemondbfs.c` from `.lemondbfs.c.swp`. Then based on the original skeleton recovered from the swap file you decide to rewrite lemonDBfs and prove that he was responsible for the crash...

This is the perfect opportunity to kill two birds with one stone: you learn FUSE and prove that Mr. Frown is evil. Without any further ado you clone the `.del` folder, carefully read and understand Mr. Frown's plans as described in the README file, and start completing the `lemondbfs.c` file.