

SOFTENG 370 Tutorial 7

Timo van Veenendaal

29 September 2020

- 1 What is FUSE?
- 2 How does FUSE work?
- 3 How does the code from Assignment 2 work?

File systems, recap

- On Linux, all files are identifiable by their path from the root `/`. e.g. `/tmp/test`, `/home/timo/.bashrc`, ...
- Different paths can represent files from different places: a USB drive, the hard disk, a location on network...
- Each of these 'places' will have a corresponding filesystem implementation
- The Linux kernel's VFS (Virtual File System) resolves a path to a specific file system
- File systems are typically implemented as kernel modules, but this is hard
- Enter FUSE: FUSE is a kernel module which allows for file systems to be implemented in user space.

What happens when I run `ls` (on a FUSE filesystem)?

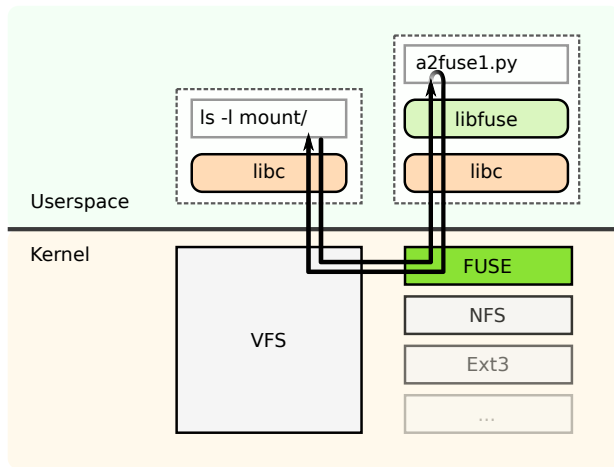
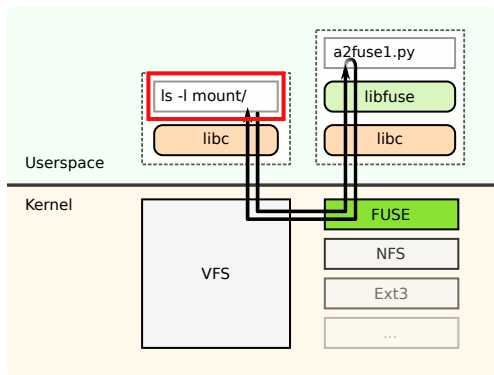


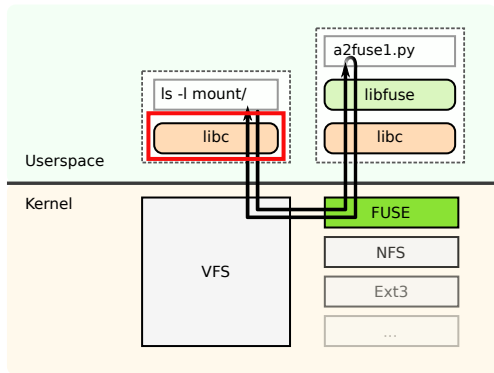
Figure: Original: CC-BY-SA 3.0, Wikipedia user Sven.

What happens when I run `ls` (on a FUSE filesystem)?



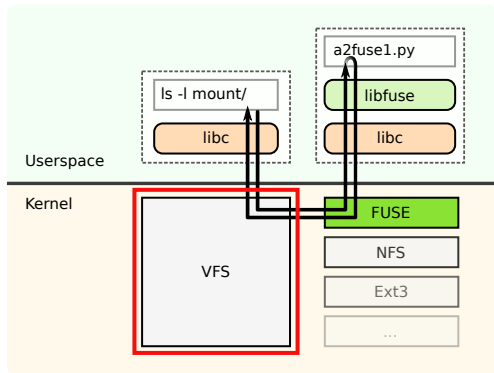
`ls` asks the C standard library (`libc`) what's in `mount/` using `readdir(3)`

What happens when I run `ls` (on a FUSE filesystem)?



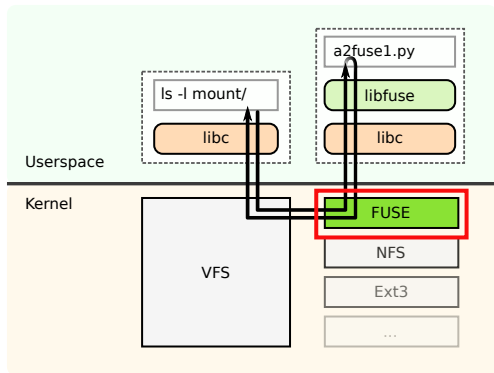
`libc` makes the right system calls to get this information. In particular, it uses the `readdir(2)` system call to find what files are in the directory. In doing this, it resolves the relative path provided by `ls` into an absolute path (starting with `/`).

What happens when I run `ls` (on a FUSE filesystem)?



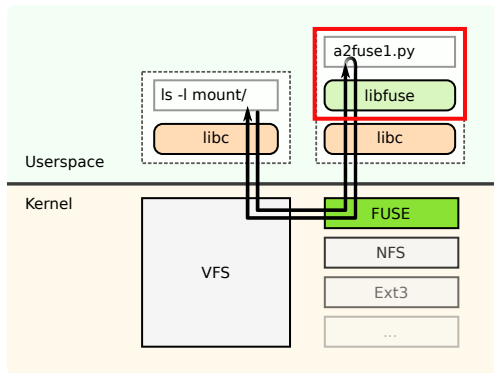
The VFS in the kernel determines what filesystem module needs to be used based on the absolute path provided. Since our FUSE filesystem is mounted at the `mount/` directory, the VFS knows it needs to go to the FUSE module. It forwards the `readdir` request to the module.

What happens when I run `ls` (on a FUSE filesystem)?



The FUSE kernel module sends a message to the user space `libfuse` library asking for it to deal with the `readdir` request (the exact detail of how this works is unimportant).

What happens when I run `ls` (on a FUSE filesystem)?



The `libfuse` library has a Python binding which calls the `readdir` function in your Python code. The result from the Python code is passed back through all the layers, which `ls` then uses to give its output.

Generalizing this

- This process applies for all the different filesystem operations that exist, including open, read, write, stat, and so on...
- You will have to implement a number of these operations for the assignment
- If unsure, look at how the two provided filesystems do it!

Questions?

Any questions?