

COMPUTER SYSTEMS

PROJECT 3

A versioning file system

1 Versioning

Normally, when a file is saved, the previous copy of the file is replaced by the newly updated copy. If a previous revision is desired, then either the user must manually make a copy, or use *version control software* (e.g., CVS, Subversion, git) to keep a history of previous versions. Meanwhile, *cloud storage services* (e.g., Google Drive, Dropbox), maintain some history of previous versions all on their own.

What we would like, however, is to have the preservation of multiple versions—a complete history of each file at each moment it was saved—performed by the computer on which we are working. Some systems have provided various forms of such *versioning file systems* (e.g., OpenVMS, LMFS). Therefore, our goal is to **create a virtual versioning file system** using FUSE.

2 Stuff to do

2.1 Getting some source code

Login to `vega.cs.amherst.edu`, make a directory for this project, and grab some source code and open it:

```
$ mkdir project-3
$ cd project-3
$ wget -nv -i https://goo.gl/Cb7Xh3
$ emacs versfs.c &
```

You will see that this file is quite like the beginning `amhfs.c` code from `.`. It provides the *passthrough file system* as a starting point for your new file system.

2.2 Your assignment

Create a versioning file system, `versfs`, that does the following:

1. No version of a file is lost. All versions are somehow represented in the storage directory. A *write* operation on a file adds a new version. Additionally, a *trunc* operation creates a new, shorter version of the file.
2. The files appear, in the mount-point, as normal files that, when *read*, it is the data from the most recent version that is provided.
3. A mechanism will be provided to read a specific version of a file, where the files are numbered from 1 to n (assuming n versions of the file have been created). Specifically, the 3rd version of the file `foo.txt` could be read by specifying the file name `foo.txt;3`.¹

3 How to submit your work

Submit your new, snazzy `versfs.c` file. You may use either of the following two methods to use the CS submission system:

- **Web-based:** Visit the submission system web page.
- **Command-line based:** Use the `cssubmit` command at the shell prompt on `remus/romulus`.

This assignment is due on Sunday, Dec-03, 11:59 pm.

¹Note that this case should be treated carefully. I could read version 3, but what if I then modify that version and then save it? What happens now? Is this a new and separate file, headed off in a different direction? Are previous versions from 4 to n kept, but no longer relevant to what version $n + 1$ represents? Do you just trust the user not to screw up and treat this older version as a continuation of the newer versions? Choose a behavior and stick with it.