Very Very Simple File System (VVSFS)

Jack Kilrain (u6940136) Daniel Herald (u7480080) Angus Atkinson (u7117106)

22 October 2023

Contents

1 Overview

TODO

2 Tasks Completed

TODO

3 Testing

- We created our own test suit (vvsfs/vvsfs_tests)
  - This was used to drive test driven design as we could create tests for expected behaviour and build new features.
  - Additionally we utilised it as a regression test suit to ensure that new code didn't break anything. And whenever we fixed problems that were discoved we built a test to ensure we didn't break it again.

  - It is composed of a set of helper scripts that provide automatic generation of a test environment, and an assertion framework to provide nice error messages.
- We used the pjdfstest to check our implentation for posix complience and various other edge cases. By the end we passed all tests with the following exceptions:
  1. The tests for large files (2gb) files.
  2. The filesystem does not keep track of the `.` & `..` files in directories as such we failed the test testing that folder link counts were incremented correctly. We chose to ignore this due to https://edstem.org/au/courses/12685/discussion/1633469.
  3. The filesystem does not correctly update ctime on truncate. (TODO: Does anyone want to fix this?)
  4. The filesystem does not store high presision time, only seconds like minix & ext2. (TODO: Does anyone want to fix this?)

4 Baseline

4.1 Unlink Dentries and Removing Directories

TODO

4.2 Renaming

TODO

4.3 Inode Attributes

We added support for storing `GID / UID / atime / ctime / mtime`. We acheived this by:

1. Adding the fields to the `vvsfs_inode` structure.
2. Loading the data within the `vvsfs_iget` method.
   - Following Minix / EXT2's lead we set the tv_nsec time to zero.
3. Syncing the data to disk within the `vvsfs_write_inode` method.
4. We chose to not implement `setattr / getattr` at this time since we didn't have anything meanful to change from the generic default function provided by the VFS.

Challenges implementing this feature:

1. During initial development it was discovered that the filesystem was somehow relying on the order of the inital fields in the `vvsfs_inode`. Instead of properly resolving this issue we decided to store the new fields at the end of the struct.
2. During testing it was discovered that the Linux kernel has measures to prevent disk trashing by not updating an inodes `atime` all the time. To override this and force the kernel to always update the times we added `strictatime` to our test mount script.

4.4 Supporting FS Stats

TODO

5 Advanced

5.1 Indirect Blocks

TODO

6 Extensions

6.1 Hardlinks and Symbolic Links

TODO

6.2 Special Devices

TODO