“Clean Up on Aisle fsconfig(2)!”

Aleksa Sarai
github.com/cyphar
2025-12-12

```
man 2 fsopen
man 2 fspick
man 2 fsconfig
man 2 fsmount
man 2 open_tree*
man 2 move_mount
```

Brand New!

Buy 0 Get 7 FREE!

Get Yours Today!

`FSCONFIG_SET_PATH{,_EMPTY}`

- Only one in-tree user (`ext4`’s `journal_path`).
- Very useful in principle, but `fs_lookup_param` needs work.
  - Do users want full `openat` behaviour or just `O_PATH`?
- Should we just make every path parameter accept it?
  - `FSCONFIG_SET_FD` needs some work too…

`fsconfig(2)` vs singletons

- `FSCONFIG_CMD_CREATE_EXCL` is a very big hammer.
- `get_tree_*` and `vfs_get_super` are mostly magic.
- Intended semantics of superblock reuse are (still) unclear.
    - We broke a few legacy mount APIs.
    - What if we provided information about this to `fc_log`?
    - (Classic `mount(2)` all over again…)

`fc_log` limitations

- Only 8 log entries before overflowing.
  - No priority, so info entries can overwrite fatal errors.
- Not to restart a flame war, but should we consider switching more things to `fc_log`?
  - util-linux worries around silently losing kmsg data…
  - What if we output unread messages to kmsg?

`FS_CONTEXT_FAILED`

- Is it always necessary for this to fail completely?
    - For “easy” failures it seems quite aggressive…
    - Even VFS permission errors can trigger it.
- Userspace needs to recreate the context to deal with errors.
    - Duplication of the context information in userspace.
    - Introspection API? (CRIU doesn’t C/R `fs_context`?)

Tag, you’re it!

東京 2025

LINUX PLUMBERS CONFERENCE

TOKYO, JAPAN / DECEMBER 11-13, 2025