

---

# “Clean Up on Aisle fsconfig(2)!”

CC-BY-SA 4.0



**Aleksa Sarai**  
[github.com/cyphar](https://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**



---

## 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?

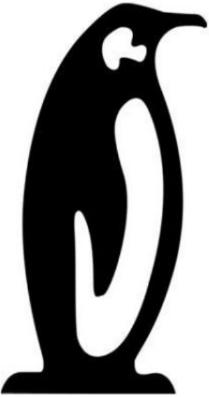
---

# 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

