

# Linux Kernel LFX Mentee Project: Patchwork Blog Report

Arnav Kapoor

September 6, 2025

## Contents

1	Introduction	2
2	Add Extra Version String to Makefile	2
3	Globally Enable Fall-Through Warning in Makefile	2
4	Trigger Kernel Panic for Testing	2
5	bcachefs: Avoid BUG When Journal Index Is Out of Bounds	3
6	Documentation: Fix Typo in proc.rst	3
7	mm: Add cond_resched() in do_zap_pte_range to Prevent RCU Stalls	3
8	ntfs3: Log Error When Cleanup in ni_rename Fails	3
9	openvswitch: Fix Static Key Underflow in ovs_dp_change Error Path	3
10	selftests: pidfd: Fix Grammar in Test Message	4
11	Documentation: Fix Typo 'proccess' to 'process'	4
12	team: Fix Sleeping Function Called from Invalid Context	4
13	ext4: Replace BUG_ON with Proper Error Handling in ext4_split_extent_at	4
14	arm64: KVM: Fix SError ESR Validation	5
15	hpfs: Fix Use-After-Free in hpfs_get_ea (Accepted Patch)	5
16	Conclusion	5

# 1 Introduction

This blog-style report documents the work completed during the Linux Kernel LFX Mentorship, focusing on kernel bug fixes, improvements, and contributions. Each section details a patch submitted, its motivation, technical summary, and impact on the kernel. The goal is to provide a professional yet readable account of the journey, challenges, and achievements.

## 2 Add Extra Version String to Makefile

**Patch:** 0001-Add-extra-version-string-custom1-to-Makefile.patch

**Summary:** This patch customizes the kernel build by changing the extra version string in the Makefile from `-rc6` to `-custom1`. This is useful for identifying custom builds and tracking changes during development.

**Technical Details:** The patch modifies the `EXTRAVERSION` variable in the top-level Makefile, ensuring that the kernel version string reflects the custom build.

**Impact:** This change helps distinguish custom kernels from official releases, aiding in debugging and deployment.

## 3 Globally Enable Fall-Through Warning in Makefile

**Patch:** 0001-Makefile-Globally-enable-fall-through-warning.patch

**Summary:** This patch enables compiler warnings for implicit switch-case fall-throughs globally in the kernel build system. It also updates documentation to mark implicit fall-through as deprecated.

**Technical Details:** The Makefile is updated to pass the relevant warning flag to the compiler. Documentation is improved to guide developers on best practices and to avoid ambiguous code.

**Impact:** By enforcing this warning, the kernel codebase becomes more robust and less prone to subtle bugs caused by missing break statements in switch cases.

## 4 Trigger Kernel Panic for Testing

**Patch:** 0001-Trigger-kernel-panic-for-testing.patch

**Summary:** This patch introduces a module or mechanism to deliberately trigger a kernel panic for testing purposes. It is useful for validating panic handling and crash recovery workflows.

**Technical Details:** The patch provides a simple way to invoke a panic, typically via a custom kernel module or sysfs interface.

**Impact:** Enables developers and testers to simulate kernel panics, ensuring that monitoring and recovery systems respond correctly.

## 5 bcachefs: Avoid BUG When Journal Index Is Out of Bounds

Patch: 0001-bcachefs-avoid-BUG-when-journal-index-is-out-of-bounds.patch

Summary: This patch replaces a critical `BUG_ON` assertion with a proper bounds check in the `bcachefs` journal.

Technical Details: The change ensures that if `*idx > keys->nr`, an error is logged and the function returns `NULL`, preventing a kernel panic.

Impact: Improves kernel robustness by handling out-of-bounds errors gracefully, preventing crashes and aiding in debugging.

## 6 Documentation: Fix Typo in proc.rst

Patch: 0001-docs-proc.rst-fix-typo-proccess-to-process.patch

Summary: This patch corrects a minor typo in the `proc` filesystem documentation, improving clarity and professionalism.

Technical Details: Changes `'help compiling'` to `'helping to compile'` in `proc.rst`.

Impact: Enhances documentation quality for kernel developers and users.

## 7 mm: Add `cond_resched()` in `do_zap_pte_range` to Prevent RCU Stalls

Patch: 0001-mm-add-cond\_resched-in-do\_zap\_pte\_range-to-prevent-R.patch

Summary: This patch adds calls to `cond_resched()` in memory management code to prevent CPU stalls during large unmap operations.

Technical Details: By yielding the CPU periodically, the patch avoids soft lockups and RCU stall warnings under heavy memory pressure.

Impact: Improves system stability and responsiveness during memory-intensive operations.

## 8 ntfs3: Log Error When Cleanup in `ni_rename` Fails

Patch: 0001-ntfs3-log-error-when-cleanup-in-ni\_rename-fails.patch

Summary: This patch adds error logging when cleanup operations fail in the NTFS3 rename function, making failures visible in kernel logs.

Technical Details: If both `ni_remove_name` and its undo fail, an error is logged and a bad state is flagged.

Impact: Improves error visibility and aids in diagnosing filesystem issues.

## 9 openvswitch: Fix Static Key Underflow in `ovs_dp_change` Error Path

Patch: 0001-openvswitch-Fix-static-key-underflow-in-ovs\_dp\_change.patch

Summary: This patch fixes a static key underflow bug in the Open vSwitch datapath error handling. The bug could cause reference count underflows and warnings during cleanup.

Technical Details: The assignment of `dp->user_features` is moved after all operations that can fail, ensuring the datapath state is only updated when all steps succeed.

Impact: Prevents kernel warnings and potential instability in Open vSwitch deployments.

## 10 selftests: pidfd: Fix Grammar in Test Message

Patch: 0001-selftests-pidfd-fix-grammar-in-test-message.patch

Summary: This patch corrects the grammar in a test result message in the pidfd selftests, improving clarity and professionalism.

Technical Details: Changes 'passed' to 'Passed' in the test output string.

Impact: Enhances the quality of kernel selftest output.

## 11 Documentation: Fix Typo 'proccess' to 'process'

Patch: 0002-docs-proc.rst-fix-typo-proccess-to-process.patch

Summary: This patch fixes a typo in the proc filesystem documentation, changing 'proccess' to 'process'.

Technical Details: Simple string replacement in `proc.rst`.

Impact: Improves documentation accuracy.

## 12 team: Fix Sleeping Function Called from Invalid Context

Patch: 0002-team-Fix-sleeping-function-called-from-invalid-context.patch

Summary: This patch addresses a bug where sleeping functions were called from atomic context in the team network driver, which could lead to kernel warnings and instability.

Technical Details: Introduces a work queue to defer RX flags changes to process context, ensuring mutex-acquiring functions are called safely.

Impact: Improves driver reliability and prevents context-related kernel warnings.

## 13 ext4: Replace BUG\_ON with Proper Error Handling in ext4\_split\_extent\_at

Patch: 0004-ext4-replace-BUG\_ON-with-proper-error-handling-in-ex.patch

Summary: This patch replaces critical `BUG_ON` assertions with warnings and error returns in the ext4 filesystem's extent splitting code, allowing the kernel to handle invalid conditions gracefully.

Technical Details: Invalid split positions and flags now trigger warnings and return error pointers instead of crashing the kernel.

Impact: Improves filesystem robustness and error reporting, reducing the risk of kernel panics.

## 14 arm64: KVM: Fix SError ESR Validation

Patch: `fix_arm64_kvm_serror_esr.patch`

Summary: This patch corrects the validation logic for SError ESR in KVM on arm64, ensuring only the ISS field is user-controllable and preventing userspace from setting reserved bits.

Technical Details: The validation now only allows bits 23:0, rejecting attempts to set the ISV bit or other reserved bits.

Impact: Prevents security and stability issues in virtualized arm64 environments.

## 15 hpfs: Fix Use-After-Free in hpfs\_get\_ea (Accepted Patch)

Patch: `accepted_patches/hpfs-use-after-free-fix-6.16-final.patch`

Summary: This accepted patch fixes a use-after-free vulnerability in the HPFS filesystem's extended attribute handling, preventing crashes and potential security issues.

Technical Details: Adds comprehensive bounds checking in EA iteration loops, validating structure and buffer boundaries before accessing memory.

Impact: Resolves a long-standing bug, improving filesystem safety and reliability. This patch was accepted upstream.

## 16 LinkedIn Experience Brief

During my Linux Kernel LFX Mentorship, I contributed to the upstream Linux kernel by developing and submitting multiple patches addressing bugs, improving documentation, and enhancing kernel robustness. My work spanned filesystems (ext4, ntfs3, hpfs), memory management, networking (Open vSwitch, team driver), and kernel build infrastructure. I authored fixes for use-after-free vulnerabilities, improved error handling, and added developer-friendly features. My accepted patches and active participation in the kernel community reflect my commitment to open source and technical excellence. This experience strengthened my skills in C, git, kernel debugging, and collaborative development, preparing me for impactful roles in systems programming and open source leadership.

## 17 Conclusion

Throughout the Linux Kernel LFX Mentorship, a wide range of kernel bugs and improvements were addressed. From documentation fixes and test improvements to critical bug fixes in filesystems, memory management, and drivers, each

patch contributed to a more robust and maintainable kernel. The experience provided deep insights into kernel development, patch submission, and open source collaboration. Accepted patches and upstream contributions highlight the impact and value of this work. The journey continues, with ongoing learning and contributions to the Linux kernel community.