

# Linux Kernel on Reddit

- [Help regarding bugzilla](#) (2026/06/21 07:28)  
hello i was looking at the bugzilla and looking for beginner friendly tags there and couldn't find one. Apart from that I found a bug which i am interested in solving but I don't know if it has been solved or assigned to someone else how do I figure that out? submitted by /u/lzumi994 [link] [comments]
- [Hello everyone, I'm interested in learning Linux kernel development and would like to know the best steps to get started. Could you guide me to the right resources or share tips that helped you at the beginning of your journey?](#) (2026/06/21 02:17)  
submitted by /u/Ahmed\_cs [link] [comments]
- [connect\(2\) syscall errno=74 \(EBADMSG\) on Android?](#) (2026/06/19 20:05)  
EDIT: Solved! details in my comment PoC Source: 

```
#include <stdio.h> #include <string.h> #include <errno.h> #include <unistd.h> #include <arpa/inet.h> int main(void) { struct sockaddr_in a = { .sin_family = AF_INET, .sin_port = htons(5050), .sin_addr = { .s_addr = inet_addr("127.0.0.1") } }; int s = socket(AF_INET, SOCK_STREAM, 0); if (s < 0) { perror("socket"); return 1; } if (connect(s, (struct sockaddr*)&a, sizeof a) < 0) { printf("connect failed: %s (errno=%d)\n", strerror(errno), errno); return 1; } puts("connected"); return 0; }
```

 From adb shell: 

```
$ su # cd /data # ./test2 connect failed: Not a data message (errno=74) # uname -a Linux localhost 4.19.110-android-x86_64-g066cc1d #1 SMP PREEMPT Wed Mar 25 11:32:46 CST 2020 x86_64
```

 It doesn't matter if something is listening on tcp://localhost:5050 or not, output is the same. I am running inside QEMU, Android x86 9 (amd64). The application itself is aarch64 (running through android native bridge) Problem I've never seen this errno before and I can't seem to figure it out by myself. I tried: Zero-initializing sockaddr\_in and initializing field by field Using IPPROTO\_TCP instead of 0 as third argument for socket(2) Using other addresses I can ping from environment (10.0.2.2 for PC host and 10.0.2.15 for QEMU localhost) Setting socket to blocking/nonblocking mode with ioctl(2) via FIONBIO and fcntl(2) Snippet works perfectly on GLIBC/MUSL-based GNU/Linux systems: as expected errno is ECONNREFUSED Any help will be appreciated, thank you submitted by /u/Capable-Cap9745 [link] [comments]
- [My static analysis tool now supports compile\\_commands.json for linux kernel](#) (2026/06/18 13:21)  
submitted by /u/Choice\_Bid1691 [link] [comments]
- [Faking bits](#) (2026/06/17 17:51)  
I was in need to verify CoreFreq with new CPUID leaves I don't have. Patching QEMU source code <https://discord.com/channels/1422511936842502166/1422512302858440704/1515654629789274152> submitted by /u/Cyrlng [link] [comments]
- [Why is qemu an emulator and not a simulator?](#) (2026/06/17 17:27)  
I searched online for a precise definition for both emulation and simulation but could not find anything that clearly distinguishes emulation from simulation. submitted by /u/dezerev [link] [comments]
- [almost got it but egl stopped me again](#) (2026/06/11 08:59)  
submitted by /u/Senior-Question693 [link] [comments]

- [How can I transfer a structure to kernel module then store it?](#) (2026/06/08 17:04)

Hey everyone, I created a lightweight firewall with C, created a kernel module with the netfilter API, and a pre-routing hook. Now I want to send rules via netlink socket. My idea is to create a structure, then send it. But I cannot find the best way to store all rules in the kernel, then use them in a hook. Sometimes I think I can compress the rules into bits, then send them. If anyone has experience with my problem, please help me understand how I can implement a optimize protocol and store it in the kernel module submitted by /u/Yousef\_Tele [link] [comments]

- [Kernel dev setup advice](#) (2026/06/05 21:14)

I am new to kernel development, I am having trouble building my setup I am unable to decide if I should use my host machine for development and qemu for testing OR use a separate VM all together like Multipass or Virtualbox. What is the standard/professional setup. submitted by /u/dezerev [link] [comments]

- [Kernel and Programming Topics to focus on for Linux Kernel Engineer role](#) (2026/06/04 23:25)

Hi everyone, I'm preparing for a Linux kernel/systems role and would appreciate suggestions on key topics to focus on. My background is in Linux kernel work including character device driver development, QEMU-based kernel testing, board bring-up, and an upstream patch accepted into LKML. I'm trying to go deeper into kernel internals and would be grateful for recommendations on important areas like memory management, scheduling, concurrency, driver architecture, and any resources or mental models that help think like a kernel developer. submitted by /u/PlayfulKnowledge2788 [link] [comments]

- [Making sure a mtd is gone before unregistering a spi-nor device?](#) (2026/06/03 18:38)

Hi guys, I am currently in the process writing a spi controller driver for some custom hardware. In the driver, I register my spi-controller and a spi-device. For that I am using spi-nor with a mtd table for the storage I want to show up. So far so well. Now for the tear down, I do it in reverse, unregister the spi\_device and then the spi\_controller. I test my kernel driver everything works, I can access my storage, I can mount it, when the driver gets unloaded the mount vanishes as expected, everything is good. Now, I wanted to automate the mount and unmount via udev rules. So I add systemd mount files and udev rules to trigger them. Mounting works just fine, unmounting results in a kernel crash. Apparently the unmount is still in progress while the spi\_device gets already deallocated resulting in a null pointer dereference. I have no clue how to fix that. How do I delay my device unregister until the mtd subsystem is done? Thanks in advance for any pointers! submitted by /u/foobar93 [link] [comments]

- [How do I apply this kind of patch to my kernel?](#) (2026/06/03 10:01)

I want to add Simple LMK (specifically linux-4.19 branch) to my android kernel source, but that repo seems like a full kernel source rather than a patch or a mailbox. How can I add it to my kernel source? Solved Apparently I had to cherry pick all the commits related to Simple LMK from beginning to end. I went to the commit history and found the first commit related to simple LMK was 58ecd97 (before that, the normal kernel commits began) and the most recent commit is 77d6486. Now the commit hashed used by github is shortened, so you need to use git rev-parse to get the full one. Fetch the repo: `git fetch https://github.com/kerneltoast/simple_lmk linux-4.19` Rev-parse: `$ git rev-parse 58ecd97`  
`58ecd977a2866dbd2010a89e4077a9ceae076931` `$ git rev-parse 77d6486` `77d64863e6f56c3acf522d43a2f90aa67e9785c4` Now just run cherry pick for this commit range. `git cherry-pick 58ecd977a2866dbd2010a89e4077a9ceae076931^..77d64863e6f56c3acf522d43a2f90aa67e9785c4`  
There will only be a few conflicts now (if any), which you can easily solve. submitted by /u/IsHacker003 [link] [comments]

- [ACPI table dump for Asus Zenbook A16 \(Snapdragon X2 Elite Extreme\)](#) (2026/06/03 05:56)

I have successfully extracted the ACPI table dump from my Asus Zenbook A16. The full binary set is available here:

[https://drive.google.com/file/d/1lwYydyrnHOrItThc1TWbGePFIRxGumV-/view?usp=drive\\_link](https://drive.google.com/file/d/1lwYydyrnHOrItThc1TWbGePFIRxGumV-/view?usp=drive_link) This data serves as the firmware roadmap for the system and should help identify the necessary configurations for better hardware compatibility under Linux. submitted by /u/Putrid\_Draft378 [link] [comments]

- [Built PulseBook, a low-latency C++20 trading engine using DPDK Ring PMD, fixed-size Ethernet protocols, L2 order book, imbalance strategy, and inline risk checks.](#) (2026/06/02 20:27)

Achieved ~111ns median and ~550ns p99 virtual RX-to-TX latency over 1M events with zero failures on my laptop. Next improvements: Real NIC + VFIO benchmark AF\_XDP/io\_uring comparison Multi-core matching engine Hardware timestamping NASDAQ ITCH replay support As a student systems/HFT project, is this actually impressive for backend/low-latency roles?

<https://preview.redd.it/dsk0v9anix4h1.png?width=740&format=png&auto=webp&s=c8ccfd856ca382fa7912bdb03e40d7eb9fa9b0f5> submitted by /u/Federal\_Tackle3053 [link] [comments]

- [ALGUN KERNEL PERSONALIZADOS PARA OVERCLOCK EN MI MOTOROLA G 5G 2022?](#) (2026/06/01 03:47)

submitted by /u/xeromk1122 [link] [comments]

- [Enabling the mainline rkvddec \(4K H.264/HEVC\) hardware decoder on Orange Pi 5 Plus](#) (2026/05/31 23:03)

submitted by /u/That\_Direction3907 [link] [comments]

- [C project](#) (2026/05/30 00:08)

Give me some projects , to go deeply into c programming systems , so i can with this project learn a lot of things in linux . submitted by /u/cdtrmnbaell [link] [comments]

- [Graphical issue on the newest kernel](#) (2026/05/29 18:01)

submitted by /u/gwyddbwyll [link] [comments]

- [GKH discusses impact of Rust on Linux](#) (2026/05/29 10:34)

<https://www.youtube.com/watch?v=0vhGWcIF7LU&t=810s> submitted by /u/killjoy\_buzzkill [link] [comments]

- [Question: UIO without device tree](#) (2026/05/28 21:09)

I have some experience with kernel modules and drivers, however everything I did was on device tree based platforms, not ACPI. Now for a custom IO device I wanted to use UIO. However I can't figure out how to get the kernel to generate the device nodes without a device tree entry. Is there a trick I missed or do I have to implement custom kernel modules? submitted by /u/ukezi [link] [comments]

- [Linux-Koltin](#) (2026/05/26 06:23)

Hey everybody So I've been doing an experiment in operating systems. I'm trying to make a Linux kernel environment where you can develop programs using Kotlin. This is a completely sandboxed environment. It does not require any other programs to run. When the Linux kernel starts up, it usually hands control to a program that helps get things going. This program contains a lot of C code and bash scripts. I wanted to see if I could kill all of that and only use Kotlin. Instead of a standard root filesystem, I wrote a Kotlin program and compiled it ahead-of-time into a statically linked linux\_x64 binary using Kotlin/Native. By passing `init=/init.keke` in the kernel boot parameters (via QEMU), the Linux kernel hands control directly to the Kotlin executable as PID 1. From there, Kotlin is completely in charge of the system lifecycle: Filesystem Mounts: Using `kotlinx.cinterop`, the Kotlin script natively executes raw POSIX syscalls to mount `/proc`, `/sys`, `/dev`, and creates a tmpfs RAM disk over `/tmp` (which

Java/Gradle requires to unpack JNI libraries). Network Stack: Because we bypassed standard networking daemons, the Kotlin init process has to manually fork and configure the loopback interface (lo) and the ethernet interface (eth0), assign static IP routes, and securely bind-mount a custom /tmp/resolv.conf over the host's DNS configuration to establish internet connectivity. ( QEMU ETHERNET ONLY ) The Build Pipeline: The repository acts as its own root filesystem via a virtio-9p passthrough. We embedded a standalone OpenJDK and the Android SDK directly into the kernel tree. Once the Kotlin init process stabilizes the network and mounts the filesystems, it dynamically injects the environment variables (JAVA\_HOME, ANDROID\_USER\_HOME) and forks a child process to launch the Gradle Build Daemon. The system successfully resolves dependencies from Maven/Google, orchestrates the build cache, and compiles a native Android application (kernel.kotlin.system) directly from the Linux boot loop. If there is no ethernet the build fails and you continue on without kernel panics! Also it comes with a package manager at kernel level! When you boot up and have internet/ethernet access just run kotlib sync! It's completely standalone, bypasses standard Linux userspace utilities entirely, and proves that Kotlin/Native is robust enough to handle low-level POSIX environment orchestrations. submitted by /u/Ok\_Sky3062 [link] [comments]

- [Kernel Dev Roadmap](#) (2026/05/26 06:22)

Hi there, As of right now i am a backend dev with java for about 2 years of experience. Recently i learned Os and Computer Architecture as a subject in college and i liked it. I want to learn more of it, and i want to explore Kernel Dev, this is what i have researched and came up, that i can go in this field. so what i am asking is -> If anyone can help me with the roadmap and can guide me too. I want guidance on should i really go into this field or not, and i mean i wont be getting job just after college right, so i will be pursuing market with my Backend + Devops (current skill set) and side by side learning it. or do i need to do master for it too, i can afford, and i mean if it is necessary that is. And then again overall roadmap, please. Thankyou submitted by /u/Be\_akshat [link] [comments]

- [Problema](#) (2026/05/26 04:13)

submitted by /u/ReturnNumerous4553 [link] [comments]

- [Question: Kernel module that provides interface that returns an incrementing number.](#) (2026/05/24 12:31)

I am currently ramping up on Linux kernel module development and thought that I would start with something small. For our iceorxy2 project, we need an interface from which every process that uses it can acquire a number. It could be just an atomic u64 that increments with every call. It is just important that this is guaranteed to be unique. This could be simply an atomic in shared memory but then other processes could fiddle around with it. I implemented this by providing a proc entry /proc/atomic\_counter and cat /proc/atomic\_counter prints that incrementing number. A character device approach would also be possible. Is there a preferred way? Or any recommendations? But I failed to implement this in Rust, it seems that kernel::bindings do not yet provide proc\_create , or am I mistaken? What I was also wondering is, how to test such an interface idiomatically? It is just a simple counter but lets assume I have a complex thing in there and would like to have an extensive test suite. My idea was to extract all logic in a separate lib/crate, test it and keep the actual module as simple as possible. submitted by /u/elfenpiff [link] [comments]

- [Struggling with PID1 + Chain-of-Trust Boot Flow \(Custom Runtime OS Project\)](#) (2026/05/23 00:15)

Hey folks, I'm building a small experimental OS/runtime hybrid and I've hit a wall with PID1 behavior and the chain-of-trust during early boot. Hoping someone here has fought similar dragons. Context I'm not building a traditional Linux distro. This is a governed runtime with its own mediation layer, identity checks, and a compatibility membrane for foreign binaries. PID1 is extremely minimal — it's basically: initialize the

invariant engine mount the pattern ledger bring up the mediation layer hand off to the user-level runtime No systemd, no BusyBox init, nothing fancy. The Problem When the system boots, the firmware verifies the shim → kernel → initrd just fine. But once my custom PID1 takes over, the chain-of-trust becomes fragile: PID1 sometimes fails to verify its own signature Even though the binary is signed and measured correctly, the verification step occasionally returns “unreadable” or “missing measurement.” Ledger mount timing issues The pattern ledger (think: structured state log) sometimes mounts after PID1 tries to validate it, causing a soft-fail that cascades. PID1 is too fragile Any hiccup in the trust chain causes PID1 to panic instead of gracefully retrying or falling back. Firmware vs runtime identity mismatch The firmware expects a static identity, but the runtime uses a dynamic identity model (based on behavior + signature). They don’t always agree. What I’ve Tried Delaying ledger mount Moving signature verification earlier Moving signature verification later Rebuilding PID1 to be even smaller Re-signing the entire boot chain Re-measuring the initrd Rebuilding the shim Re-generating the root key Still getting intermittent failures. What I’m Looking For Anyone who has experience with: custom PID1 implementations minimal init systems secure boot chains measured boot TPM-based identity checks early-boot race conditions I’m not trying to reinvent Linux — this is a research OS with a very different runtime model. I just need PID1 to stop collapsing the entire trust chain every time one measurement is late or unreadable. Any advice, patterns, or “don’t do this, do that instead” would be hugely appreciated. submitted by /u/Objective\_Ad5748 [link] [comments]

From:

<https://wiki.tromjaro.alexio.tf/> - **TROMjaro wiki**

Permanent link:

<https://wiki.tromjaro.alexio.tf/doku.php?id=news:reddit:kernel&rev=1584393415>

Last update: **2021/10/30 11:38**

