

Linux Kernel on Reddit

- [Which kernel has the best support for Navi 14 GPU](#) (2026/01/29 06:26)

I'm currently trying to get my Navi 14 GPU(AMD Radeon Pro 5300) working perfectly without any major issues Right now it won't turn on first boot with the stock kernel provided by arch. So I decided to compile one for myself. So what kernel version should I use and what should I set in the .config (or make nconfig)? submitted by /u/Striking-Flower-4115 [link] [comments]

- [Where can I find remote work related to the kernel?](#) (2026/01/23 06:06)

submitted by /u/Infinite-Feed-3904 [link] [comments]

- [Linux Device Drivers, I/O Port Operations in VM](#) (2026/01/21 17:03)

I'm trying to get into Linux kernel module development and saw lots of people recommending starting out with 'Linux Device Drivers Third Edition'. I have a custom kernel loaded in a VM (VirtualBox) on my laptop. I'm currently on chapter 9, "Communicating with Hardware", and it deals with I/O Port operations and hardware memory barriers; the book's a bit dated and suggests using a PC's built-in parallel port (or a printer, I think it said) for testing. The next chapter expands off of this one, and I would like to follow along and test the code before continuing, but I'm not sure how. The closest thing I could find to use was the Adafruit FT232H Breakout — I suppose I would connect LEDs to the GPIO pins with jumper cables and use USB passthrough— would this work and allow me to follow along with the examples in these chapters? submitted by /u/entheo6 [link] [comments]

- [LKM Rootkit Singularity vs eBPF security tools - Sophisticated Linux Malware](#) (2026/01/21 16:20)

submitted by /u/Worldly-Fruit5174 [link] [comments]

- [Confused between embedded systems vs Linux kernel path, looking for grounded advice, not hype.](#) (2026/01/21 10:03)

Hey folks, I'm early in my career and trying to make a sensible decision about how to get into Linux kernel / low-level systems work long term (drivers, OS internals, later virtualisation and hypervisors) I keep seeing two opposing pieces of advice: "Jump straight into kernel development" "Start with embedded / firmware to build fundamentals" What's confusing is that these often get framed as completely different career paths. Right now I'm leaning toward: Bare-metal embedded (MCU, no OS) Then firmware / RTOS Then embedded Linux bring-up Then drivers / kernel work The idea is that embedded isn't the goal, but a foundation so things like memory, interrupts, boot, and concurrency aren't abstract later. My doubts: Is this a solid way to build toward kernel roles? Or am I just delaying real kernel experience unnecessarily? I'm not chasing quick titles, I care more about building real understanding over time. Would really appreciate hearing from people who've actually worked in embedded or kernel roles: How did you start? What would you change in hindsight? Thanks. submitted by /u/KernelLicker [link] [comments]

- [I wanna experiment with building rootkits](#) (2026/01/20 17:35)

I've been studying linux kernel programming for about a month now for school, and recently I got interested in rootkits. I then thought „would be a cool experiment to try building a simple rootkit“. Disclaimer: I don't want to deploy it anywhere, I just wanna have some fun learning about the linux kernel. So, what I wanna build is a kernel module or driver that „intercepts“ write system calls and modifies the buffer under certain conditions . It checks if the file descriptor is a terminal (I don't want to change the buffer sent to a driver or something), and if true check if the

first n bytes of the buffer equals some other buffer, and if that is true modify those n bytes to something else. So what do I need help with? I just wanted to know what knowledge should I seek, what to research, so I can build it. submitted by /u/Avivush2001 [link] [comments]

- [Average life-span of Linux kernel bugs](#) (2026/01/20 11:04)

Source: <https://pebblebed.com/blog/kernel-bugs> submitted by /u/killjoy_buzzkill [link] [comments]

- [\[i.MX8MM / Yocto\] PCIe Realtek NIC powers down during suspend \(WOL fails\) - Link LEDs die](#) (2026/01/19 05:46)
submitted by /u/Alternative-Bake-131 [link] [comments]

- [Need advice: Firmware vs Kernel dev for high-paying career \(ECE undergrad\)](#) (2026/01/18 17:39)

submitted by /u/JustAnotherHuman0007 [link] [comments]

- [Dell Pro 13 Premium freezing with CATERR](#) (2026/01/14 18:01)

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

- [Should I get into kernel Development ?](#) (2026/01/10 06:33)

Greetings folks, I do hustle with my studies and I aspire to create a startup. But I also want to contribute to OSS. Kernel has high technical bar. Now with AI I am worried a lot, I don't know if actually LLMs or any other architecture manages to outpace the devs (without slops). It makes me a bit sad to be honest... I am really into Engineering but I am really worried, and bills won't pay themselves. My questions are Is it worth it starting now? (I actually want to hear maintainers with really good contribution, their feedback) What is the hardest problem in linux kernel that poses open challenge lately or even long before that maybe I can take a look at. Something challenging not something easy... Or just go for that blue color job after all ? submitted by /u/InterviewMediocre879 [link] [comments]

- [How welcome are newbies sending mail to the linux mailing list \(bluetooth in particular \)?](#) (2026/01/06 12:01)

For context, I want to get the attention of the bluez team. I've submitted an issue on their official github page but looking through the recently opened issues, they don't seem that much active on it. There's certainly work being done on the repo though. I want to know whether a PR (or anything really at this point) will be welcome improving their documentation situation. I've wanted to look at bluez documentation, while the repo does have .rst files in a nicely labeled doc folder, They are not organised, nor easily accesible (one has to go to the repo itself and look through the github previews). I've forked their repo and converted all (almost all) the .rst files to markdown and created a github page with the docs and I feel like this could be a good addition to the project submitted by /u/Striking-Storm-6092 [link] [comments]

- [How to compile a minimal functional kernel that uses least amount of ram possible?](#) (2026/01/05 18:40)

How to i achieve it, i need it to be below 50 mb of ram? Any suggestions? submitted by /u/GuiltyVisit9119 [link] [comments]

- [Every server at Meta runs eBPF, 50% over 180 programs](#) (2025/12/31 16:53)

submitted by /u/xmull1gan [link] [comments]

- [Why can't I get my iX86 initramfs to see block devices?](#) (2025/12/30 19:56)

Hello. I compiled stable kernel for x86. I used old Debian config from when Debian used to have support for i386, and this failed to boot. I then used i386_defconfig from already inside the kernel and built with that, but the initramfs didn't recognise any block devices when I ran "blkid" and "fdisk -l". I tried on two separate machines and it didn't find block devices. I tried USB boot and from CD. It didn't find them. I also tried in QEMU, and it didn't find block devices there either. I then tried using alpine linux Its defconfig. And the same happened. It didn't find the block devices. I tried packing the initramfs using busybox and using Debian's update-initramfs tool. I tried packing modules manually and running "modprobe

ahci" and "modprobe usb-storage". It just didn't want to go and didn't find the block devices. Is this typical behavior for ix86 kernel? Is there some bug I don't know about? Has the code rotted? Or is there something I am missing? Thanks EDIT: I finally managed to get block devices found. I had to run modprobe on all the drivers within the init script and copy the required modules/firmware/drivers to the initrd too. submitted by /u/Own-Pattern-7862 [link] [comments]

- [Booting on a e-core](#) (2025/12/24 17:07)

I have a 12th Gen Intel(R) Core(TM) i7-12700H with 12 threads on p-cores and 8 on e-cores. From time to time, for instance on battery, I'd like to completely shut down the p-cores. But the system always boot with cpu 0 being a p-core, making that impossible afaict. Is there a way to ensure that cpu #0 is a e-core? submitted by /u/galibert [link] [comments]

- [Questions about new mount api](#) (2025/12/19 17:02)

AT_EMPTY_PATH If pathname is an empty string, operate on the file referred to by dirfd (which may have been obtained from open(2) with O_PATH, from fsmount(2) or from another open_tree()). If dirfd is AT_FDCWD, the call operates on the current working directory. In this case, dirfd can refer to any type of file, not just a directory. This flag is Linux-specific; define _GNU_SOURCE to obtain its definition. Func in question is open_tree Does that mean that dirfd can't be a file if it is not AT_FDCWD? So it isn't possible to bind mount a file using fds in the new api? Additionally must it be `open` or can it also be `openat`? submitted by /u/AdventurousFly4909 [link] [comments]

- [Need help with compiling](#) (2025/12/18 22:44)

1) make is building all the unnecessary drivers for no reason. How do I fix this? 2) What should I do to optimise kernel for gaming? Currently running a HP Notebook 14 i3 Tiger Lake I don't have much experience other than compiling a 5.11.x kernel (Successfully failed) I'm currently on Ubuntu. Not sure if my distro has anything to do with building a kernel submitted by /u/Striking-Flower-4115 [link] [comments]

- [Is it possible to replace GNU Make \(Kbuild\) with another build system?](#) (2025/12/18 03:43)

I've been diving into kernel building for several weeks, and I'm wondering if it's possible to replace Kbuild with another build system? Like CMake or Meson? submitted by /u/Summer_1228 [link] [comments]

- [PSA: When making a kernel module makefile it must be capitalized as Makefile](#) (2025/12/16 19:55)

Hello everyone, I was writing my first kernel module and kept running into an error with kernel-headers/scripts/Makfile.build running into an include error on line 41 and couldn't find any info on this whatsoever online, so I figured I should post my solution in case anyone runs into the same issue. Basically, your module makefile must be capitalized as Makefile (not makefile or MakeFile) because the kernel module build system is hard coded to look for either a "KBuild" file or "Makefile" in your source directory and doesn't check for different capitalizations. So, in case anyone else has this issue the error is in Makefile.build line 41: no such file or directory. Just rename your makefile or MakeFile to Makefile and that should fix it. Edit: For those saying makefiles are always capitalized that is incorrect, make commands will work just fine with lowercase, that being said, it was a mistake for me to say MakeFile, not that I've actually tested it. I usually use lowercase because my editor (zed) only shows the correct icon with lowercase makefiles (it shows a generic text file icon otherwise). Also, could you please direct me to the docs where it says Makefile should be capitalized as I didn't see this mentioned anywhere in the docs. Thanks. submitted by /u/NoahNoah011 [link] [comments]

- [I'll rephrase the question.](#) (2025/12/14 22:16)

I'll rephrase the question. Is there anyone competent in the Linux kernel, not just the basics, but the very deep workings of Linux? Specifically, how it routes incoming and outgoing network requests. When I say deep, I mean memory addresses. Binary. Network company, network card

assembler submitted by /u/coder-true [link] [comments]

- [looking for kernel devs, competitive salary.](#) (2025/12/13 21:00)

looking for kernel devs to bring on for a project, offering competitive salary. message me if interested submitted by /u/Yavimaya401 [link] [comments]

- [Is it possible to use DMA like only input output system for peripheral device?](#) (2025/12/13 19:58)

for peripheral device? I answered: "no, because we need to initialize device, get information about the area of memory it can use for DMA". I was answered that, there is possible to use default memory such as circle buffer and it's possible and there is another reason why we need PMIO and MMIO in addition to DMA. Any ideas? submitted by /u/Conscious_Buddy1338 [link] [comments]

- [Is it secure to use this kernel ?](#) (2025/12/13 11:23)

I get some errors with latest kernel-longterm (6.12.61-200.fc42.x86_64 #1 SMP PREEMPT_DYNAMIC Sun Dec 7 11:59:15 UTC 2025): journalctl -r --priority=err Dec 09 13:31:18 maketopsite kernel: rtw89_8852ce 0000:62:00.0: [ERR]fw PC = 0x2013e75e Dec 09 13:31:18 maketopsite kernel: rtw89_8852ce 0000:62:00.0: [ERR]fw PC = 0x2013e76e Dec 09 13:31:18 maketopsite kernel: rtw89_8852ce 0000:62:00.0: [ERR]fw PC = 0x2013e76e Dec 09 13:31:18 maketopsite kernel: rtw89_8852ce 0000:62:00.0: [ERR]fw PC = 0x2013e77c Dec 09 13:31:18 maketopsite kernel: rtw89_8852ce 0000:62:00.0: [ERR]fw PC = 0x2013e766 Dec 09 13:31:18 maketopsite kernel: rtw89_8852ce 0000:62:00.0: [ERR]fw PC = 0x2013e766 Dec 09 13:31:18 maketopsite kernel: rtw89_8852ce 0000:62:00.0: [ERR]fw PC = 0x2013e77c Dec 09 13:31:18 maketopsite kernel: rtw89_8852ce 0000:62:00.0: [ERR]fw PC = 0x2013e76e Dec 09 13:31:18 maketopsite kernel: rtw89_8852ce 0000:62:00.0: [ERR]fw PC = 0x2013e75e Dec 09 13:31:18 maketopsite kernel: rtw89_8852ce 0000:62:00.0: [ERR]fw PC = 0x2013e76e Dec 09 13:31:18 maketopsite kernel: rtw89_8852ce 0000:62:00.0: [ERR]fw PC = 0x2013e75e Dec 09 13:31:18 maketopsite kernel: rtw89_8852ce 0000:62:00.0: [ERR]fw PC = 0x2013e766 Dec 09 13:31:18 maketopsite kernel: rtw89_8852ce 0000:62:00.0: [ERR]fw PC = 0x2013e774 Dec 09 13:31:18 maketopsite kernel: rtw89_8852ce 0000:62:00.0: [ERR]fw PC = 0x2013e766 Dec 09 13:31:18 maketopsite kernel: rtw89_8852ce 0000:62:00.0: [ERR]fw PC = 0x2013e77c Dec 09 13:31:18 maketopsite kernel: rtw89_8852ce 0000:62:00.0: [ERR]fw PC = 0x2013e75c Dec 09 13:31:18 maketopsite kernel: rtw89_8852ce 0000:62:00.0: [ERR]fw PC = 0x2013e766 Dec 09 13:31:18 maketopsite kernel: rtw89_8852ce 0000:62:00.0: [ERR]fw PC = 0x2013e76e Dec 09 13:31:18 maketopsite kernel: rtw89_8852ce 0000:62:00.0: [ERR]fw PC = 0x2013e77c Dec 09 13:31:18 maketopsite kernel: rtw89_8852ce 0000:62:00.0: [ERR]fw PC = 0x2014abdc Dec 09 13:31:18 maketopsite kernel: rtw89_8852ce 0000:62:00.0: [ERR]fw PC = 0x2003c97e Dec 09 13:31:18 maketopsite kernel: rtw89_8852ce 0000:62:00.0: [ERR]fw PC = 0x2003c93c Dec 09 13:31:18 maketopsite kernel: rtw89_8852ce 0000:62:00.0: [ERR]fw PC = 0x2003c9a6 Dec 09 13:31:18 maketopsite kernel: rtw89_8852ce 0000:62:00.0: [ERR]fw PC = 0x201453a2 Dec 09 13:31:18 maketopsite kernel: rtw89_8852ce 0000:62:00.0: [ERR]fw PC = 0x2002d98e Dec 09 13:31:18 maketopsite kernel: rtw89_8852ce 0000:62:00.0: [ERR]fw PC = 0x2003c9fe Dec 09 13:31:18 maketopsite kernel: rtw89_8852ce 0000:62:00.0: [ERR]fw PC = 0x2003c93c Dec 09 13:31:18 maketopsite kernel: rtw89_8852ce 0000:62:00.0: [ERR]fw PC = 0x2002db4c Dec 09 13:31:18 maketopsite kernel: rtw89_8852ce 0000:62:00.0: [ERR]fw PC = 0x2014544e Dec 09 13:31:18 maketopsite kernel: rtw89_8852ce 0000:62:00.0: [ERR]fw PC = 0x20030efe Dec 09 13:31:18 maketopsite kernel: rtw89_8852ce 0000:62:00.0: [ERR]fw PC = 0x2003c98c Dec 09 13:31:18 maketopsite kernel: rtw89_8852ce 0000:62:00.0: [ERR]fw PC =

- [Finally ! i made my own OS from scratch ^_^](#) (2025/12/12 21:22)

Check github : <https://github.com/SonicExE404/FastOS> <https://reddit.com/link/1pl3kej/video/9gx92ii3bu6g1/player> submitted by /u/G_Detective
[link] [comments]

From:

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



Permanent link:

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

Last update: **2021/10/30 11:41**