Linux Kernel on Reddit

- What’s up with bugzilla website currently? (2025/03/18 09:49)
  I have many error 502 or 504 since yesterday (sometimes I can connect briefly) Is there some server issue ? https://bugzilla.kernel.org submitted by /u/Bonjour31 [link] [comments]
- Researching the Evolution of Kconfig Semantics and Parsers in Forked Projects (2025/03/17 12:22)
  Hello everyone, As a computer science student, I am conducting research on Kconfig semantics. I want to establish a method to investigate how projects like BusyBox and Coreboot, which have forked Kconfig and use this language in their applications, have modified it and how they differ from the Linux kernel. Additionally, I am interested in researching how the parsers in these veteran Kconfig projects have evolved over time. Is there a way to analyze the evolution of around 10-15 projects beyond just examining their Git logs? Since I am not an expert in this field, I am unsure about how to approach this research. Any guidance or suggestions would be greatly appreciated! submitted by /u/Funny-Citron6358 [link] [comments]
- What does the kernel do after starting init ? (2025/03/17 02:30)
  I read through a passage on kernel.org about initrd and initramfs The program run by the old initrd (which was called /initrd, not /init) did some setup and then returned to the kernel, while the init program from initramfs is not expected to return to the kernel. (If /init needs to hand off control it can overmount / with a new root device and exec another init program. See the switch_root utility, below.) but I don't really understand what it means. - Did the old /initrd just return & stop ? What would the kernel do after that ? - With the new /init, does it just run forever ? What does it do after finishing booting up the OS ? EDIT: typo submitted by /u/BabyAintBuffaloYoung [link] [comments]
- I built an OS to be compatible with Windows (2025/03/07 07:36)
  submitted by /u/kappetrov [link] [comments]
- PTE flag bits for deferred allocation (2025/03/07 03:55)
  Hello, in the book Understanding the Linux Kernel it says: "If the page does not have any access rights, the Present bit is cleared so that each access generates a Page Fault exception. However, to distinguish this condition from the real page-not-present case, Linux also sets the Page size bit to 1" However, I do not see in the code where this is done. For example when a page table is page is allocated, I do not see a page size bit being set and on a page fault I don't see a check for this. What am I missing? Further, I don't see why this would even be needed. The kernel already checks the VMA access writes to see if there is a VMA containing the virtual address. This already indicates whether the page fault is a true page not present or a programming error. submitted by /u/4aparsa [link] [comments]
- Host dev environment on Arch? (2025/03/05 17:31)
  I am trying to learn kernel development using my Arch desktop as my development machine. I am curious what the typical environment setup is for most people. I want to run my kernel in QEMU. Do you all install your tool chain on the main system alongside your other packages? Do you make any scripts to automate any aspects of the development flow? submitted by /u/landonr99 [link] [comments]
- Why do secondary CPUs wait till primary CPU initialises itself? (2025/02/27 18:11)

I have noticed secondary cpus spin in a holding pen routine until the primary CPU signals them to execute (some flag). Why is this? Why cant the secondary CPUs start executing from the same path the primary CPU takes? submitted by /u/OstrichWestern639 [link] [comments]

- CFS replacement by EEVDF as the main scheduler (2025/02/24 19:02)
  I'm trying to study and understand the CFS and EEVDF linux schedulers, and I have started reading kernel source code. As far as I know EEVDF replaced CFS for the normal scheduling classes in version 6.6 of the linux kernel (replaces as in like a modular system, CFS never existed, we all now use this shiny thing called EEVDF). Why, though, in the source code are there references of CFS? I can find the commits that introduce the new terms like, eligibility, lag etc. but e.g. the queue is still named cfs_rq, comments still reference it etc. Am I missing something? Moving to a new scheduler wouldn't also mean cleaning up the codebase in favour of clarity/readability and maintainability? submitted by /u/purplelemon42 [link] [comments]
- Learn kernel development on my Youtube series (2025/02/24 10:36)
  submitted by /u/Puzzleheaded-Try8077 [link] [comments]
- Why logical not twice in kernel codes? (2025/02/23 05:04)
  When reading code in kernel codes(I'm reading `handle_mm_fault` now), I found the kernel developers use logical not twice in many places. For example: static inline bool is_vm_hugetlb_page(struct vm_area_struct *vma) { return !!(vma->vm_flags & VM_HUGETLB); } Why use `!!(vma->vm_flags & VM_HUGETLB)` here? Isn't that `(vma->vm_flags & VM_HUGETLB)` okay? submitted by /u/DantezyLazarus [link] [comments]
- Sendmsg syscall (2025/02/22 09:48)
  I am using sendmsg syscall to send data for my serialization library. For larger sizes (8mb,40mb,80mb), it takes times on orders of milliseconds, even after applying optimizations to networking parameters. Protobuf on the other hand is still able to perform its heavy serialization and send same sized data in under 100 us. what am missing? submitted by /u/Additional_Quote5776 [link] [comments]
- find_vma_prepare (2025/02/21 16:34)
  Hello, I was looking at this function find_vma_prepare which traverses the VMA rbtree to find the previous VMA in the linked list and the parent of where a new VMA should be inserted. However, I'm confused on whether it's properly handling the case where the previous VMA is the predecessor of the VMA returned. It only seems to keep track of the previous VMA when we traverse right in the rbtree which isn't correct because if the returned VMA left subtree is non empty, we should find the predecessor. Can someone explain what I'm missing? I've attached the code. ``` static struct vm_area_struct * find_vma_prepare(struct mm_struct mm, unsigned long addr, struct vm_area_struct *pprev, struct rb_node **rb_link, struct rb_node * rb_parent) { struct vm_area_struct * vma; struct rb_node ** __rb_link, * __rb_parent, * rb_prev; __rb_link = &mm->mm_rb.rb_node; rb_prev = __rb_parent = NULL; vma = NULL; while (*__rb_link) { struct vm_area_struct *vma_tmp; __rb_parent = *__rb_link; vma_tmp = rb_entry(__rb_parent, struct vm_area_struct, vm_rb); if (vma_tmp->vm_end > addr) { vma = vma_tmp; if (vma_tmp->vm_start <= addr) return vma; __rb_link = &__rb_parent->rb_left; } else { rb_prev = __rb_parent; __rb_link = &__rb_parent->rb_right; } } *pprev = NULL; if (rb_prev) *pprev = rb_entry(rb_prev, struct vm_area_struct, vm_rb); *rb_link = __rb_link; *rb_parent = __rb_parent; return vma; } ``` submitted by /u/4aparsa [link] [comments]
- Recompiled kernel [Jetson Orin Nano 8GB] - Lost all networking (2025/02/20 21:39)
  submitted by /u/Matt1XS [link] [comments]

- Why does KVM have KVM_MEM_MAX_NR_PAGES limit on how many pages userspace(Qemu) can give to guest? (2025/02/17 13:15)
  I get that KVM_MEM_MAX_NR_PAGES is (1<<31) - 1, which is pretty huge, but why is there a limit? submitted by /u/OstrichWestern639 [link] [comments]
- Are kernel developers underpaid? (2025/02/17 12:23)
  From what I see, people working on web development, and calling APIs are making 200k+ on top companies. Although these companies do pay a lot, but every job is different. (Right?) As a kernel programmer, I believe we solve pretty hard problems (biased opinion). Is it true that we are underpaid? Looking for some experiences. submitted by /u/OstrichWestern639 [link] [comments]
- [Newbie Question] Not Showing in dmesg After Building Kernel with virtme-ng (2025/02/16 18:28)
  Hi all, I'm a newbie trying to build the kernel for the first time. To speed up compilation, I decided to use virtme-ng, which seemed like a good option. I'm following the steps from KernelNewbies: First Kernel Patch. Specifically, I modified the probe function of my WiFi driver by adding a printk, as described in the "Modifying a driver on native Linux" section. I tried also with the e1000e driver. Both of them are listed as result inlsmod. I have also updated the .config section related to printk to enable the maximum level for log. I compiled the kernel using vng -b and booted it with vng, but I don't see the printk output in dmesg. Am I missing something? Any ideas on what I might be doing wrong? Thanks! submitted by /u/Crafty_Yoghurt7610 [link] [comments]
- Where to find resources for memory management as of 2025? (2025/02/10 12:59)
  I mostly find articles about buddy allocator, slab/slub, etc. which are fairly high level. Are there resources which I can go through before delving into the source code? submitted by /u/OstrichWestern639 [link] [comments]
- Guidance to compile the linux kernel (2025/02/09 07:45)
  Hi, I am trying to recompile the linux kernel and facing some issues can y'all help me out please? My OS is the ubuntu 24.04 LTS. The kernel is the 5.19.8 from here. When I run make I used to get the following issue: CC kernel/jump_label.o CC kernel/iomem.o CC kernel/rseq.o AR kernel/built-in.a CC certs/system_keyring.o make[1]: *** No rule to make target 'debian/certs/debian-uefi-certs.pem', needed by 'certs/x509_certificate_list'. Stop. make: *** [Makefile:1851: certs] Error 2CC kernel/jump_label.o CC kernel/iomem.o CC kernel/rseq.o AR kernel/built-in.a CC certs/system_keyring.o make[1]: *** No rule to make target 'debian/certs/debian-uefi-certs.pem', needed by 'certs/x509_certificate_list'. Stop. make: *** [Makefile:1851: certs] Error 2 I did as one of the user in thie stackoverflow post said scripts/config --disable SYSTEM_TRUSTED_KEYS scripts/config --disable SYSTEM_REVOCATION_KEYS Now I get the and then when I run make I get the following issue, this I am not sure how I should go about solving it make[1]: *** No rule to make target 'y', needed by 'certs/x509_certificate_list'. Stop. make: *** [Makefile:1847: certs] Error 2 submitted by /u/rakk109 [link] [comments]
- error: section type conflict when compiling old kernel with newer gcc (2025/02/08 00:05)
  # v2.6.39 61c4f2c81c61f73549928dfd9f3e8f26aa36a8cf === drivers/acpi/osl.c === 1094 static struct osi_setup_entry __initdata osi_setup_entries[OSI_STRING_ENTRIES_MAX]; // ... 1599 acpi_status __init acpi_os_initialize(void) 1600 { 1601 acpi_os_map_generic_address(&acpi_gbl_FADT.xpm1a_event_block); 1602 acpi_os_map_generic_address(&acpi_gbl_FADT.xpm1b_event_block); 1603 acpi_os_map_generic_address(&acpi_gbl_FADT.xgpe0_block); 1604 acpi_os_map_generic_address(&acpi_gbl_FADT.xgpe1_block); 1605 1606 return AE_OK; 1607 } == === error messages === drivers/acpi/osl.c:1600:1: warning: ignoring attribute ‘section (".init.text")’ because it conflicts with previous ‘section

(".init.data")’ [-Wattributes] drivers/acpi/osl.c:1094:42: error: ‘osi_setup_entries’ causes a section type conflict with ‘acpi_os_initialize’ 1094 | static struct osi_setup_entry __initdata osi_setup_entries[OSI_STRING_ENTRIES_MAX]; | ^~~~~~~~~~~~~~~~~ drivers/acpi/osl.c:1599:20: note: ‘acpi_os_initialize’ was declared here 1599 | acpi_status __init acpi_os_initialize(void) ======================== === CFLAGS === gcc -Wp,-MD,drivers/acpi/.osl.o.d -nostdinc -isystem /usr/lib/gcc/x86_64-pc-linux-gnu/14.2.1/include -I/home/xmori/trylinux/linux/arch/x86/include -Iinclude -include include/generated/autoconf.h -D__KERNEL__ -Wall -Wundef -Wstrict-prototypes -Wno-trigraphs -fno-strict-aliasing -fno-common -Werror-implicit-function-declaration -Wno-format-security -fno-delete-null-pointer-checks -Os -m64 -mtune=generic -mno-red-zone -mcmodel=kernel -fno-pie -funit-at-a-time -maccumulate-outgoing-args -DCONFIG_AS_CFI=1 -DCONFIG_AS_CFI_SIGNAL_FRAME=1 -DCONFIG_AS_CFI_SECTIONS=1 -DCONFIG_AS_FXSAVEQ=1 -pipe -Wno-sign-compare -fno-asynchronous-unwind-tables -mno-sse -mno-mmx -mno-sse2 -mno-3dnow -Wframe-larger-than=2048 -fno-stack-protector -fno-omit-frame-pointer -fno-optimize-sibling-calls -Wdeclaration-after-statement -Wno-pointer-sign -fno-strict-overflow -fconserve-stack -DCC_HAVE_ASM_GOTO -Os -D"KBUILD_STR(s)=#s" -D"KBUILD_BASENAME=KBUILD_STR(osl)" -D"KBUILD_MODNAME=KBUILD_STR(acpi)" -c -o drivers/acpi/osl.o drivers/acpi/osl.c ============= === gcc version === gcc version 14.2.1 ==================== === include/linux/init.h === #define __init __section(.init.text) __cold notrace #define __initdata __section(.init.data) ================================ _static struct osi_setup_entry __initdata osi_setup_entries[OSI_STRING_ENTRIES_MAX]; acpi_status __init acpi_os_initialize(void) `osi_setup_entries` is an unintialized static variable, so it goes to .bss. `acpi_os_initialize` is a function, so it goes to .text. Why these two caused a section-type-conflict error? submitted by /u/VegetablePrune3333 [link] [comments]

- question on DM verity (2025/02/07 16:08)
  tldr where in the kernel code does the verity check occur on the IO read request to verify the block is part of the merkle tree Hi, I'm relatively new when it comes to the Linux Kernel Implementation. I was wondering how DM Verity is actually invoked when the Kernel does a read operation (ie. where does it hash the requested block and calculates the roothash with the merkel tree in the meta-data of the verity-hash partition. I wanted to extend the logging capabilities of DM Verity, not just logging a corruption but giving more measurements and information. I wanted to find the implementation of that in the Kernel's source code (github.com/torvalds/linux) but I couldnt really find the code where the mentioned check occurs. Can anyone with more expirience point me in the right direction? submitted by /u/1Goal_1Dream [link] [comments]
- follow_page() on x86 (2025/02/03 02:23)
  Hi, I was looking at the implementation of follow_page for 32bit x86 and I'm confused about how it handles the pud and pmd. Based on the code it does not seem to handle it correctly and I would have assumed that pud_offset and pmd_offset would have 0 as their 2nd argument so that these functions fold back onto the pgd entry. What am I missing? ``` static struct page * __follow_page(struct mm_struct *mm, unsigned long address, int read, int write) { pgd_t *pgd; pud_t *pud; pmd_t *pmd; pte_t *ptep, pte; unsigned long pfn; struct page *page; page = follow_huge_addr(mm, address, write); if (! IS_ERR(page)) return page; pgd = pgd_offset(mm, address); if (pgd_none(*pgd) || unlikely(pgd_bad(*pgd))) goto out; pud = pud_offset(pgd, address); if (pud_none(*pud) || unlikely(pud_bad(*pud))) goto out; pmd = pmd_offset(pud, address); if (pmd_none(*pmd) || unlikely(pmd_bad(*pmd))) goto out; if (pmd_huge(*pmd)) return follow_huge_pmd(mm, address, pmd, write); ptep = pte_offset_map(pmd, address); if (!ptep) goto out; pte = *ptep; pte_unmap(ptep); if (pte_present(pte)) { if (write && !pte_write(pte)) goto out; if (read && !pte_read(pte)) goto out; pfn = pte_pfn(pte); if (pfn_valid(pfn)) { page = pfn_to_page(pfn); if (write && !pte_dirty(pte) && !PageDirty(page)) set_page_dirty(page); mark_page_accessed(page); return page; } } out: return NULL; } ``` submitted by

/u/4aparsa [link] [comments]

- Help UEFI Configurations Problems (2025/01/28 23:42)
  submitted by /u/Heavy_Spite6441 [link] [comments]
- Is futex_wait_multiple accessible from userspace? (2025/01/22 18:47)
  I'm trying to figure out how/if I can call futex_wait_multiple from an application. I'm on kernel 6.9.3 (Ubuntu 24.04). As far as I can tell from the kernel sources, futex_wait_multiple is implemented in futex/waitwake.c, but there's no mention of it in the futex(2) manpage or in any of my kernel headers. submitted by /u/hazard02 [link] [comments]
- Can I submit a driver upstream to the kernel if it wasn't written by me? (2025/01/22 04:49)
  I recently found a driver on GitHub that seems to work. An equivalent driver is not currently in the kernel tree. The driver was not written by me, but has appropriate Copyright/compatible license headers in each file. Can I modify the driver and upstream it to the kernel? I would happily maintain it, and I would probably drop it off in staging for a while, but are there any issues with me submitting code that I have not wholly written? I would of course audit all of it first. submitted by /u/speedcuber111 [link] [comments]
- Will Linux allocate pids < 300 to user processes? (2025/01/20 16:42)
  I was looking at the Linux 2.6.11 pid allocation function alloc_pidmap which is called during process creation. Essentially, there's a variable last_pid which is initially 0, and every time alloc_pidmap is called, the function starts looking for free pids starting from last_pid + 1. If the current pid it's trying to allocate is greater than the maximum pid, it wraps around to RESERVED_PIDS which is 300. What I don't understand is that it doesn't seem to prevent pids < 300 from being given to user processes. Am I missing something or will Linux indeed give pids < 300 to user processes. And why bother setting the pid offset to RESERVED_PIDS upon a wrap around if it doesn't prevent those being allocated the first time around. I've included the function in a paste bin for reference: https://pastebin.com/pnGtZ9Rm submitted by /u/4aparsa [link] [comments]