

# 欢迎参加龙架构双周会

## • 编辑权限申请

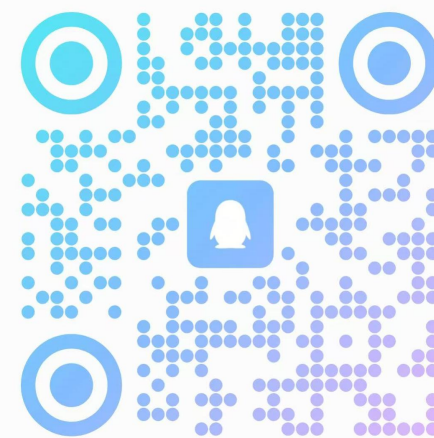
- 计划好主讲的议题和大致用时
- 在本文档申请编辑权限且附上简短的申请理由
- 在龙架构双周会交流群中 **@群主** 或 **管理员** 获取权限
- 向 [loongarch@whlug.cn](mailto:loongarch@whlug.cn) 发送主题为龙架构双周会报告的邮件
  - 邮件内请简要说明您将要报告的内容，我们将在收到邮件后同您取得联系，为您提供文档的编辑权限

## • 内容编辑

- 请在对应的议题版块下添加您想要分享的内容
- 若无对应议题，请直接在幻灯片其他议题最前方添加
- 快速报告一页控制在 3 分钟以内，报告期间请勿讨论发言
- 专题报告 15~30 分钟，分享结束后可讨论交流

# 龙架构双周会

2026年5月10日·第36次





# 会前注意事项

# 会前注意事项

- 本次会议仅涉及软件技术课题
  - 关于龙芯相关的硬件产品，除官方层面已解禁的消息及本文档内可公开的消息外，其他均不作任何回应
- 本次会议与股市无关，不构成任何投资建议

**倡议：龙架构双周会及附属群组主要为社区开发者提供技术沟通和协调渠道，而非投资者交流或政治、商业讨论，请自觉控制话题及占用时长，更不要主动引发甚至煽动厂商间对立、饭圈争议等非建设性议题**

(白铭骢)



# 快速报告

龙架构上游动向

# Binutils

- zhaozhou [修复了](#)龙架构汇编器 (gas) 在无显式对齐指令时不对齐的问题，强制机器指令 4 字节对齐，避免链接后指令读取错误；同时 mengqingang 增加测试用例 insn\_align\_4.s 验证对齐行为

# GCC

- GCC 16.1 已发布
  - 本页所述事项需在 16.2 处理
- LuLu Cheng:
  - [修复](#) PR 125057: 拆分向量搬移操作时没有考虑 GCC 16 新增的 `__lasx_cast_128` 生成的描述从 128 位向量到 256 位向量寄存器搬移的 RTL，将其当作 GPR 到 FPR 的搬移去拆分了，拆成了非法的 RTL，导致崩溃
- Xi Ruoyao:
  - [修复](#) PR 125049: stack protector 实现中 stack canary 的真值和存放真值的地址生存期过长，导致攻击者可通过读取寄存器中遗留的真值，或（在寄存器压力较大时）覆盖 spill 到栈上的地址，绕过保护机制
  - 在内嵌汇编中[允许](#) `%c` 占位符输出符号名（在 GCC 16 新增的 extended top-level assembly 中有较高概率被用到）

# GCC (17 周期)

- Xi Ruoyao:
  - [修复](#)为 LA64 展开 32 位整数 ctz 操作时 ctz.w 指令后有一条多余的符号扩展指令 (slli.w ..., 0) 的问题
  - 增加 spaceship 操作, 将  $(a == b) ? 0 : (a < b) ? -1 : 1$  优化成两条 slt[u] 指令和一条减法指令
    - Lulu Cheng 指出其中寄存器符号/零扩展状态标记疑似有误, 可能需修改
  - 模仿 RISC-V zbb 将  $a \wedge b \wedge (a | c)$  改写成  $(c \& \sim a) \wedge b$ , 以使用上 andn 指令
    - Lulu Cheng 和 Waterman 指出提交消息有误, 需修改

# LLVM

- lrzlin 为龙架构 LSX 向量符号扩展[实现了](#)自定义 Lowering，通过组合 vslti 和 vilvl/vilvh 指令，生成更高效的指令序列
- lrzlin 为龙架构的 LSX 和 LASX 向量扩展 sitofp、uitofp [添加了](#)测试用例，覆盖了多种位宽和向量长度；CI 自动检查发现测试中使用了已弃用的 undef，需要修正。
- heiher 为龙架构 vNi128 类型[添加了](#)向量加/减支持，LLVM 目前对 v1i128（128-bit 向量中包含一个 128-bit 整数）和 v2i128（256-bit 向量中包含两个 128-bit 整数）类型的向量 ADD/SUB 操作，会将其完全展开为标量运算，为了将此行为合法化，扩展了 LSX/LASX 的指令选择模式，使其能够生成龙架构原生的 Q 元素大小的向量 ADD/SUB 指令：VADD.Q、VSUB.Q(LSX)和 XVADD.Q、XSUB.Q(LASX)，并[添加了](#)测试文件
- CSharperMantle 为龙架构 LLVM 后端和 Clang 驱动[添加了](#) -fstack-clash-protection 栈冲突保护 (stack clash protection) 支持，实现方式参考了 RISC-V 后端：对于固定大小栈分配使用循环展开策略进行探测，对于动态栈分配生成探测循环

# Rust

- heihier [重构了](#) Rust 标准库的 SIMD 架构支持模块 stdarch 中龙架构部分的可移植 SIMD 的辅助代码的命名和内部路径，将 SimdL 重命名为 SimdExt，相关宏和辅助函数也做了对应调整，并使用 is::（内部 intrinsics）、cs::（核心架构 SIMD 模块）、ls::（龙架构本地 SIMD 辅助函数）等别名显式限定了不同来源的路径
- heihier [重构了](#) 龙架构在 stdarch 库中实现 vbitclr、vbitrev、vbitset 等 SIMD 位操作的方式，改为统一使用 crate::intrinsics::simd 路径

# Linux Kernel

- 平台支持

- Huacai Chen: 为每个 PCIe 根桥添加 PCIH 标志检查, 跳过内核的无条件 mem resource fixup ([第 1 版](#))
- Hongliang Wang: 为龙芯 2 号 I2C 总线驱动添加时钟属性并调整总线速率 ([第 3 版](#))
- Qunqin Zhao: 为 SE 驱动添加多节点支持 ([第 2 版](#))
- Binbin Zhou: 龙芯 CAN-FD 总线驱动 ([第 1 版](#))

- KVM 子系统

- Tao Cui: 使用 memset64() 简化 \_kvm\_pte\_init() ([第 1 版](#), 已驳回, 潜在性能劣化)
- Qiang Ma: 将龙架构 KVM 的 KVM\_CAP\_NR\_VCPUS (用于返回推荐的最大 vCPU 数量) 返回值用 KVM\_MAX\_VCPUS (用于返回架构允许的绝对最大 vCPU 数量) 作为上限限制, 避免推荐 vCPU 数量超过架构硬限制 ([第 2 版](#))
- Bibo Mao: 将一部分变量挪入 paravirt.h, 避免 CONFIG\_SMP 未开启时, 由于 asm/qspinlock.h 未被引用导致的的编译警告 ([第 1 版](#))
- Bibo Mao: 去除在软件计时器过期时不必要的中断植入 ([第 1 版](#))

# Linux Kernel

- 代码或功能修复
  - Huacai Chen: 默认启用 CONFIG\_64BIT, 在 7.1 周期引入 32BIT 选项之后若不设置 64BIT 默认将破坏现有配置 ([第 1 版](#))
  - Huacai Chen: 使用 cc-option 启用 -m32/-m64 选项 (GCC 暂不支持), 修复 Clang 构建 ([第 1 版](#))
  - Huacai Chen: 修复 32 位 SYM\_SIGFUNC\_START 定义 ([第 1 版](#))
  - Wentao Guan: 修复 loongson\_gpu\_fixup\_dma\_hang() 可能导致 ADE 例外的问题 ([第 3 版](#))
  - Rui Wang: 将 KASLR 逻辑移入 EFI Stub, 避免与 initrd 内存段重合导致启动失败 ([第 4 版](#))
- 代码重构或清理
  - Qiang Ma: 删除暂未使用的 cpu\_has\_perf 定义 ([第 1 版](#))
  - Qiang Ma: 清理 show\_cpuinfo() 代码逻辑 ([第 1 版](#), 被驳回)
  - Qiang Ma: 修正 CSR\_CPUID\_COREID{,\_WIDTH} 定义 ([第 1 版](#), 被驳回)

# Box64 (四周进度)

- v0.4.2 发版
- 继续完善单元测试基础设施
- 继续修复 JIT 中指令翻译错误若干
- 使用 4 位查表法优化了 PCLMUL 指令的 C 模拟函数性能
- 修复库包装小错误若干
- 环境变量文档新增中文
- 修复代码文件缓存的一些问题（计划下次发版前默认启用）
- EFLAGS 延后计算函数从 C 变为 JIT 内联
  - 优化 context switch 开销
  - 充分利用 LBT

# UEFI 固件 (EDK II 上游)

- kilaterlee [修复并优化了](#) PC 相对寻址的重定位逻辑，原有的重定位处理方法无法正确处理高位 (HI) 与低位 (LO) 不相邻以及一个 HI 对应多个 LO 的复杂场景
  - 同时增加了 CLANGDWARF (LLVM) 工具链的支持，可以选择使用 LLVM 22.1.0+ 来构建龙架构的 UEFI 固件

# 其他开源项目动向

## • 问题报告/英雄帖

- bh1xaq [提出了](#)为开源 3D 打印机控制编程软件 BambuStudio 添加龙架构支持
  - 目前主程序已经在龙芯平台 (Loongson-3B6000x1-7A2000x1-EVB) 上成功构建，但网络插件因缺少 loongarch64 的预编译包而无法使用
  - bh1xaq 希望官方将其加入发布列表，同时指出社区维护的开源替代插件可作为备选
- wojiushixiaobai 向 python-build-standalone 项目[报告了](#)在龙架构运行预编译的 Python 二进制文件时，因 ELF 段的对齐方式（基于 4KB 页面）与目标系统内核实际使用的页面大小（16KB 或 64KB）不匹配，导致加载时发生段错误
  - **上周，上游已通过更新 patchelf 到 0.14 并设置龙架构默认内存页为 64KiB 解决该问题**

## • 上游功能实现、修复等

- MarsDoge 和 yetist 龙架构上 gnu-efi 库可能被工具链默认向量化优化编入 LSX/LASX 指令的问题，通过在编译显式添加 -mno-lsx -mno-lasx 标志，确保 gnu-efi 不包含 SIMD 指令，从而避免 UEFI 环境下因使用向量指令导致崩溃；**该补丁仍待审阅和测试**
- phprus 为 zlib-ng（zlib 压缩库的下一代分支）的 /delta CI workflow [添加了龙架构支持](#)；目前该提交已被合并

# 其他开源项目动向

## • 上游功能实现、修复等

- zevorn 为 machina（一个用 Rust 编写的虚拟化/模拟器项目）[添加了](#)龙架构支持，包括指令解码/翻译、MMU/中断、设备模型及 Linux 引导能力，新增 loongarch64-ref 参考机器，并通过了测试验证
- MutsukiC [修复了](#)微软 Edit（Edit 是一个具有文本用户界面 (TUI) 的无模式编辑器）项目中 stdext 模块的 crate 属性在龙架构上使用了 stdarch\_loongarch 的不稳定特性，但该 crate 未通过 `#![feature(stdarch_loongarch)]` 属性显式声明启用此特性，导致编译失败；已合入
- martin-frbg [修复了](#) OpenBLAS 在龙架构上配置时，无法将 LSX 和 LASX 的启用状态正确打印到配置文件中的问题；该修复目前已合并
- lrzlin [修复了](#) PPSSPP 模拟器龙架构版本中 Jit\_WeightsU16Skin 函数的代码错误，解决了在运行《初音未来 - 歌姬计划 扩展版》等游戏时角色模型被错误拉伸的图形渲染问题；该补丁已合并
- Ponsanthini 从 BitBucket [迁移了](#) Hao Chen 编写的 x265 视频编码器龙架构 SIMD 优化支持补丁，通过利用 LSX/LASX 向量扩展指令集，对 DCT、量化、帧内预测、运动补偿、环路滤波等核心模块进行了汇编级优化，预计会显著提升龙芯平台上的视频编码性能；**该补丁仍有待审阅，欢迎各位开发者建言献策**



# 快速报告

龙架构发行版动向

# 发行版维护贴士

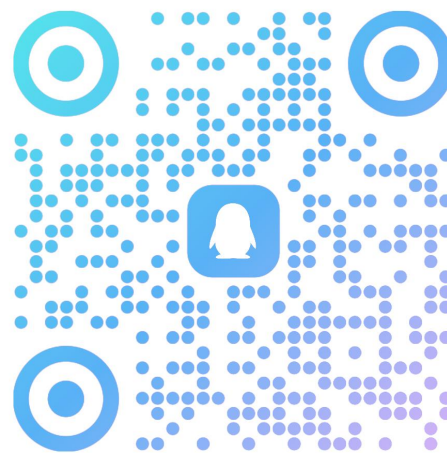
- 建议尽早回合 GCC PR 125049 的修复（后续除 GCC 16.2 外还会合入 15.3、14.4、13.5，如果还在用 GCC 12 请自行处理），因为必须用修复后的编译器重新编译软件包才能起效
- Linux 6.19 周期的一笔 AMDGPU [提交](#)无解释地删去了之前临时处理 `dml21_map_dc_state_into_dml_display_cfg` 触发内核态 FPU 未启用例外的[代码](#)，而最终的[修复](#)在 7.1 开发周期才进入主线，因此 6.19 和 7.0 内核目前无法使用 9060XT 等显卡
  - 猜测是 AMD 的人整理内部代码往上游发的时候搞混了
  - 正在考虑是加回临时处理还是回合最终修复

# Linux From Scratch

- 为 Chip & Cheese 定制了一版 LFS 12.0 改版，供他们构建 SPEC CPU 2026 的二进制工具并提交 SPEC 官方
  - 用 12.0 是为了卡 glibc 和 GCC 版本，使得编译出的二进制能在使用 glibc  $\geq 2.38$ , GCC  $\geq 13.2$  的发行版运行
  - 和实际跑分用的编译器没有关系
- 停止将引导协议的差异描述为“新旧世界”差异，以避免“新旧世界”概念的无限扩张引发误解

# 安同 OS

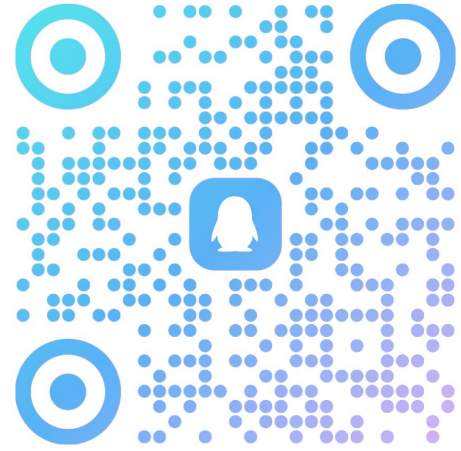
- **Linux 6.18.27 已推送，包含安全漏洞修复**
  - Dirty Frag 及 Copy Fail
  - 请择机更新您的安同 OS（服务器、多用户或公用设备用户请加急此更新）
- **Linux 7.0 可供测试**
  - 已知问题：Intel 显卡无法驱动、AMD RX 7600 在处理器负载高时出现渲染错误
  - **oma topics --opt-in linux-kernel-7.0.3**
- **OpenJDK 25**
  - 已交付并设置为默认 OpenJDK 版本，实测性能有显著提升
- **本期安全更新（非预装软件未列出）**
  - **Exiv2 0.28.8**：修复了两个越界读取漏洞（CVE-2026-25884、CVE-2026-27596，严重性：高）及一个整数溢出漏洞（CVE-2026-27631，严重性：中）
  - **Expat 2.8.0**：修复一个哈希洪水漏洞（CVE-2026-41080，严重性：高）
  - **libde265 1.0.18**：修复一个缓冲区溢出漏洞（CVE-2025-61147，严重性：中）、一个空指针解引用漏洞（CVE-2026-33164，严重性：高）和一个越界写入漏洞（CVE-2026-33165，严重性：中）



# 安同 OS

## • 本期安全更新（非预装软件未列出）

- **libexif 0.6.26**: 修复两个整数下溢漏洞（CVE-2026-32775、CVE-2026-40386，严重性：高）和一个整数溢出漏洞（CVE-2026-40385，严重性：高）
- **libjxl 0.11.2**: 修复一个使用未初始化资源的漏洞（CVE-2025-12474，严重性：中）和一个无限资源分配漏洞（CVE-2026-1837，严重性：高）
- **LibRaw 0.22.1**: 修复含 4 个严重漏洞、2 个高危漏洞及 4 个中危漏洞
- **LibVNCServer 0.9.15**: 修复了一个内存泄漏漏洞（CVE-2020-29260，严重性：高）、一个堆越界读取漏洞（CVE-2026-32853，严重性：高），以及一个空指针解引用漏洞（CVE-2026-32854，严重性：高）
- **libxml2 2.15.1**: 修复 10 个安全漏洞（含 1 个严重漏洞和 8 个高危漏洞）
- **Little CMS 2.19**: 修复了一个整数溢出漏洞（CVE-2026-41254，严重性：高）
- **lrzip 0.660**: 修复拒绝服务漏洞（CVE-2022-33067、CVE-2023-39741，严重性：中）、一个释放后使用漏洞（CVE-2025-15570，严重性：高）和一个空指针解引用漏洞（CVE-2025-15571，严重性：中）
- **mako 1.3.12**: 修复一个路径遍历漏洞（CVE-2026-41205，严重性：高）
- **rsync 3.4.2**: 修复一个哈希洪水漏洞（CVE-2026-41080，严重性：高）



# 安同 OS

- 本期安全更新（非预装软件未列出）
  - **Thunderbird 150.0**: 修复 Mozilla 基金会第 2026-33 号安全公告中披露的 40 个安全漏洞  
高危漏洞)
  - **CUPS 2.4.19**: 修复一个越界读取 (Out-of-bounds Read) 漏洞 (CVE-2026-41079, 严重)
  - **Libgcrypt 1.12.2**: 修复一个越界写入漏洞 (CVE-2026-41989, 严重性: 中)
  - **libssh 0.12.0**: 修复自 0.8.9 版本以来的 9 个安全漏洞
  - **Vim 9.2.0437**: 修复一个系统命令注入漏洞 (严重性: 中)
- 忒多了, 对吧? 下次我就不念了, 各位麻烦看看 oma 的更新详情描述吧.....
- 建议关注公众号“安同开源”或社区主页 ([aossc.io](http://aossc.io)) 新闻





# 快速报告

社区事务

# 外语双周会（英 + 俄）

- **下周三（5月13日）晚9时，第一期外语双周会即将开始试运行**
  - 本次双周会的内容都将用英 + 俄语复述
  - 欢迎各位参和分享（尤其是直接参与上游项目贡献的同志们）！
- 参与方式
  - 幻灯片：[Google Docs](#)
  - 在线会议：[Zoom](#)
- 将尝试在 VK 和 YouTube 直播，录播将在这两个平台以及 Bilibili 发布



# 问答环节

社区问答及意见反馈

# 龙芯爱好者社区

共生 · 共建 · 共进

