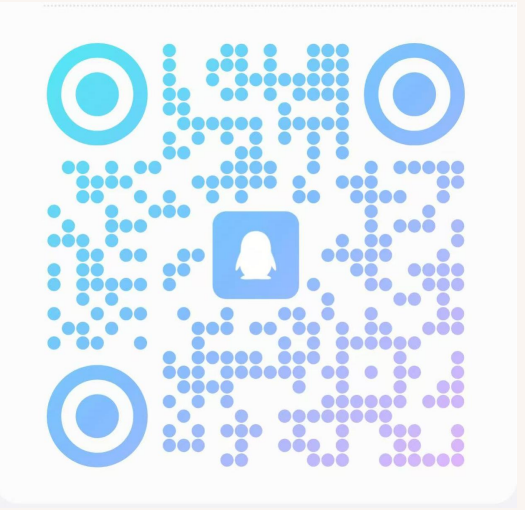
欢迎参加龙架构双周会

- **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 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 LoongArch Biweekly 双周会

龙架构双周会

2025 年 3 月 2 日 · 第 6 次



龙架构 LoongArch Biweekly 双周会

快速报告

龙架构社区八卦

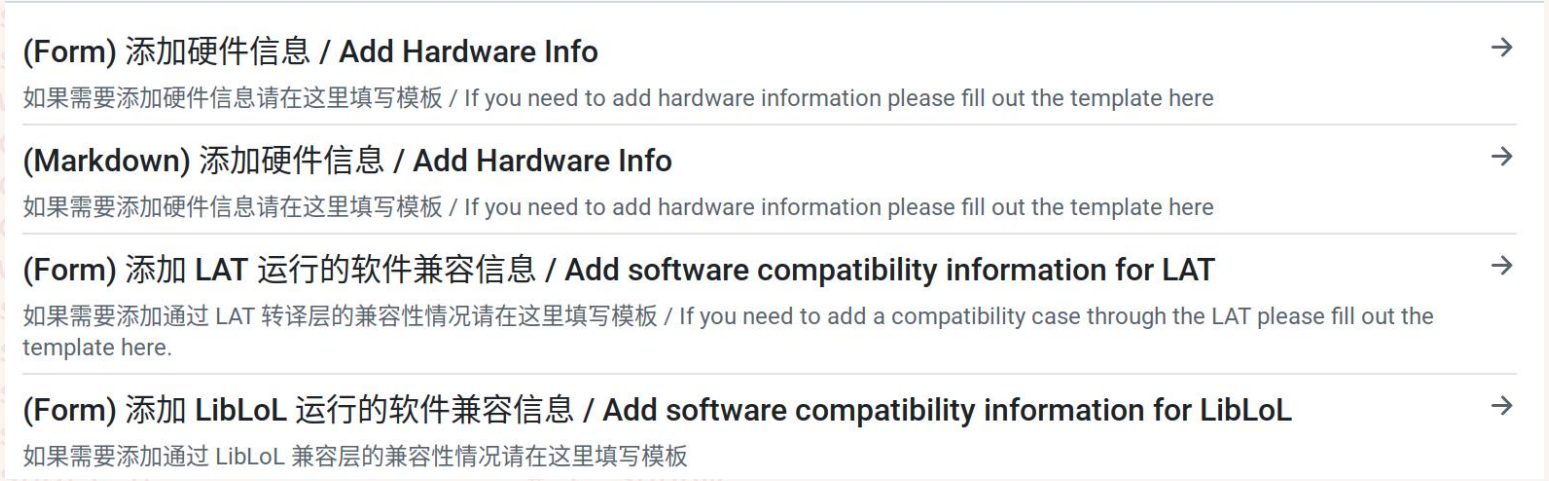
龙架构 LoongArch Biweekly 双周会

LoongArch 兼容性数据库信息收集

网站地址：https://loongarch.liaronce.com

降低提交门槛

- 先前有人提到步骤多，对小白不太友好，在研究了 GitHub 议题表单后现已提供表单样式直接填写
  - 现在仅需根据表单提示填写即可
    （GitHub 地址：https://github.com/LiarOnce-LoongAL/loong-compatible-database/issues/new/choose）
  - 对于硬件则保留 Markdown 模板，可以通过该模板提交 linux-hardware.org 的链接
    - 原因是因为发现提交 linux-hardware 提交的信息更简单直观。。。



龙架构 LoongArch Biweekly 双周会

Ren'Py

项目源地址：https://www.renpy.org

Ren'Py是一个强大的视觉小说引擎，采用Ren'Py引擎的游戏包括治愈向galgame《Doki Doki Literature Club》与国产galgame《候鸟》和《夏末白夜》等。

通过移植pygame-sdl2库和renpy本体，Ren'Py引擎的游戏得以在龙架构原生运行。

还有部分调试和打包工作需要完成，此后会放出构建方法和二进制包。

以下是《候鸟》和《夏末白夜》的运行截图：

OceanBase

- 背景
  - 在 2022 年尝试给 OceanBase 上游提 PR 建议在不影响现有的静态编译方式基础上，增加动态编译、支持高版本工具链和库、支持更多的发行版等特性，将 OB 改造成通用 cmake 项目。上游以商业数据库追求稳定，不愿更新工具链为由一直搁置。当时 OceanBase 还是 v3版本，2025年 OceanBase 已经到 v4 版本了，基于最新的 develop 分支拉了一个新分支，希望在新世界重新编译 OceanBase (目前以ArchLinux为例)
- 现状
  - 由于 OceanBase 上游不愿意更新依赖包，目前如果要在新世界上编译，需要手动编译低版本的部分依赖包，如 RocksDB，Boost, MySQL-Connector-C++ 等等。
- 更好的解决方式
  - 参考 ob-deps 的方式，在ArchLinux上使用aur的方式提供指定版本的第3方依赖包，其他基于rpm打包方式的新世界操作系统则需要提供 rpm 包。

快速报告

龙架构上游动向

龙架构 LoongArch Biweekly 双周会

Firefox

- 官方列车时刻表
- Firefox 137：04-01 正式发布
  - 02-27 软冻结（大宗修改最后期限），03-03 离开主线分支
  - WebRTC 支持已主线化，下游补丁数 -3
  - libvpx LSX 支持已主线化，下游补丁数 -2
  - libpng LSX 支持恢复即将主线化，下游补丁数 -2
  - Skia & libyuv 暂无进展

Linux 内核（loongarch 列表）

- Bibo Mao
  - 修复 HAVE_VIRT_CPU_ACCOUNTING_GEN 错误依赖 !SMP 的问题（第 2 版）
  - 新增 KVM 客户机 perf 事件监测支持（第 1 版）
- Xi Ruoyao
  - 修复 vgetrandom-chacha.S 未利用 t8 寄存器的问题（第 1 版）
  - 构建 vDSO 代码时使用 GNU Hash Style 而非 GNU + Sys-V（第 1 版）
- Qunqin Zhao
  - 龙芯安全引擎、TPM 设备支持（第 4 版）
- Binbin Zhou
  - 修复 LS2K2000/LS7A I2C 中分频器寄存器的访问方式（第 3 版）
  - 龙架构平台 PWM 支持（第 9 版）

龙架构 LoongArch Biweekly 双周会

Linux 内核（loongarch 列表）

- Haoxiang Li
  - 为 loongson2_guts_probe( ) 函数中的 devm_kstrdup( ) 调用增加错误检查（第 2 版）
- Yuli Wang, Wentao Guan
  - 删除未使用的 get_numa_distances_cnt() 函数声明（第 2 版）
  - 删除龙架构 ACPI 代码 parse_acpi_topology() 函数中无用的返回码检查（第 1 版）

Linux 内核（外设）

- 白铭骢提交了 Intel Xe DRM 驱动在非 4KiB 分页内核上的功能修复
  - 原型补丁来自“自发对称破缺”，补丁来自此处
  - 安同 OS 在去年开始就引入了这组补丁的最初版本进行测试，长期以来用户反响比较正面且没有发现显著稳定性问题
  - 过去几天进行了修缮、对相关原理和修改依据进行了整理和查验，邀请多位社区好友参与不同平台的测试验证，最终将这组补丁整理了出来
    - Intel Xe (DG1) 及 Arc A/B 系列均已测试通过
  - 感谢社区好友“自发对称破缺”的原型补丁以及他与 Kexy Biscuit 的审阅支持，也感谢一起熬夜测试的“安慕希”、Komatsuzaki_Takizawa 和 PowerVR 等人

GNU 工具链

- Glibc fm{ax,in}imum{,_num,_mag,_mag_num}f 优化
  - 已经合并
- GCC 原子操作代码清理和优化，及 16 字节原子操作实现
  - 已经发送（可能需要硬件部门确认正确性）
- GCC 另有两项小修改正在审阅中
  - 位操作-alsl 指令对的针对性识别
  - 使用 LSX 进行整数标量 popcount 的微优化

Skia

- xen0n 指出 `scaled_mult` 这个函数当前的 LSX 实现没有进行四舍五入，导致单元测试失败，并提交了修复
  - 预期的语义是 `(2*a*b+(1<<15))>>16`
    - `1<<15` 这一项的作用是：如果乘法运算结果的那一位为 1，则最终结果的最低位将获得一个进位。等效于将十进制数 x.0 不变，x.5 变为 (x+1).0
  - 但当前的 LSX 实现直接取乘法运算高半部分的结果进行移位，没有舍入
- 然而原作者以性能问题为由反对，并认为这个误差是可以接受的
  - ……吗?
  - 如果可接受，是不是应该加注释&改写单元测试，而不是假装没看见测试失败?
- 目前陷入搁浅，希望关注 Skia 的人士发表意见

软件包对手册未指明行为的依赖

- sljit 等项目依赖 `ftint` 指令将超出范围的数值钳位到整数类型能表示的最大或最小值，将 NaN 转成 0，但手册未有此规定
- glibc 等项目依赖 `fmax` 和 `fmin` 指令将 -0.0 视为小于 +0.0，但手册未有此规定
- 希望硬件设计部门能确切答复软件是否可依赖这两项行为(即未来的硬件是否可能有不同行为)，如可依赖则应记入手册，否则维护者们就要尽早修复软件了

快速报告

龙架构发行版变动

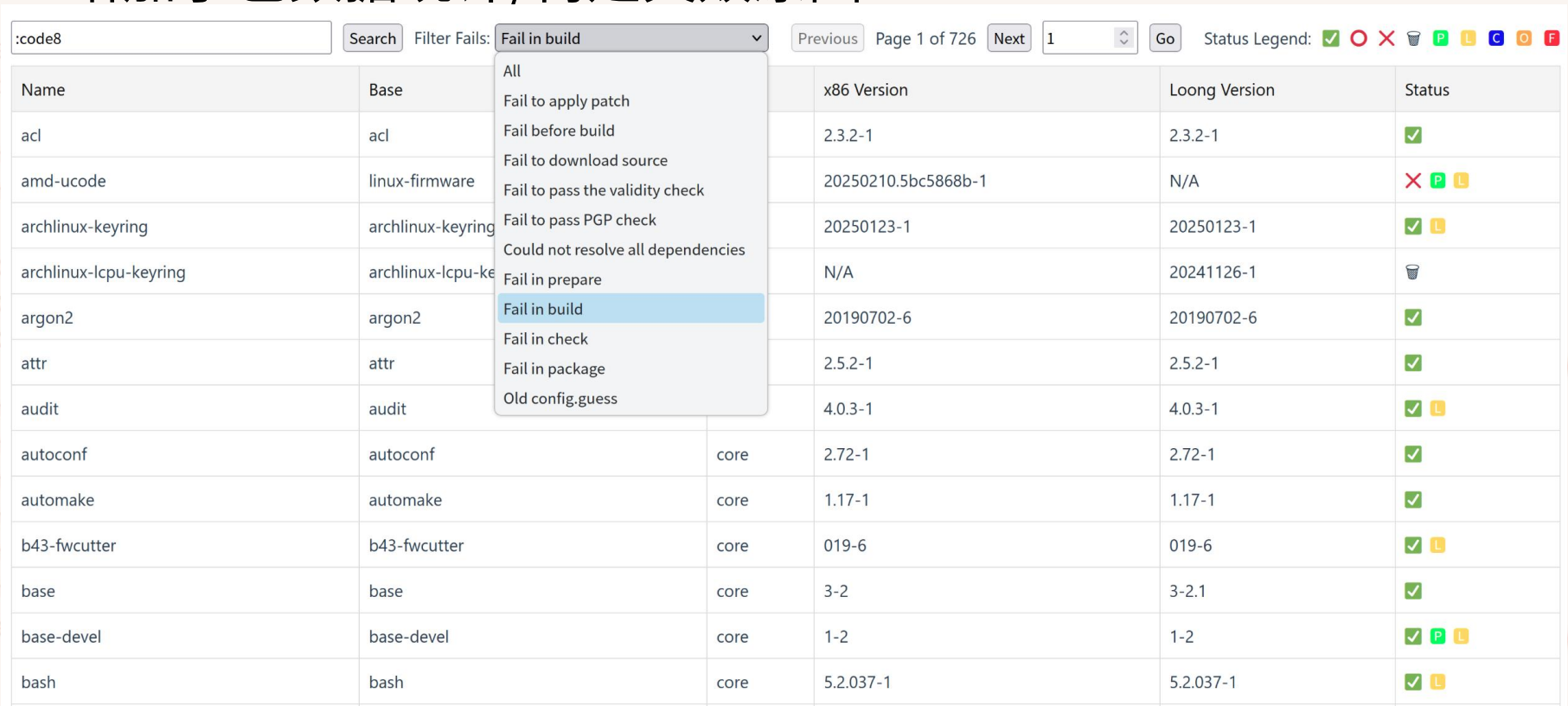
龙架构 LoongArch Biweekly 双周会

Gentoo

- Keywording
  - #949902: qemu 功能测试用
- 打包
  - www-client/firefox 下游 ebuild 修改已稳定良久，已上游化
  - sys-firmware/edk2
    - 无法用 GCC 15 构建（默认 C23 标准），GCC 14 OK（默认 C17）
    - 启用网络引导的 TLS 协议支持时，在极早期即初始化失败
      - 某处→`RAND_seed`→`drbg_seed`→`movgr2fr.d`
      - FPD 例外未被处理，或有违 UEFI 2.6 的龙架构执行环境保障

Arch Linux for Loong64

- 新的网站前端(by Pluto Yang)
  - https://loongarchlinux.lcpu.dev/
    - 重新设计了网站结构和布局
    - 增加了包数据统计/构建失败原因



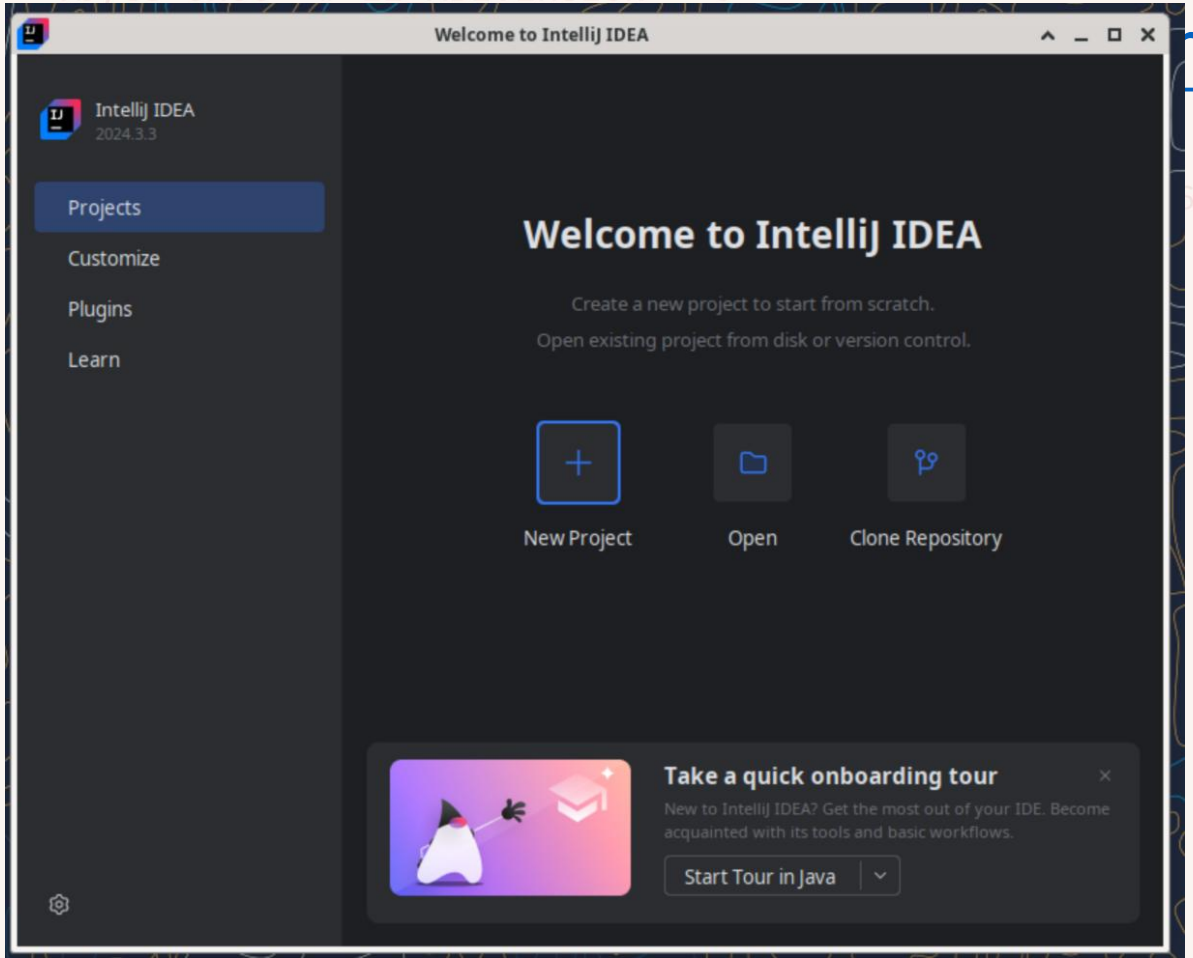
龙架构 LoongArch Biweekly 双周会

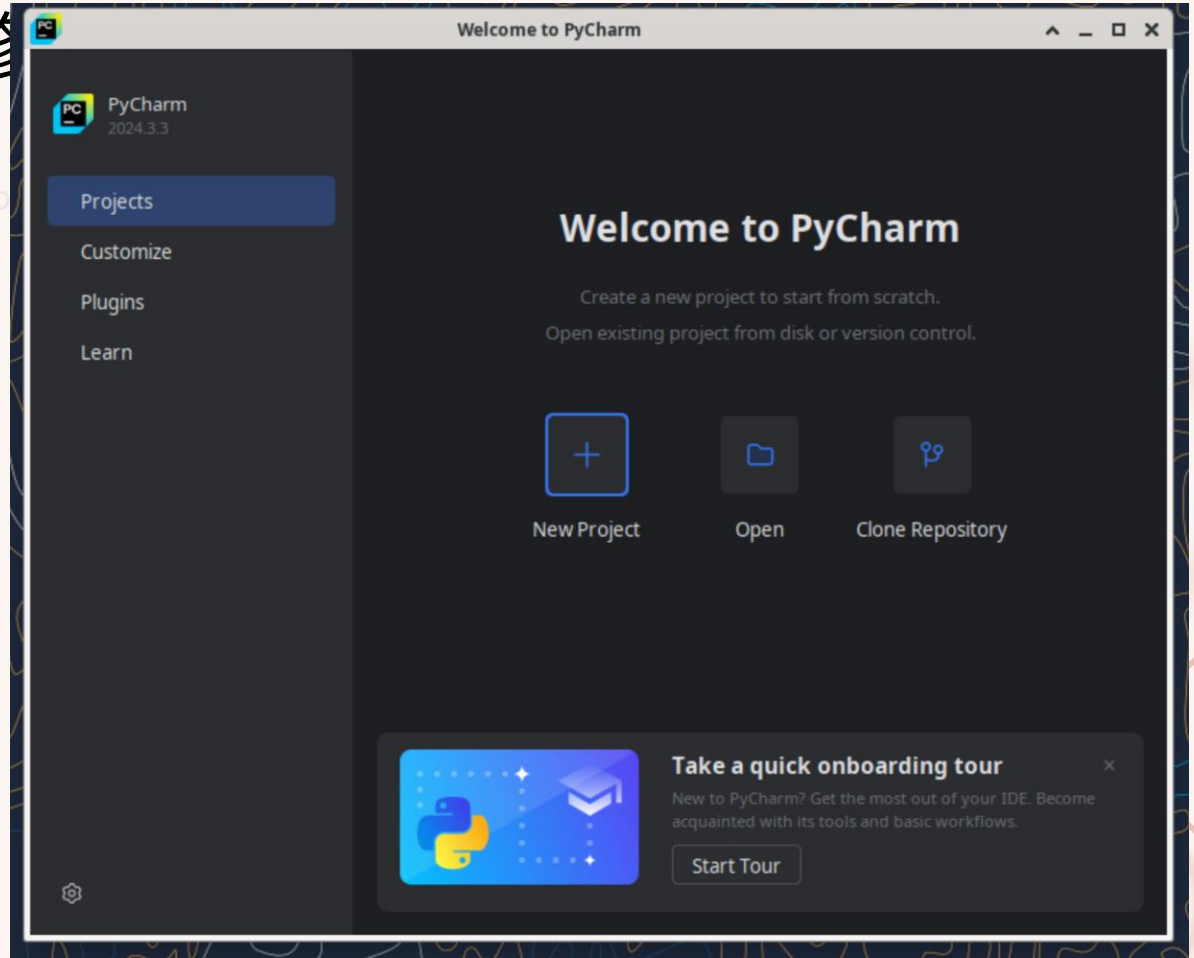
Arch Linux for Loong64（续）

- icu完成升级，除个别大型软件包以外重打完毕(by Pluto Yang)
- 使用gcc打包hsa-rocr时会缺少头文件mm_malloc.h，尚不清楚原因，可以与wszqkzqk讨论（暂时改为clang构建）
  - https://github.com/lcpu-club/loongarch-packages/pull/440
- 4k内核维护考量：部分软件兼容性需求
  - QEMU USER/Box64在16k下仍有尚未修复的bug
  - 部分硬件可能有需求
  - 可能风险：稳定性？新的兼容性？4k下构建的部分软件在16k下无法运行
- pkgstats统计用户安装的软件包列表
  - https://www.loongbbs.cn/d/190-arch-linux%E4%BD%BF%E7%94%A8%E6%8A%80%E5%B7%A7pkgstats%E7%BB%9F%E8%AE%A1%E4%BF%A1%E6%81%AF

Arch Linux for Loong64（续）

- JetBrains IDE：已支持IntelliJ IDEA与PyCharm社区版(by wszqkzqk)





Arch Linux for Loong64（续）

- rocm-llvm等rocm软件包打包完成(by wszqkzqk)
- 目前Arch Linux官方支持的所有rocm包均已完成打包且全部启用上游支持的配置/特性
- 部分支持rocm的包已启用支持
  - python-pyopencl
  - ginko-hpc
  - magma
- Ollama: 之前的修复失效，暂未重打，故尚未集成rocm支持
- 无可用硬件，尚未测试（内核方面支持情况未知）

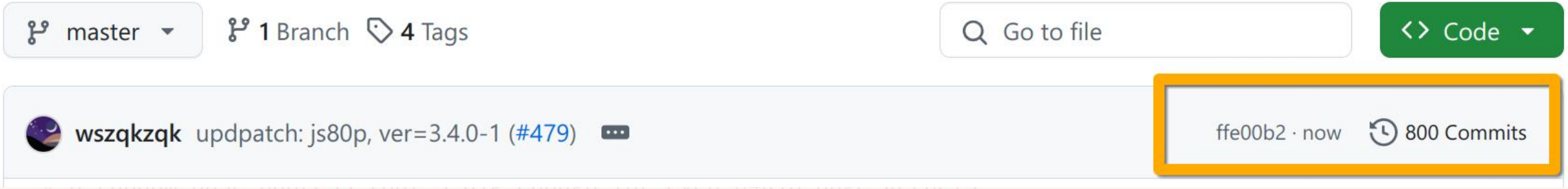
龙架构 LoongArch Biweekly 双周会

Arch Linux for Loong64（续）

- rocm-llvm的问题修复 (by wszqkzqk)
  - fp16支持：基于无fp16支持的llvm 18，需补充
  - rocm-llvm下编译器生成的对象（无论是否LTO）无法使用LLD链接（无论是系统的还是rocm-llvm中的）（未排查原因）
    - 依赖链：amdclang -> libc++（要求用lld链接）
    - 补充信息1：这些对象可以用bfd/mold链接
    - 补充信息2：LLD可以正常链接其他编译器生成的对象
  - 解决：强制构建时使用mold & 指定rocm-llvm中的编译器默认使用mold链接
    - 使用mold：mold对bfd与lld都有很好的兼容性
    - 不可只设置LDFLAGS/CMAKE_EXE_LINKER_FLAGS，中途会分别切换到bfd/lld
- 补丁：https://github.com/lcpu-club/loongarch-packages/blob/master/rocm-llvm/loong.patch
- 修复历程：https://github.com/lcpu-club/loongarch-packages/commits/master/rocm-llvm/loong.patch

Arch Linux for Loong64（续）

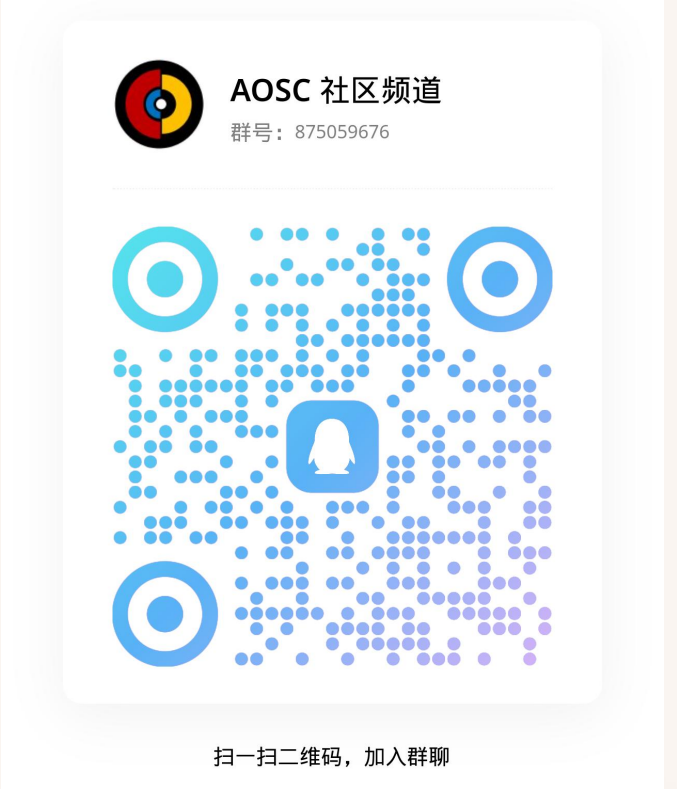
- 补丁仓库总commit数突破800



- 前两大维护者半年的commit数与archriscv项目最活跃贡献者5年commit数接近
  - 其他贡献者commit数相对较少（commit占比 — #1: 53%, #2: 44%）

安同 OS

- 硬件平台支持
  - 推送 Linux 6.13.3/6.12.15 内核更新，整合部分来自 deepin 的厂商补丁 (deepin-community/kernel#599)
    - 补丁集中数个补丁的功能说明不足，已咨询对接同事
    - 对补丁进行了溯源、诊断和查验，可参考此处笔记
  - 修复 3A6000 笔记本（攀升）触摸板不可用的问题
    - 联想 G60d 系列目前用户报告触摸板依然不可用，待查
- Firefox 修缮与新特性
  - 从上游 136 及 137 分支回合了 HEVC 硬件解码支持
    - Bilibili 三大编码支持至此已支持完全
  - 移除 User-Agent 中长期存在的 AOSC OS 字样
    - 避免站点跟踪



龙架构 LoongArch Biweekly 双周会

安同 OS

- 打印机
  - 佳能打印机无法配置使用的问题已解决
  - 奔图打印机实测可以顺利安装、配置和使用，请注意安装 `openssl-1.1` 包提供必要运行时依赖
- LoongGPU 相关
  - 同时插入 7A2000 及独显视频输出时，桌面可能启动失败
  - 将分辨率设置为 800x600 后，KDE 可能无法启动
  - 6.7 及以上版本的 Linux 内核无法正确显示 VT (TTY)
  - 以上问题已与 GPU 组沟通，将继续提供调试信息
- LATX 与 x86 运行时
  - 目前推送的测试版，用户反馈无法启动 Wine 程序
  - 仍在调查中

龙架构 LoongArch Biweekly 双周会

安同 OS

- 4/16KiB 分页双内核
  - 流程及配置方面无问题，暂时保持默认 16KiB 以保障应用兼容性
  - 已请求贡献者同事协助推进 GRUB 方面界面集成，后续可直接从 GRUB 引导界面选择需要的内核分页
- Ren'Py 引擎支持
  - 社区好友 Catty 已在着手移植，测试中
- 3C6000 平台虚拟化支持问题
  - 12 月固件版本的确有问题，会导致虚拟机稳定性问题
  - 经与固件组沟通，暂定下个月推送新固件后复测

龙架构 LoongArch Biweekly 双周会

安同 OS

- QEMU/libvirt 故障两例：
  - QEMU > 9.0.1（版本范围待查）使用 virtio GPU 时，edk2 报告的 GopNativeResolution 为 602x452，导致不满足 edk2 中分辨率 >= 80x25 字符模式的断言，导致固件崩溃

    GraphicsConsole video resolution 602 x 452
    ASSERT [GraphicsConsoleDxe]
    /home/kraxel/projects/qemu/roms/edk2/
    MdeModulePkg/Universal/Console/GraphicsConsoleDxe/
    GraphicsConsole.c(267): (MaxColumns >= 80) && (MaxRows >= 25)

  - libvirt >= 10.10.0 默认 cpu_check="partial"，但龙架构平台未实现 Host CPU model passthrough，导致 libvirt 崩溃
    - 该问题已在上游修复，尚未发版，可按需摘补丁修复

龙架构 LoongArch Biweekly 双周会

安同 OS

- 安全更新与用户公告
  - **PostgreSQL 13.19、Open H.264 2.6.0 均包含高危漏洞修复；X.Org Server 21.1.16 及 XWayland 24.1.6 包含安全漏洞修复**
  - NetworkManager 1.50.0 可能无法通过 DHCP 获取 IPv4 地址，该问题已推送修复，如遇到问题请确保组件版本为 1.50.2
  - 自动镜像跳转服务 (redir.aosc.io) 开放测试
    - 通过 oma mirror 命令选中如下镜像即可测试
      `AOSC Automatic Redirect Mirror [TESTING] (Global) (redir)`
    - 相信可以提高国内绝大多数地区的软件安装和更新体验
- 建议关注公众号“安同开源”或社区主页 (aosc.io) 新闻

龙架构 LoongArch Biweekly 双周会

专题报告

龙架构 LoongArch Biweekly 双周会

又不是不能用？

浅评商业软件发行乱象

龙架构 LoongArch Biweekly 双周会

一些基本现状

- 众所周知，龙架构的软件生态
  - 需要从头建立（大家辛苦了！）
  - 历史上“新世界”和“旧世界”生态的割裂
  - 目前几乎所有的商业、商用软件**均为“旧世界”应用程序**
  - 除 ABI 区别外，这些软件**假设了对应的“旧世界”系统运行环境**
- 换言之，在“新世界”系统组建工作环境的难度不低
- libLoL 为您打开新世界的大门
  - 大多数的软件都可以兼容
  - 但是依赖和质量问题很多！
    libLoL 提供原理上的解决方案，不解决应用本身存在的问题
  - 实际上，就算在原定兼容的系统上安装“旧世界”应用，
    这些问题也是照样存在的

好像不能用——

- 实际使用中，用户往往会遇到如下这种问题：
  - 为什么在配置好 libLoL 后，在 Debian、安同 OS 等环境解压或安装 QQ 和微信后无法启动？
- 为什么？
  - 二者均依赖“新世界”系统中已经废弃的老版本运行时库
  - QQ：`libcrypto.so.1.1, libssl.so.1.1`
  - 微信：`libcrypto.so.1.1, libssl.so.1.1, libtiff.so.5`
- 诶，这些不都是依赖带进来的吗？
  - 巧了，他们没写——
    - 啊？他们没写，他们不是腾讯吗，怎么连这都不懂

又不是不能用！

- 前面的例子都有具体的解决方法
  - 安装仓库中的老版本运行时库
    - 如安同 OS 下 `oma install openssl-1.1`
  - 找群友求各种库文件，或者自己编译
- 又不是不能用！
  - 上述的例子中的问题都有对应解法
  - “只要有群友教，没什么不可以解决的”
- 但请稍等
  - 这两个案例属于非常容易解决的例子
  - 前面提到的问题出现的原因，大家好奇吗？
  - 这样安装老版本库的做法，有什么后果呢？

龙架构 LoongArch Biweekly 双周会

还真不能用（还有其他例子）

- 某云桌面：自带 sudoers 配置，允许所有用户执行其更新脚本，而更新脚本本质上是对 `dpkg -i "$@"` 的调用
  - **所有用户都可以通过这个脚本对系统进行包管理操作！**
- 某浏览器：使用自带沙箱程序提权，安装自更新包
  - **所有用户都可以通过该沙箱程序提权执行 dpkg 管理操作**
  - **配合特别制作的恶意 .deb 软件包，可以安装恶意软件**
- 某显卡厂商驱动包：使用 `postinst` 脚本，使用通配符复制来自 `/var` 的安装数据缓存
  - 该操作可绕过包管理对文件冲突的检查
  - **在此处摆放关键系统库等可实现恶意文件覆盖**

真的不能用！

- 他到底想说啥啊
  - 目前龙架构的软件生态，尤其是商业软件和来自发行版和企业的“应用商店”仓库中的软件，**普遍缺少最基本的审查和测试**
  - **某些软件正有意无意地将您暴露在巨大的信息安全风险中**
- 本次专题报告的意图
  - 指出龙架构生态中软件质量问题的典型案例（这次就这些）
  - 介绍常见质量、安全与发布管理问题，以期提高认知与警惕
  - 提出有关提高软件发行质量的倡议（及面向发行方的警告）
  - 简述披露、批评与投诉计划与（合法）方式

龙架构 LoongArch Biweekly 双周会

一笔烂账

依赖、文件路径和包管理脚本

龙架构 LoongArch Biweekly 双周会

顽疾杂症综述

- 在测试 libLoL 兼容性数据库中的各类软件和在日常工作中，我发现为龙架构发行的软件包时常存在一些可归类的问题
  - 当然，我相信其他架构很可能也有类似的情况
- 总地来说，大概有这几类
  - 大量依赖失去维护的、漏洞频出的运行时库
  - 依赖编写不严谨、不完整，甚至完全缺失
  - 随意引入系统级提权配置 (sudo, polkit 等) 及工具
    - 亦有随意设置 777 权限位的行为
  - 随意修改用户目录下的配置文件
  - 大量、随意地引入 FreeType、GTK 及 Qt 等公用运行时库
  - 绕过软件包管理部署和移除文件
  - 不使用系统、用户服务设施进行系统状态探测，或随意猜测配置文件内容

老祖宗留下的都有啥？

- 目前，龙架构“旧世界”商用系统运行时版本落后严重
  - 但版本只是表象，基于固定版本的持续维护和修缮也很重要
  - 很可惜，此类维护修缮往往缺席，而各类第三方软件亦对运行时版本和其中附带安全漏洞缺乏重视
- **案例：几乎所有涉及网络通信和加密的软件依然在使用已经失去维护的 OpenSSL 1.1 运行时库**
  - 自 2023 年 9 月停止维护以来，涉及该运行时库的高危安全漏洞已突破两位数大关，而这一数字只会随着时间推移不断递增
  - **从 OpenSSH 到微信、钉钉，OpenSSL 1.1 阴魂无处不在**
  - **迁离 OpenSSL 1.1，换用受维护的密码学库刻不容缓**
- 这是本次报告中唯一一个不提供具体解法和分析的案例
  - 迁离路径和缘由过于清晰，不须多言
  - 兼容性问题规避手段请见后续介绍

依赖，依赖，莫要无赖

- 最常见的一类软件包质量问题
  - 在 Linux 生态常见的动态链接体系下，任何 ELF 动态可执行文件和库都有对应的依赖
  - 许多软件发行商由于种种原因选择不记录依赖，这是绝对错误的
  - 以微信为例，排除其自带的运行时，**对系统库的依赖多达 83 项**
    - 但他们发行的软件包**没有记录任何一个依赖项**
    - 是的，微信的包大概在 UOS 和麒麟装上不会出依赖问题，因为他们的桌面环境很可能已经依赖上了所有相关的库
    - 微信唯独在程序包里带一个 `libvlc.so.5` 系统库也足够证明他们（研发或产品管理人员）可能对软件包依赖编写有抗拒
    - 至于抗拒的理由是什么，我们不得而知
    - **唯一明确的是，这是个问题，因为有相当通用和可靠的方法解决这一问题！**

解：依赖，依赖，莫要无赖

- 实际上，探测动态可执行与库文件的依赖的方法非常简单
  - 甚至乎可以程序化
  - “笨办法”（最通用）：以查阅 /usr/bin/bash 的依赖为例

```
readelf -d /usr/bin/bash | grep ' (NEEDED)'
0x0000000000000001 (NEEDED)  共享库：[libreadline.so.8]
0x0000000000000001 (NEEDED)  共享库：[libc.so.6]
```

  - 此时可得，/usr/bin/bash 依赖 libreadline.so.8 及 libc.so.6 两个动态库
  - 接下来，通过 dpkg -S 等命令即可得出对应库的来源即 libreadline8 及 libc6

龙架构 LoongArch Biweekly 双周会

解：依赖，依赖，莫要无赖

- 使用 Debian 工具集打包 (dpkg-buildpackage, debuild, ...)，依赖探测更为自动化
  - 在 `debian/control` 中使用 `${misc:Depends}`, `${shlibs:Depends}`, `${perl:Depends}` 等关键字即可引导打包工具直接探测和填写相关依赖
  - 详情请参见 Debian 维护者手册，尤其是第四章
- 讲者按：对依赖的理解不应本末倒置
  - **依赖是控制目标系统环境完整性的强力手段**
  - 在软件包发行的语境下，“适应目标系统的运行时有无”是错误的、不假思索地套用 Windows 软件生态思维的结果

提权不是请客吃饭

- 相对少见，但破坏性最大、危险性最高：随意提权
- 常见的错误提权实现有三：
  - 以“助手”(helper) 程序实现无密码提权
  - 为 sudo 和 polkit 等提权工具引入无密码配置
  - 为具体的路径设置 777（全用户/组可读、可写、可执行）权限
- 这样做的理由有什么？以实际软件为例
  - 某浏览器：实现类似 Windows 下的“热更新”功能
  - 某远程管理工具：允许用户写入主题路径，以便安装“皮肤”
  - 某固件管理工具：以为 777 权限的数据才能被所有用户访问
- 对... 对吗？
  - **废话，当然不对**，提权不是请客吃饭，是引狼入室！

提权不是请客吃饭

- 风险点到底在什么地方？
  - **免密码提权若对一个软件生效，那么它一定能给所有人和程序用**
  - 本地提权漏洞几乎是默认高危的，软件主动引入高危漏洞，不论动机为何，**都应该立即整改撤销**
- 错在哪里？
  - 对 Linux 系统权限模型、文件系统结构有根本性的认知错误
    - 系统层面上安装的软件包，更新、增减操作必然需要最高权限
  - 在自动索取最高权限时没有询问用户和说明风险点
    - **这就是恶意软件的基本原理（尽管您可能不是故意的）**
  - **再次：主动引入免密提权入口，就是在主动引入安全漏洞**

解：提权不是请客吃饭

- 解决此类问题的方法非常多，这里列出几个恰当的惯例
  - **在热更新前**提示更新并索要管理员密码进行提权
  - 通过软件商店等图形前端，**搭配合理的提权警示**引导用户更新
  - 安装软件包时引入对应的 APT 等软件源配置文件，用户方可使用 apt/oma 等前端，搭配系统更新进行更新
- 讲者按：提权不是请客吃饭，请客吃饭也得问主人意愿！
  - **提权意味着什么？意味着索要用户对软件的绝对信任**
  - 绕过用户感知直接索取最高权限
    - 用户也许觉得方便，但放任系统权限管理给第三方软件本身就是对个人信息安全的漠视
    - **对于软件发行商来说，未经允许骗取用户的信任就是恶意行为，<u>没有任何讨论余地</u>**

不是所有的关怀都叫爱

- 是的，Linux 下的软件包和依赖概念令人陌生
  - 设计不好甚至令人害怕，所以要合理设计
- 是的，Linux 发行版之间有不同的运行环境，可能不兼容
  - 如果不带好经常不兼容的库（比如频繁破坏 ABI 兼容性的 FFmpeg、特性不同导致符号不同的 libjpeg-turbo 等），导致安装或使用困难
  - 前面提到的微信就是一个妥善处理的例子（虽然没写依赖）
- 如此贴心的软件发行商很多，但往往找错了办法
  - 许多软件发行商将所有“可能用得上”的系统库都带到了软件目录下，而没有通过正确设置链接器参数配置好优先级，导致系统库和应用库混用，破坏兼容体系的自洽性
  - 也有一部分发行商根本不知道具体需要什么，依靠“实验物理学”在不可控的环境下随意复制并附带一部分库，导致运行时不通用

解：不是所有的关怀都叫爱

- 解决这一问题并不简单，也没有捷径
  - 首先，**调查清楚、明确规定**软件的兼容范畴，进而决定是否需要前面提到的内包 (bundle/vendor) 库等兼容性规避手段
  - 如决定要使用上述手段，**以最小化原则**确定兼容性风险较大的运行时库，并将其设为内包，并在链接过程中**确保其最高优先级**
  - **在一切完成后，确保使用前面提到的依赖查验和标记法，正确地标记软件依赖**
  - 恭喜您，现在您的软件内容合理且兼容性实现了最大化！
- 讲者按：用户关怀如同对用户的友情，但时刻讲究方法
  - 目前市面上的商业软件往往在内包运行时的做法上没有投入足够的原理性调查，只以业务完成作为主要目标，必将弄巧成拙
  - 正确认识依赖和兼容性的性质与需求才能最大化双方效率和利益

大灰狼都不敢这么开门见山

- 小红帽，快开门，我要改你的配置文件！
  - **可别，这太下头了**
- 令人大跌眼镜的“软件配置迁移工程”
  - 通过包管理，在提权状态下直接篡改用户数据与配置文件
  - 需求一般有二：
    - 之前特性实现不符合预期，希望用户尽快用上新的特性
    - 强烈推荐用户关闭某个特性或调整某个配置，但难以通知用户
- 这是原则问题：**别碰用户的文件**
  - 用户目录如同私人宅宇，神圣不可侵犯，经过提权的包管理更不应该利用这一条件去随意篡改用户的私人文件
  - **如果确实有需要用户介入的修改的配置，请妥善通知或推送修复**

解：大灰狼都不敢这么开门见山

- 解决这一问题的方法非常多，唯独别碰用户的文件
  - 技术原则与思路
    - 引入默认配置模板系统，甚至可以引入配置版本体系，自动对**未经修改**的用户配置项目进行迁移
    - 引入新的配置文件路径，并妥善通知用户
  - 沟通原则
    - 软件有任何显著的行为更改，理应通知用户
    - **KPI 压力和控制欲不是不征求用户意见和许可的借口**
- 讲者按：尊重用户边界，切勿诉诸侵略
  - 用户配置某个软件的理由就算不完全理性，也不能被随意忽视
  - 而实际上，如果一个软件的配置行为迁移需要依赖包管理触碰用户目录的内容，**那么您的方案从原则到技术方法上都大错特错，切勿执拗，立刻整改**

此路不通，故走后门?

- 软件包管理的核心功能
  - 定义各组件的依赖关系，可靠地重现必要的运行环境
  - 记录软件包中个附带的文件、属性等信息
  - **确保软件包内容与依赖关系自洽而无冲突**
- 其中第三点是经常被驱动包有意规避的功能
  - 通过在 `/var` 等地方存放文件，通过 `postinst/postrm` 等包管理脚本部署和卸载文件
  - 可能是因为驱动包中附带了多种型号的驱动文件？动机令人迷惑
- 有何风险?
  - 许多此类脚本并未定义确切的、需要复制部署的文件列表
  - 在某个目录中可以放置任意文件，上述脚本便会“忠实”地将文件复制到位，不论后果... **能猜到可以怎么搞坏系统吗？太容易了**

解：此路不通，故走后门？

- 不论何种解法：**请勿规避包管理的文件管理功能！**
  - 最小化文件操作：通过包管理脚本确定需要登记部署的 DKMS 模块
  - “文件替代”机制：使用 `update-alternatives(1)` 进行冲突的文件的取代和优先级设置，将候选文件放置在软件包中的候选路径下，并使用 `postinst/prerm` 脚本操作**精确到文件的**选择
  - **注意：上述两种方法不涉及软件包文件的复制和增减，但需通过 postinst/prerm 脚本辅助精确的模块/文件注册与选择、部署**
- 讲者按：正道光明坦荡，何苦偏走险路？
  - 实际上，目前成熟的包管理（尤其是国产桌面系统常用的 dpkg）都具有完备的复杂文件操作工具，只需耐心寻找方案便可
  - **软件包管理就是为了降低组件间文件管理的成本而生的**，何苦拒绝它的好意呢？

请讲普通话

- 今日介绍的最后一个常见问题类别
  - 一部分安全审查和运维类软件似乎并不屑于利用系统服务管理和各类服务侦测设施（如 systemd、UNIX socket 等）
  - 它们似乎对“文件级分析”有什么奇怪的癖好，且不说其准确度奇差
- 某司“计算机终端检查系统”便是典型案例
  - 其检查“网络合规性”时要求禁用“文件共享”功能，其中指向了 sftp 及 Samba 服务
  - 用户报告无论如何禁用服务都不能满足检查需求
  - 最后发现“检查系统”查验的并非服务状态，**而是配置文件中是否带有对应的配置字样，甚至忽略注释状态**
  - 这玩意能靠谱吗？反正我是不信了（何苦呢？）

解：请讲普通话

- 解法：请充分了解当今系统的管理方式
  - 2025 年，绝大多数的系统都使用 systemd 服务，搭配相关端口 (Socket) 或文件描述符 (File Descriptor) 提供服务
  - 最不济，也还有进程管理信息
- 引申：标准化实现利人利己
  - 小到桌面图标，大到网络架构，**遵从开放和普遍标准是降低开发和用户双方使用成本，最大化可靠性的不二法则**
  - 请耐心阅读和理解规范，并根据需要实现相关接入支持，并在有必要时参与规范反馈和修订工作，利人利己
- 讲者按：请讲普通话，不能浮于形式
  - 正如语言的通用性有赖于一定的使用方法约定，对标准的理解同样**需要深入结合应用，更要驱动对功能的验证**

我们是最后的防线

给各位软件发行商的留言

龙架构 LoongArch Biweekly 双周会

基础软件之苦与重

- 本人涉足基础软件行业近 15 年，充分理解此行业的不易
  - 但也正因为如此，我希望让大家的工作能够达到最好的效果，让用户满意，并最大化自己时间和精力投入的回报
- 基础软件质量工程重于泰山
  - 作为基础软件研发和分发的第一责任人，我们是推动实现信息系统国产化的骨干力量，也是**防御安全和可靠性漏洞的最后防线**
  - 国产化信息系统可靠性仰靠**对每个组件、应用软件和架构设施质量工程的精益求精和对技术、沟通原则和基准的坚持和贯彻执行**
- 现状有待改观
  - 诚然，前面提到的问题，尽管可笑、严重，责任不全在我们身上
  - 对质量工程的把控不仅需要不懈的努力和踏实的工作，更需要管理人员对行业相关知识培训及劳资分配进行深入了解，并提供必要的资源和支持

请群众监督

发掘问题、报告问题、解决问题

龙架构 LoongArch Biweekly 双周会

以监督促质量

- 龙架构软件生态由数千款商用软件组成
  - 质量问题频繁出现、形式多样，前文介绍了普遍存在的问题案例
  - **作为用户，倡议各位提高警惕，并行使必要权利**
- 反馈、投诉、解决
  - 虽然软件发行商一般都有能力和设施跟踪用户反馈，但也可以考虑通过主动反馈和投诉加速问题的发掘和解决
- 群众团结监督
  - 在后续双周会中，我会组织“问题软件报告与展示”环节，针对存在显著质量问题的软件进行重点展示与评析，以求提高用户认知与警惕，并引起相关方面重视
  - **原则：先报告后公示，不展示未经解决披露的安全漏洞**

附录：安全漏洞报告（您的义务）

- 在发现如上文中提到的安全漏洞时
  - **切勿披露漏洞相关软件产品名或任何可能导致产品版本与漏洞利用方式配对的信息，并第一时间将漏洞上报工业和信息化部网络安全威胁和漏洞信息共享平台，并等待发行商修复及平台正式披露**
- 根据我国工业和信息化部、国家互联网信息办公室、公安部印发的《网络产品安全漏洞管理规定》第九条：
  - 从事网络产品安全漏洞发现、收集的组织或者个人通过网络平台、媒体、会议、竞赛等方式向社会发布网络产品安全漏洞信息的，**应当遵循必要、真实、客观以及有利于防范网络安全风险的原则**，并遵守以下规定：
  - （一）**不得在网络产品提供者提供网络产品安全漏洞修补措施之前发布漏洞信息**；认为有必要提前发布的，应当与相关网络产品提供者共同评估协商，并向工业和信息化部、公安部报告，由工业和信息化部、公安部组织评估后进行发布。
  - （二）**不得发布网络运营者在用的网络、信息系统及其设备存在安全漏洞的细节情况。**

附录：安全漏洞报告（发行商义务）

- 三部门《网络产品安全漏洞管理规定》第七条：
  - 网络产品提供者应当履行下列网络产品安全漏洞管理义务，确保其产品安全漏洞得到及时修补和合理发布，并指导支持产品用户采取防范措施：
  - （一）发现或者获知所提供网络产品存在安全漏洞后，**应当立即采取措施**并组织对安全漏洞进行**验证，评估**安全漏洞的**危害程度和影响范围；对**属于其**上游**产品或者组件存在**的安全漏洞，应当立即通知**相关产品提供者。
  - （二）应当**在 2 日内向工业和信息化部网络安全威胁和漏洞信息共享平台报送相关漏洞信息。**报送内容应当包括存在网络产品安全漏洞的产品名称、型号、版本以及漏洞的技术特点、危害和影响范围等。
  - （三）应当**及时组织**对网络产品安全漏洞进行**修补**，对于**需要产品用户**（含下游厂商）**采取软件、固件升级等措施的，应当及时**将网络产品安全漏洞风险及修补方式**告知可能受影响的产品用户，并提供必要的技术支持。**

立即评估、告知上游

2 日内报送

及时修补、
告知受影响下游

和你在一起

咨询与技术支持

龙架构 LoongArch Biweekly 双周会

共同进步

- 如果您是龙架构及各类基于 Linux 平台的软件发行商且对软件质量、发布工程相关话题有任何疑问、意见和需要
  - 欢迎您与我取得联系，我将在力所能及范围内提供协助

微信：mingcongbai
邮箱：jeffbai@aosc.io

龙架构 LoongArch Biweekly 双周会

问答环节

社区问答及意见反馈

龙架构 LoongArch Biweekly 双周会

社区问答

- 龙芯处理器的 JTAG 调试标准社区主要通过逆向工程了解之（还发现了违反 IEEE 标准的定义），请问这部分有无像 ARM、RISC-V 一样规范化、公布的计划?
- 龙芯目前在OpenOCD上做的适配工作会开源或者上游化吗?

龙架构 LoongArch Biweekly 双周会

双周会讨论（请先添加管理员）

爱好者交流群

龙架构 LoongArch Biweekly 双周会