

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

```
.section ".blob", "aw", @progbits
```

```
filestart:
```

```
# e_ident
```

```
.ascii "\177ELF"
```

```
.byte 0x02 # ELFCLASS64
```

```
.byte 0x01 # ELFDATA2LSB
```

```
.byte 0x01 # EV_CURRENT
```

```
.byte 0x00 # ELFOSABI_NONE
```

```
.byte 0x00 # EI_ABIVERSION = 0
```

```
.rept
```

```
.byte
```

```
.endr
```

龙架构双周会

```
# a random base address that's big enough for even 64KiB-page kernels
```

```
.set base_addr, 0x200000
```

2025年7月20日·第16次

```
.short 0 # ET_EXEC
```

```
.short 0x102 # EM_LOONGARCH
```

```
.word 1 # e_version = 1
```

```
.dword base_addr + entry - filestart # e_entry
```

```
.dword phdr - filestart # e_phoff
```

```
.dword 0 # e_shoff
```

```
.word 0x41 # objabi v1, soft-float
```

```
.short ehsize # e_ehsize
```

```
.short phentsize # e_phentsize
```

```
.short 1 # e_phnum
```

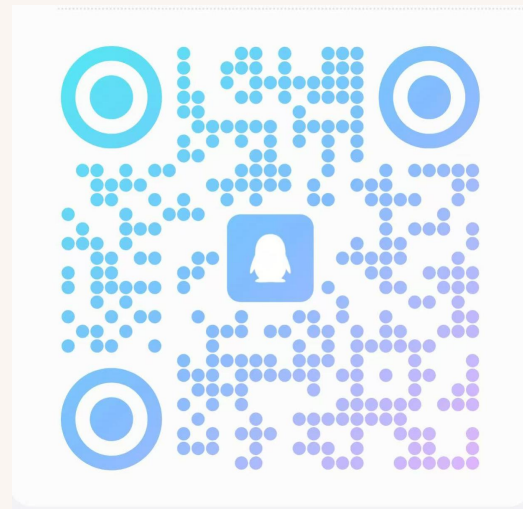
```
.short 0 # e_shentsize
```

```
.short 0 # e_shnum
```

```
.short 0 # e_shstrndx
```

```
.set ehsize, . - filestart
```

```
phdr:
```



龙架构 LoongArch
Biweekly
双周会

```
.section ".blob", "aw", @progbits
```

```
filestart:
```

```
# e_ident
```

```
.ascii "\177ELF"
```

```
.byte 0x02 # ELFCLASS64
```

```
.byte 0x01 # ELFDATA2LSB
```

```
.byte 0x01 # EV_CURRENT
```

```
.byte 0x00 # ELFOSABI_NONE
```

```
.byte 0x00 # EI_ABIVERSION = 0
```

```
.rept 7
```

```
.byte 0
```

```
.endr
```

会前注意事项

```
# a random base address that's big enough for even 64KiB-page kernels
```

```
.set base_addr, 0x200000
```

```
.short 2 # ET_EXEC
```

```
.short 0x102 # EM_LOONGARCH
```

```
.word 1 # e_version = 1
```

```
.dword base_addr + entry - filestart # e_entry
```

```
.dword phdr - filestart # e_phoff
```

```
.dword 0 # e_shoff
```

```
.word 0x41 # objabi v1, soft-float
```

```
.short ehsize # e_ehsize
```

```
.short phentsize # e_phentsize
```

```
.short 1 # e_phnum
```

```
.short 0 # e_shentsize
```

```
.short 0 # e_shnum
```

```
.short 0 # e_shstrndx
```

```
.set ehsize, . - filestart
```

```
phdr:
```

本页预定讲者

龙架构 LoongArch
Biweekly
双周会

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

```
.section ".blob", "aw", @progbits
```

```
filestart:
```

```
# e_ident
```

```
.ascii "\177ELF"
```

```
.byte 0x02 # ELFCLASS64
```

```
.byte 0x01 # ELFDATA2LSB
```

```
.byte 0x01 # EV_CURRENT
```

```
.byte 0x00 # ELFOSABI_NONE
```

```
.byte 0x00 # EI_ABIVERSION = 0
```

```
.rept 7
```

```
.byte 0
```

```
.endr
```

快速报告

```
# a random base address that's big enough for even 64KiB-page kernels
```

```
.set base_addr, 0x00000000
```

龙架构上游动向

```
.short 2 # ET_EXEC
```

```
.short 0x102 # EM_LOONGARCH
```

```
.word 1 # e_version = 1
```

```
.dword base_addr + entry - filestart # e_entry
```

```
.dword phdr - filestart # e_phoff
```

```
.dword 0 # e_shoff
```

```
.word 0x41 # objabi v1, soft-float
```

```
.short ehsize # e_ehsize
```

```
.short phentsize # e_phentsize
```

```
.short 1 # e_phnum
```

```
.short 0 # e_shentsize
```

```
.short 0 # e_shnum
```

```
.short 0 # e_shstrndx
```

```
.set ehsize, . - filestart
```

```
phdr:
```

龙架构 LoongArch
Biweekly
双周会

本页预定讲者

Firefox vs. 2K3000

• 在 2K3000 (暂时以 EFI Framebuffer 作为显示输出) 测试

Firefox 时发现其直接崩溃

- 查明是 [编译 Skia 代码时无条件打开了 LASX](#)
- 改为仅仅对 *_lasx.cpp 使用 LASX, 其他代码会先判断机器是否有 LASX 再调用其中的代码
- 已经合并到 Firefox 主分支 (对应 142 版本), 正在申请向前回合

Firefox vs. 2K3000

- 在审查 AOSC Firefox 补丁集时，发现其带有一个针对内包的 libyuv 的补丁，与 libyuv 上游最终采用的版本不一致
- 进一步审查[上游版本](#)时，十分惊讶地发现其居然也无条件打开了 LASX，很可能导致未来发布的 Firefox 版本崩溃
- 作者的理由是“LASX 在多数龙架构桌面机器和服务器均可用，且性能提升十分明显”

这合理吗？

- [《Software Development and Build Convention for LoongArch™ Architectures》手册](#)难道是空气吗？
- 进一步阅读代码，发现 libyuv 有 4 个构建系统，它们对 LSX/LASX 的处理堪称群魔乱舞
- 目前已经向 libyuv 上游（Chromium）提出[撤销](#)前述的无条件打开 LASX 的补丁，并对 LSX/LASX 的使用[加入](#)运行时检查

libyuv 构建系统和龙架构向量扩展

| 构建系统 | 修复前行为 | 修复后行为 |
|----------------------|--|---|
| GN (Chromium 使用) | 需要人工指定是否打开向量扩展，但构建系统逻辑有问题，打开会导致向量扩展代码被编译，但完全不被调用 | 需要人工指定是否打开向量扩展，如打开则构建向量加速代码，运行时根据 CPU 特性判断是否使用 |
| GYP (Firefox 使用) | 自动打开向量扩展，需人工关闭，否则会在不支持该扩展的 CPU 崩溃 | 在使用 Clang ≥ 18 或 GCC ≥ 15 时自动打开向量扩展，如打开则构建向量加速代码，运行时根据 CPU 特性判断是否使用 |
| CMake (发行版单独构建使用) | 同 GYP，但判断了汇编器是否支持向量扩展 | 在汇编器支持向量扩展时自动打开，运行时根据 CPU 特性判断是否使用 |
| Makefile | 不使用向量扩展 | 自动打开向量扩展，运行时根据 CPU 特性判断是否使用 |

Firefox vs. 2K3000

- 另外代码库中已经存在检查硬件是否支持 LSX/LASX 的代码（尽管根据上一页的讨论，它在之前完全没有意义），但逻辑有问题：应该照着手册用 [HWCAP](#)，而不是 CPUCFG
 - 不然硬件支持向量，但内核不支持的时候还是会吃 SIGILL
- 总结：无论是龙芯员工还是社区开发者，都要按规范办事，不能说为了自己 KPI 好看或者自己的设备跑得快就乱来
 - [“There weren't any bad guys with black hats, they were all people just trying to get by, doing the best they could.”](#)

GCC

- 修复了上次会议提到的某项龙架构测试触发 ICE 的问题
(负责 [PR101882](#) 的上游开发者否认此问题与他相关，故新建了 [PR120983](#))
 - 但是目前的共识是这并不是最好的解决方案，后续上游维护者如果有空的话可能会翻修
- 为主分支（对应 GCC-16 版本）修复了某其他架构开发者报告的龙架构上自动向量化 (-ftree-vectorize) 导致 FFmpeg 的 APE 解码器输出无意义噪声的问题 ([PR121064](#))，后续要回合到 14、15 分支
 - 这样该开发者就能为 FFmpeg 启用自动向量化了，说不定对龙架构也有性能提升

Binutils

- 撤销了上次会议提到的，在 relax pass 2 估算距离时不再考虑代码节对齐的修改，因为此项修改导致大量软件包链接失败
- 此项修改是 xry111 使用脑内虚拟机论证的，应予批判
- binutils 2.45 预计于 7 月 27 日发布，包含上述修复

```
.section ".blob", "aw", @progbits
```

```
fi
```

```
# e_ident
```

```
.ascii "\177ELF"
```

```
.byte
```

```
.byte
```

```
.byte
```

```
.byte
```

```
.byte
```

```
.rept
```

```
.byte 0
```

```
.endr
```

```
# a random base address that is big enough for a 4KiB-pipe kernel
```

```
.sec
```

```
.short 2 # ET_EXEC
```

```
.short 0x102 # EM_LOONGARCH
```

```
.word 1 # e_version = 1
```

```
.dword base_addr + entry - filestart # e_entry
```

```
.dword phdr - filestart # e_phoff
```

```
.dword 0 # e_shoff
```

```
.word 0x41 # objabi v1, soft-float
```

```
.short ehsize # e_ehsize
```

```
.short phentsize # e_phentsize
```

```
.short 1 # e_phnum
```

```
.short 0 # e_shentsize
```

```
.short 0 # e_shnum
```

```
.short 0 # e_shstrndx
```

```
set ehsize, . - filestart
```

```
phdr:
```

Linux Kernel

- Binbin Zhou [提交了](#)之前几次会议提到的 3A6000 笔记本（联想开天除外）的触摸板问题修复的一半
- 另一半存在争议
 - 旧版 ([v4.0.1](#)) 的《龙芯 CPU 统一系统架构规范》参照 OpenFirmware Device Tree 的写法，在 GPIO 设备描述中使用 `gsi_idx_map` 属性表示 GPIO pin 和中断号的对应关系
 - 但是这和 ACPI 的规定不一致，因此新版 ([v4.1](#)) 的手册删除了这一属性，改用 ACPI 规定的写法
 - 如果上游内核不支持旧写法，存量硬件怎么办？
 - 请求厂商升级固件存在严重的非技术困难
 - 另外需要思考：新硬件就没事了吗？

Linux 内核 (loongarch 列表)

- Huacai Chen

- 将 `relocate_new_kernel_size` 定义为 `.quad` (64 位) 类型值 ([第 1 版](#))

- Chenghao Duan

- 为龙架构 BPF 子系统实现函数蹦床 (trampoline) 支持 ([第 3 版](#))

- Haoran Jiang

- 修复 BPF 系统中的两处尾呼叫 (tail call) 问题 ([第 1 版](#))

- Bibo Mao

- 使用 ELF 程序头实现内核物理内存地址探测 ([第 1 版](#))
- 实现 3C6000 的部分 KVM 平台特性探测 ([第 1 版](#))
- 增强 KVM 中对 EIOINTC 中断控制器的模拟功能 ([第 6 版](#))

Linux 内核 (loongarch 列表)

- Yuri Norov

- 简化部分 KVM 代码逻辑 ([第 1 版](#))

- Tiezhu Yang

- 为 BPF 子系统的函数蹦床 (trampoline) 实现 struct ops 支持 ([征求意见第 1 版](#))

- Binbin Zhou

- 龙芯 2K0500 BMC 支持 ([第 8 版](#))

- 龙芯 2K0500 SDIO 控制器支持 ([第 1 版](#))

- 龙芯 2K0500/2K1000/2K2000 MMC 控制器支持 ([第 4 版](#))

PPSSPP LoongArch64 JIT

- 为 PPSSPP 项目添加了完整的 LoongArch64 JIT
- 包括 Vertex-jit (顶点 JIT) 和 IR-jit, 后者支持 2K0300 等 no-LSX 设备
- 支持 LSX 的设备可获得额外优化 (QuickTexHash/Vec4)
- 绝大部分游戏都获得了 2 倍左右的性能提升 (e.g. Mortor Storm)
- 感谢 @phorcys 和 @KatyushaScarlet 提供的测试和 Profile 支持
- 预计将在 v1.20 版本发布
- 欢迎使用/捉虫!


```
.section ".blob", "aw", @progbits
```

```
filestart:
```

```
# e_ident
```

```
.ascii "\177ELF"
```

```
.byte 0x02 # ELFCLASS64
```

```
.byte 0x01 # ELFDATA2LSB
```

```
.byte 0x01 # EV_CURRENT
```

```
.byte 0x00 # ELFOSABI_NONE
```

```
.byte 0x00 # EI_ABIVERSION = 0
```

```
.rept 7
```

```
.byte 0
```

```
.endr
```

快速报告

```
# a random base address that's big enough for even 64KiB-page kernels
```

```
.set base_addr, 0x00000000
```

龙架构发行版变动

```
.short 2 # ET_EXEC
```

```
.short 0x102 # EM_LOONGARCH
```

```
.word 1 # e_version = 1
```

```
.dword base_addr + entry - filestart # e_entry
```

```
.dword phdr - filestart # e_phoff
```

```
.dword 0 # e_shoff
```

```
.word 0x41 # objabi v1, soft-float
```

```
.short ehsize # e_ehsize
```

```
.short phentsize # e_phentsize
```

```
.short 1 # e_phnum
```

```
.short 0 # e_shentsize
```

```
.short 0 # e_shnum
```

```
.short 0 # e_shstrndx
```

```
.set ehsize, . - filestart
```

```
phdr:
```

龙架构 LoongArch
Biweekly
双周会

本页预定讲者

Debian

- DebConf 25 于 7 月 19 日闭幕
 - 10 名中国大陆与会者，1 名澳门与会者，5 名台湾与会者
- Debian 13 (trixie) 将于 8 月 9 日发布
- Debian 发版团队对社区/龙芯公司的 Debian 13 非官方复刻计划表示认同
 - 计划：由龙芯公司与社区同志各自搭建一套 Debian 包构建设施，复现 trixie 软件包，交叉验证（绝大部分包的构建结果可复现），以保证工程质量、获得用户信任
- loong64 于 Debian 14 (forky) 伊始晋级为正式架构
 - 现场展示了 XC-LS3A6M 主板与 3A6000 笔记本，群众反应好
 - 板子借给韩国内核开发者做龙架构的密码学操作优化了

安同 OS (AOSC OS)

- 推动龙芯 2K3000 支持中
 - 为 Linux 6.14 内核 回合了 2K3000 网卡支持
 - 排查应用兼容性
 - Firefox 的 libyuv 及 Skia 组件存在 LASX 指令集，导致 SIGILL，推送 140.0.4 更新时将修复该问题
 - 腾讯 QQ 可用、WPS 可用，微信暂时不可用（LASX 指令）
 - LATX 暂不可用，需提供无 LASX 版本
 - 目前正进行其他功能支持及兼容性的摸排工作
 - LG200 显卡驱动尚未到货，等待中
- Linux 6.16-rc6 测试内核预计今日推送
oma install linux+kernel+rc
 - 刷新了各类外设支持补丁：BMC、Security Engine 等
 - 包含来自 6.6 商业系统内核的 3C6000/S 虚拟化稳定性修复

本页预定讲者



AOSC 社区频道

群号：875059676



扫一扫二维码，入群聊

龙架构 LoongArch
Biweekly
双周会

安同 OS (AOSC OS)

• 本期安全更新

• **Git 2.50.1**

- 修复五处高危漏洞，包含两处远程代码执行漏洞 (CVE-2025-48385, CVE-2025-48386)

• **OpenJDK 安全更新 (2025 年 7 月)**

- 修复 Oracle 重要更新公告 (2025 年 7 月) 中披露的逾 10 个安全漏洞 (含多个高危漏洞)

• **Poppler 25.07.0 安全更新**

- 修复一处远程拒绝服务攻击漏洞 - CVE-2023-34872 (严重性: 中)

• **7-Zip 25.00**

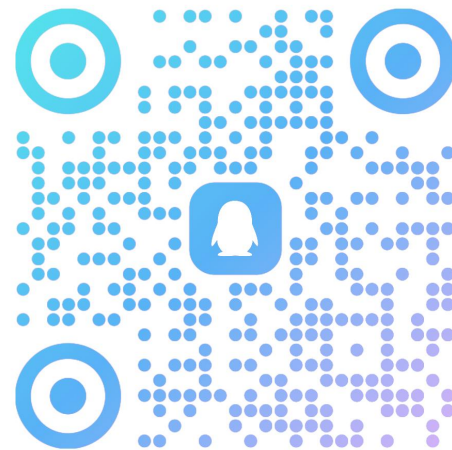
- 修复一处堆缓冲区溢出漏洞，漏洞编号 CVE-2025-53816 (严重性: 中)
- 修复一处空指针解引用漏洞，漏洞编号 CVE-2025-53817 (严重性: 中)

• 建议关注公众号“安同开源”或社区主页 (aosc.io) 新闻



AOSC 社区频道

群号: 875059676



扫一扫二维码，入群聊

龙架构 LoongArch
Biweekly
双周会

Arch Linux for Loong64

- 更新公告制度：便于查验的更新记录
 - 除双周会外，Arch Linux for Loong64将同步在BBS发布更新公告
 - <https://www.loongbbs.cn/d/288-arch-linux-for-loong64-%E6%9B%B4%E6%96%B0%E6%97%A5%E5%BF%9720250706-20250719>
(by NakanoMiku)
- DeepWiki集成
 - AI驱动的代码文档工具
 - 智能文档自动化/AI 交互式支持/视觉化架构图谱
 - <https://deepwiki.com/lcpu-club/loongarch-packages>
 - GitHub Wiki仍然照常人工维护
 - <https://github.com/lcpu-club/loongarch-packages/wiki>
 - 有望降低开发门槛，欢迎大家参与Arch Linux for Loong64开发！

龙架构 LoongArch
Biweekly
双周会

Arch Linux for Loong64

• 镜像相关:

- 原来由武老师维护的原Loong Arch Linux的镜像站均已迁移为北京大学学生Linux俱乐部 (LCPU) 维护的版本。武老师维护的原Loong Arch Linux已彻底结束支持，用户将在更新时切换到现有维护版本。
- PACMAN 镜像扩展 (by Felix Yan, Holger Huo, wszqkzqk)
 - 新增武昌首义学院 (wsyu)、南京大学 (NJU) 和中科院软件所 (ISCAS) 镜像源。其中武昌首义学院镜像的同步间隔为1-1.5小时，其余镜像同步间隔为1天。

Arch Linux for Loong64

• Blender (by wszqkzqk)

- 移除了已被上游接受的 __LP64__ 检测和 Loong64 支持补丁，精简代码维护。

- <https://github.com/lcpu-club/loongarch-packages/pull/656>

• Chromium (by wszqkzqk)

- 重构 Loong64 支持补丁：

- 移除上游已合并的 partition_alloc/rust.gni 等补丁

- 禁用暂时不支持 LoongArch64 平台的 xnnpack for TFLite

- 构建依赖项切换到可以正常工作的 nodejs-lts-jod 避免 nodejs 包出现的问题（详见下）

- <https://github.com/lcpu-club/loongarch-packages/pull/643>

• svelte-language-server & termusic (by Pluto Yang)

- 因上游修复问题，移除临时补丁。

Arch Linux for Loong64

• 全系列内核 (by wszqkzqk)

- 修复 amdkfd (AMD GPU 支持) 在 Linux (6.15+)、Linux-lts (6.12+)、Linux-zen、Linux-hardened 的启动问题, 提升显卡兼容性。

- <https://github.com/lcpu-club/loongarch-packages/pull/652>

• Intel 媒体驱动 & Gmmlib (by wszqkzqk)

- 新增补丁 (媒体驱动 PR / Gmmlib PR), 支持 Loong64 构建并修复 SIMDE_NO_NATIVE 配置冲突。

- <https://github.com/lcpu-club/loongarch-packages/pull/652>

- <https://github.com/lcpu-club/loongarch-packages/pull/638>

• Mesa (by wszqkzqk)

- 启用 Intel 显卡支持并优化编译选项。

- <https://github.com/lcpu-club/loongarch-packages/pull/636>

Arch Linux for Loong64

- SIMDE (by wszqkzqk)

- 回补 PR #1236, 为 SSE2 标头添加 LSX 优化。

- <https://github.com/lcpu-club/loongarch-packages/pull/640>

- Intel 驱动/Gmmlib (by wszqkzqk)

- 重启用 Native SIMD。

- <https://github.com/lcpu-club/loongarch-packages/pull/641>

- devtools-loong64 (by wszqkzqk)

- 增强补丁获取逻辑, 支持从 Pull Request 直接拉取补丁。

- <https://github.com/lcpu-club/loongarch-packages/pull/644>

```
.section ".blob", "aw", @progbits
```

```
filestart:
```

```
# e_ident
```

```
.ascii "\177ELF"
```

```
.byte 0x02 # ELFCLASS64
```

```
.byte 0x01 # ELFDATA2LSB
```

```
.byte 0x01 # EV_CURRENT
```

```
.byte 0x00 # ELFOSABI_NONE
```

```
.byte 0x00 # EI_ABIVERSION = 0
```

```
.repl
```

```
.byt
```

```
.end
```

问答环节

```
# a random base address that's big enough for even 64KiB-page kernels
```

```
.set base_addr, 0x00000000
```

社区问答及意见反馈

```
.short 2 # ET_EXEC
```

```
.short 0x102 # EM_LOONGARCH
```

```
.word 1 # e_version = 1
```

```
.dword base_addr + entry - filestart # e_entry
```

```
.dword phdr - filestart # e_phoff
```

```
.dword 0 # e_shoff
```

```
.word 0x41 # objabi v1, soft-float
```

```
.short ehsize # e_ehsize
```

```
.short phentsize # e_phentsize
```

```
.short 1 # e_phnum
```

```
.short 0 # e_shentsize
```

```
.short 0 # e_shnum
```

```
.short 0 # e_shstrndx
```

```
.set ehsize, . - filestart
```

```
phdr:
```

本页预定讲者

龙架构 LoongArch
Biweekly
双周会

```
.section ".blob", "aw", @progbits
```

```
filestart:
```

```
# e_ident
```

```
.ascii "\177ELF"
```

```
.byte 0x02 # ELFCLASS64
```

```
.byte 0x01 # ELFDATA2L
```

```
.byte 0x01 # EV_CURREN
```

```
.byte 0x00 # ELFOSABI_
```

```
.byte 0x00 # EI_ABIVER
```

```
.rept 7
```

```
.byte 0
```

```
.endr
```

```
# a random base address
```

```
.set base_addr, 0x20000
```

```
.short 2 # ET_EXEC
```

```
.short 0x102 # EM_LOONGARCH
```

```
.word 1 # e_version = 1
```

```
.dword base_addr + entry - filestart # e_entry
```

```
.dword phdr - filestart # e_phdr
```

```
.dword 0 # e_shoff
```

```
.word 0x41 # objabi v1, soft-float
```

```
.short ehsize # e_ehsize
```

```
.short phentsize # e_phentsize
```

```
.short 1 # e_phnum
```

```
.short 0 # e_shentsize
```

```
.short 0 # e_shnum
```

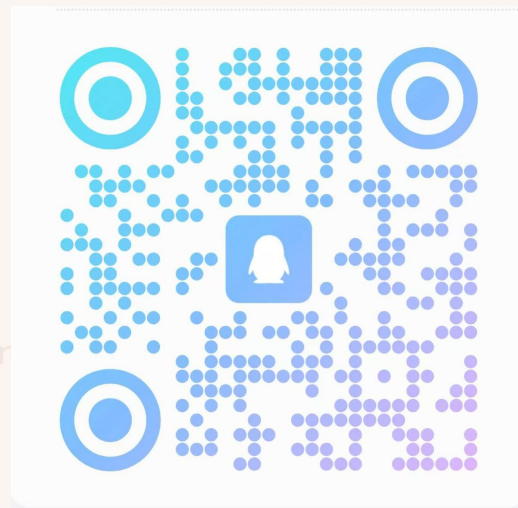
```
.short 0 # e_shstrndx
```

```
.set ehsize, . - filestart
```

```
phdr:
```



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