

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

```
.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
```

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

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

```
.short 0x0000 # e_type
```

```
.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
```

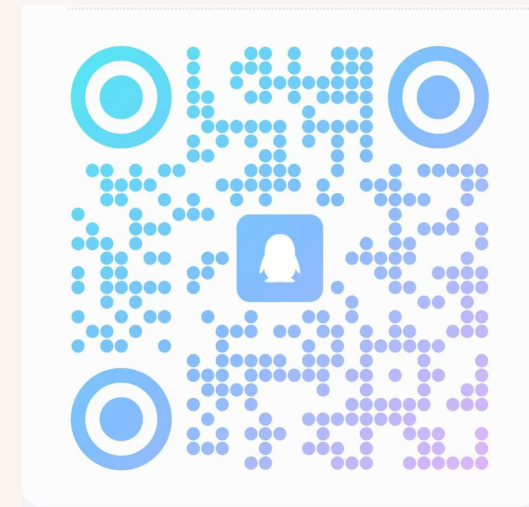
```
.short 0 # e_shentsize
```

```
.short 0 # e_shnum
```

```
.short 0 # e_shstrndx
```

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

```
phdr:
```



龙架构双周会

2025 年 4 月 13 日 · 第 9 次

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
```

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

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

```
.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:
```

快速报告

龙架构上游动向

本页预定讲者

龙架构 LoongArch
Biweekly
双周会

PlatformIO 新增 2K0300 开发板支持

- 社区开发者们为嵌入式软件开发平台 PlatformIO 实现了 2K0300 开发板支持
 - 平台支持: [loongson-community/platform-loongsonla64](https://community.platformio.org/t/new-platform-loongson-la64/46991)
 - 构建框架: [loongson-community/builder-framework-loongarch](https://community.platformio.org/t/new-builder-framework-loongarch/46991)
 - 工具链 (x86-64 交叉): [loongson-community/toolchain-loongarch64](https://community.platformio.org/t/new-toolchain-loongarch64/46991)
- 目前 2K0300 开发板支持已在其社区论坛发布
 - 有待联络上游化工作
 - <https://community.platformio.org/t/new-platform-loongson-loongarch64/46991>

Unicorn 处理器仿真框架

- Unicorn 2.2.0 将包含龙架构支持

- [unicorn-engine/unicorn#1903](https://github.com/unicorn-engine/unicorn#1903)

- 上游维护者联系爱好者社区，希望获得构建/CI 测试硬件支持

- 爱好者社区将为该项目提供一台 3C5000 虚拟机

```
.section ".blob", "aw", @progbits
file:
# e_ident
.ascii "\177ELF"
.byte 0x02 # ELFCLASS
.byte 0x01 # ELFBIN
.byte 0x01 # EV_CURRENT
.byte 0x00 # EI_ABIVERSION = 0
.rept 4
.byte 0
.endr
# 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:
```

GCC

• Lulu Cheng 禁止了 libquadmath 的构建

- 在 long double 本身就是 IEEE 128 位格式的情况下，这个库是 libm 中 long double 相关代码的重复，因此是无意义的浪费
- 而且会使得使用 128 位浮点的 Fortran 代码不必要地依赖这个库
- 触发构建的原因是 LoongArch 为了“兼容存量代码”同时提供了 __float128 和 Q 后缀的支持
- 理论上这里应改为判断 long double 是否是 IEEE 128 位格式，但这样会导致 PowerPC（其 long double 格式可以用命令行参数修改）出现问题，因此目前是针对 LoongArch 做了特判

GCC

- Yang Yujie [提升了](#) genopts 脚本的兼容性（为 FreeBSD 支持做准备）

- 某 BSD 开发者：可以把 genopts 脚本生成的代码放到 build 目录而非 source 目录

- 最初放到 source 目录的目的是允许在没有 GNU awk/sed 的情况下构建，但实际并未成功达到该目的（不然在 BSD 构建也不会触发兼容性问题）

- 现在只需要一个“能用的”awk，而 GCC 文档明确指出构建 GCC 需要 awk，因此后续确实可以这样改

- 据该开发者说已经成功在 BSD 上构建出 GCC 了

Linux 内核 ([loongarch](#) 列表)

- Huacai Chen

- 按需保留（而非重分配）PCI 设备的固件资源 ([第 1 版](#))
- 为 LIONTC（传统 I/O 中断控制器）设置 IRQ_TYPE_EDGE_BOTH 属性 ([第 1 版](#))
 - 是个 hack，《龙芯3A6000处理器用户手册》11.1 节没有提到是否支持配置“边缘触发”到底是上升沿还是下降沿触发
 - 受到了正当质疑，大概不会有结果

- Petr Tesarik

- 删除未使用的 ZONE_DMA 及 MAX_DMA_PFN 引用 ([第 1 版](#))

- Yuquan Wang

- 删除龙架构专属 numa_memblks 转而使用通用实现 ([第 1 版](#))
- 原则上可以接受，降低维护成本，但编译不通过，在看怎么改

龙架构 LoongArch
Biweekly
双周会

Linux 内核 ([loongarch](#) 列表)

• Tiezhu Yang

- 将 fp, lsx, lasx 及 lbt 相关符号导出到架构头文件中 ([第 1 版](#))

- 必要性存疑，待回复邮件

- 改善 kprobe 的健壮性 ([第 1 版](#))

- 带出了一些后续讨论，等翻新

• Tianyang Zhang

- 新增 irq-redirect (IRQ 重定向) 控制器支持 ([第 2 版](#))

• Qunqin Zhao

- 龙芯安全引擎、TPM 设备支持 ([第 7 版](#))

Linux 内核 ([loongarch](#) 列表)

- Binbin Zhou

- 龙芯 2K0500/2K1000/2K2000 MMC 控制器支持 ([第 1 版](#))

- 龙架构平台 PWM 支持 ([第 10 版](#))

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

```
file
```

```
# e_ident
```

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

```
.byte 0x00 # EI_CLASS64
```

```
.byte 0x01 # EI_DATA2LSB
```

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

```
.byte 0x00 # ET_ABIVERSION = 0
```

```
.rept 7
```

```
.byte 0
```

```
.endr
```

```
# 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:
```

龙架构 LoongArch
Biweekly
双周会

Python Wheel 生态自举

- 有同学尝试在龙架构上做学术，需要一套 Python 技术栈

总归还是pip方便啊 11:30 AM

回到这个问题本身, 用, 入乡随俗, 和发行版对...

nump

pip经

😞 官

abi1.0

我觉得

我只知道

就是被骗

你一开始就不需要做这个

那现在我就要手动编译whl了 11:53 AM

龙架构 LoongArch
Biweekly
双周会

Python Wheel 生态自举：社区立场

- “尽量避免在用户的电脑上在 pip install 阶段编译”是合理的
- 虽然不是故意的，但龙芯 PyPI 补充源确实会困扰到新世界用户
- 目前确实缺乏新世界的 PyPI 补充源，很遗憾，这是应当完成的

```
.section ".blob", "aw", @progbits
file_start
# e_ident
.ascii "\177ELF"
.byte 0
.byte 0x01 # EV_CURRENT
.byte 0x00 # EI_ABIVERSION = 0
.rept 7
.byte 0
.endr

# 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:
```

Python Wheel 生态自举：路线图

• 继续推进[龙架构容器化生态的自举工作](#)

- 为了基础镜像: manylinux (glibc linux)、musllinux

- musllinux 基线: 完成, Alpine Linux 3.21

- manylinux 基线: 令人头大

- 《开源生态发展合作倡议》?

- Linux 6.6, glibc 2.38, gcc 12, llvm 17...

- 要么工具链版本过低 (e.g. 没有 LSX), 要么加入一吨重的 backports

- xen0n 正在尝试基于 Gentoo 构建 glibc 2.38 的基础镜像

- Debian 13?

Python Wheel 生态自举：路线图

- 将龙架构支持推入 Python 打包生态上游

- auditwheel: 2025 年 1 月 wangweijie 默默地[增加了](#)支持（6.3.0 版本起）

- 仅新世界。只从代码上看应该不识别旧世界，我没有实测

- uv、maturin、rust-cross/manylinux-cross: [吴小白](#)增加了支持

- <https://github.com/loong64>、<https://loong64.com> (居然重定向到了 Loongnix 官网)

- 没在社区渠道见过这位同学，请冒泡！

- pypi/warehouse (PyPI 官方源): 改为允许 loongarch64 wheels 的上传

- TODO

- TODO: 给关键软件包的上游挨个添加支持并推动发版

```
.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
```

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

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

```
.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:
```

快速报告

龙架构发行版变动

本页预定讲者

龙架构 LoongArch
Biweekly
双周会

Arch Linux for Loong64

- Arch Linux特有工具修复 (by wszqkzqk)
- 集成式安装工具archinstall修复 (by wszqkzqk)
 - 修复grub target的硬编码
 - 修复非x86_64架构下grub因为efi target不匹配安装失败的问题
- **已上游化并得到合并**
- <https://github.com/archlinux/archinstall/pull/3320>
- 高效率包文件列表检索工具pkgfile的修复 (by wszqkzqk)
 - 该工具没有使用libalpm的CARCH机制，而是自己从conf/uname读取
 - 修改：读取到loongarch64时设定CARCH为loong64，否则link 404

Arch Linux for Loong64

- Chromium/Electron修复 (by wszqkzqk)
- 升级Chromium 135，不再需要Swiftshader的额外补丁 (by wszqkzqk)
- 新增Electron 35支持 (by wszqkzqk)
 - 基于Chromium 134补丁与electron 34补丁
 - <https://github.com/lcpu-club/loongarch-packages/pull/547>
- 修复构建Electron 33/34 (by wszqkzqk)
 - 与Arch Linux上游讨论以后改回从npm获取esbuild
 - <https://gitlab.archlinux.org/archlinux/packaging/packages/electron34/-/issues/2>

Arch Linux for Loong64

- Qt 6.9修复 (by wszqkzqk)

- 修复Qt6 Webengine, 不再需要libyuv补丁

- <https://github.com/lcpu-club/loongarch-packages/pull/551>

- LuaJIT/TexLive修复 (by Pluto Yang, Wu Xiaotian)

- 更新TexLive; 使用新补丁修复LuaJIT解决LazyVim运行bug

- <https://github.com/lcpu-club/loongarch-packages/pull/543>

- QEMU依赖更改 (by wszqkzqk)

- 此前一直复用上游依赖, qemu-base依赖qemu-system-x86不合理

- 目前改为依赖qemu-system-loongarch

- 类比x86_64下行为, 额外为qemu-system-loongarch添加固件依赖 (edk2-loongarch64)

- <https://github.com/lcpu-club/loongarch-packages/pull/558>

Arch Linux for Loong64

- AI相关修复

- 修复onnxruntime构建 (by wszqkzqk)

- LSX实现中缺少S8S8/S8U8实现：最初暂时对S8S8/S8U8使用默认kernel

- <https://github.com/lcpu-club/loongarch-packages/pull/563>

- 后续：为LSX本身增加S8S8/S8U8实现的修复：已成功构建并提交上游

- <https://github.com/microsoft/onnxruntime/pull/24397>

- 新启用LASX优化包（-opt）与集成ROCm的包（-rocm），选择矩阵：

- onnxruntime/python-onnxruntime
 - onnxruntime-opt/python-onnxruntime-opt
 - onnxruntime-rocm/python-onnxruntime-rocm
 - onnxruntime-opt-rocm/python-onnxruntime-opt-rocm

Arch Linux for Loong64

- AI相关修复

- 新增aichat支持 (by wszqkzk)

- carogo依赖链: aichat → hnsr_rs → mmap-rs → **nix 0.26**

- 依赖nix版本过低, 不支持loong64

- 修复方式: 对 hnsr_rs PR, 用积极维护的 memmap2 替代 mmap-rs

- 维护者回应较积极但暂未接受

- <https://github.com/jean-pierreBoth/hnswlib-rs/pull/23>

Arch Linux for Loong64

- 为社区引入LLM辅助的尝试 (by wszqkzqk)
 - 支持AIChat后理论上可以方便地集成RAG增强问答
- 利用项目wiki/wszqkzqk博客相关内容建立RAG库，使用LLM解答问题
 - 帮助新参与者学习，降低社区参与门槛
 - 尝试Gemini 2.5 Pro/DeepSeek v3 0324/DeepSeek R1/QwQ 32B等，**效果不佳**
 - 可能RAG参数设置待优化
- LLM+RAG辅助补丁审阅：同样效果不达预期
- 辅助生成社区工作摘要：相对较好，已在本次利用
 - <https://wszqkzqk.github.io/2025/03/30/config-and-use-aichat/>
- 处理用户反馈：多模态模型有助解析刷屏的信息（）

安同 OS

- Linux 6.14.1 内核正式推送

- 处理器核心数限制提升至 256（理论上：3C6000/Q x 2）
- 修复攀升等 3A6000 笔记本上触摸板指针极慢的问题
- 修复 AMD Radeon RX 9000 系列显卡无法驱动的问题
- 修复 3C6000/D 双路机型无法启动的问题
- 引入准上游 Intel xe 补丁集，支持 DG1/DG2/Battlemage

- Mesa 25.0.3

- 回合 AMD Radeon RX 9000 系列显卡在 S3 唤醒后可能死机的问题修复

- Chromium 135 开放测试，开启 VA-API 视频硬解码功能

- oma topics --opt-in chromium-135.0.7049.84

- Go 1.24 包含 LSX/LASX 优化支持，助力后续 ollama 适配

- LATX 1.6.0 已推送稳定源

- x86 运行时更新仍有问题（Wine >= 9 无法启动，详见：[lat-opensource/lat#31](https://lat-opensource.com/lat#31)）

- Ren'Py 视觉小说游戏引擎已进入软件源，可原生运行部分此类游戏

本页预定讲者 Kexy Biscuit



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

- 修复 Docker 在近期更新 runc 组件后无法启动的问题
- 依赖 libseccomp-golang 未定义“原生架构”，[上游已修复](#)但未发版
- 待办项目
 - 发布 4 月系统发行更新（4 月 15 日前）
 - 引入 Box64 及 4K/16K 双内核测试
 - 引入支持 CPU/Vulkan 后端的 ollama
 - 修缮 ROCm 的龙架构支持
 - Linux 6.15-rc 内核测试
 - libLoL 软件支持情况排查、更新软件安装和兼容性指南等
- 安全更新与用户公告
 - 由于先前出现更新问题导致镜像发布延迟，社区软件源公钥迭代推迟至 4 月 15 日，请确保在此日期前至少完整更新一次系统
 - 建议关注公众号“安同开源”或社区主页 (aosc.io) 新闻

本页预定讲者 Kexy Biscuit



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

```
.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
```

```
# 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:
```

快速报告

社区事务

本页预定讲者

龙架构 LoongArch
Biweekly
双周会

龙架构漂流板卡计划重构

- 龙架构漂流板卡计划现已重构，并在 GitHub 同步发布

- <https://github.com/loongson-community/1024>

- 主要变化：

- 信息同步在GitHub中（原漂流板计划一直运维在自建 Gitea 中）
 - 新增更详细的库存和借出信息
 - 新增龙架构虚拟机物品和一些用于奖励的纪念品

备注：目前 3A6000 整机和主板漂流板卡数量较少，2K0300 芯片库存较多，欢迎大家借出使用；如有 Web 类或非 UI 类应用移植业务可申请虚拟机使用

龙架构双周会线下交流会

- 时间：4月26日或4月27日
- 地点：湖北（具体地点待定）
- 线下参与会发放龙架构纪念品
 - 抽奖：2K0300系列板卡、龙芯定制水杯、3A6000 钥匙扣，**必中**



扫描二维码填写报名问卷

龙架构 LoongArch
Biweekly
双周会

本页预定讲者 寻觅

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

LoongArch兼容性数据库

```
file:
```

```
# e_ident
```

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

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

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

- 在徐老师和社区工作人员的帮助下完成域名备案，在此表示感谢

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

- 现可使用 <https://loong123.cn> 访问本数据库

```
.rept 0
```

- 原 loongarch.liaronce.com 现已 301 跳转到新域名

- ~~部分地区可能会失效，原因不明~~

```
.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 LiarOnce

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
```

```
# a reloc entry has to be big enough for even 64KiB-page kernels
```

```
.set base_addr = 0
```

```
.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:
```

专题报告

本页预定讲者

龙架构 LoongArch
Biweekly
双周会

洛书（Losu）— 超轻量级多用途编程语言

- 洛书简介：特性、语法与技术路线
- 移植与测试（3A6000）
- 快速体验（Losu4Wasm）
- 总结

```
1  def sayhello(msg):
2      print(msg)
3
4  var s = "Hello,World!"
5  sayhello(s)
```

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

```
filestart:
```

```
# e_ident
```

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

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

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

```
.byte 0x00 # EV_CURRENT
```

```
.byte 0x00 # EI_MACHINE
```

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

```
.rept 7
```

```
.byte 0
```

```
.endr
```

- 洛书简介：特性、语法与技术路线

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

- 移植与测试 (3A6000)

```
.short 2 # ET_EXEC
```

- 快速体验 (Losu4Wasm)

```
.short 0x10000000 # e_version = 1
```

- 应用场景

```
.dword base + e_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

```

1  # async function
2  async task(id):
3      for i in 1,5:
4          print("Task:", id, " is running, x",i)
5          yield
6
7  for i in 1,5:
8      task(i)
9
10 await()
11
12 print("main thread ok")
13

```

- 超轻量级的多用途语言，支持定制 DSL
- 类 **Python** 风格代码、强调代码可读性与实用性
- 多范式编程支持：OOP、FP、元编程
- 并发支持：原生协程（裸机可用）
- 性能良好，资源占用低，可移植性高

```

1  # Json file
2  let json = {
3      "name": "John",
4      "age": 30,
5      "city": "New York"
6      "hobbies": ["reading", "cooking", "traveling"]
7  }

```

```

1  var per = 2
2  var t = 0
3  def draw():
4      if time() - t < 50:
5          return
6      t = time()
7      per = per - ( 1 / 64 )
8      if per<0:
9          per = 2
10     rect(0, 0, 800, 600, 'white')
11     circle(400, 300, 100, 0, 2, "green" ,"red", 5)
12     circle(400, 300, 100, 0, per, "yellow" ,"blue", 5)
13     text( (per / 2 * 100) & " \%", 0, 0, 32, "black")
14
15     def click(x,y):
16         print("click ",x,y)
17
18     def keypress(key):
19         if chr(key)=='M':
20             sound('man.mp3')
21

```

```
ctci@ctci-pc: ~/losu — ssh ctci@192.168.42.253 — 89x30
[ctci@ctci-pc:~/losu$ time python3 test.py
real    0m22.744s
user    0m22.710s
sys     0m0.004s
[ctci@ctci-pc:~/losu$ time ./losu test.els
real    0m12.449s
user    0m12.448s
sys     0m0.000s
[ctci@ctci-pc:~/losu$ cat test.py
def fib(i):
    if i<3:
        return 1
    return fib(i-1)+fib(i-2)
fib(40)
[ctci@ctci-pc:~/losu$ cat test.els
def fib(i):
    if i<3:
        return 1
    return fib(i-1)+fib(i-2)
fib(40)
ctci@ctci-pc:~/losu$ █
```

```
1  # Naive Fibonacci
2
3  def fib(i):
4      if i<3:
5          return 1
6      return fib(i-1) + fib(i-2)
7
8  fib(40)
```

	PHP (10.85s)
	Ruby (12.17s)
	+1 Python (29.00s)
	-1 R (68.96s)

- 移植(any os, any where):
 - C Compiler, std >= c99
 - libc: glibc, musl, newlib, MicroLIB
 - RAM >= 4/8 KB, Codespace >= 32/64 KB
- 测试:
 - UOS Desktop 20, 3A6000(loongarch64)
 - Naive Fibonacci 40 (benjdd.com)

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

扩展 losu4wasm x

losu4wasm

Neuq-losu-2025

main.els x

main.els

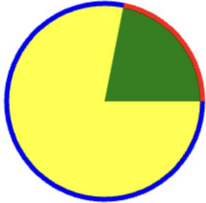
```
1 var per = 2
2 var t = 0
3 def draw():
4     if time() - t < 50:
5         return
6     t = time()
7     per = per - ( 1 / 64 )
8     if per < 0:
9         per = 2
10    rect(0, 0, 600, 600, 'white')
11    circle(400, 300, 100, 0, 2, "green", "red", 5)
12    circle(400, 300, 100, 0, per, "yellow", "blue", 5)
13    text( (per / 2 * 100) & " %", 0, 0, 32, "black")
14
15 def click(x,y):
16     print("click ",x,y)
17
18 print("Hello World!")
19
```

运行 洛书 (Losu)

main.els x

60 FPS 16.050 KB

78.125 %



1|Hello World!

行 3, 列 12 空格: 4 UTF-8 LF () losu



龙架构 LoongArch
Biweekly
双周会

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

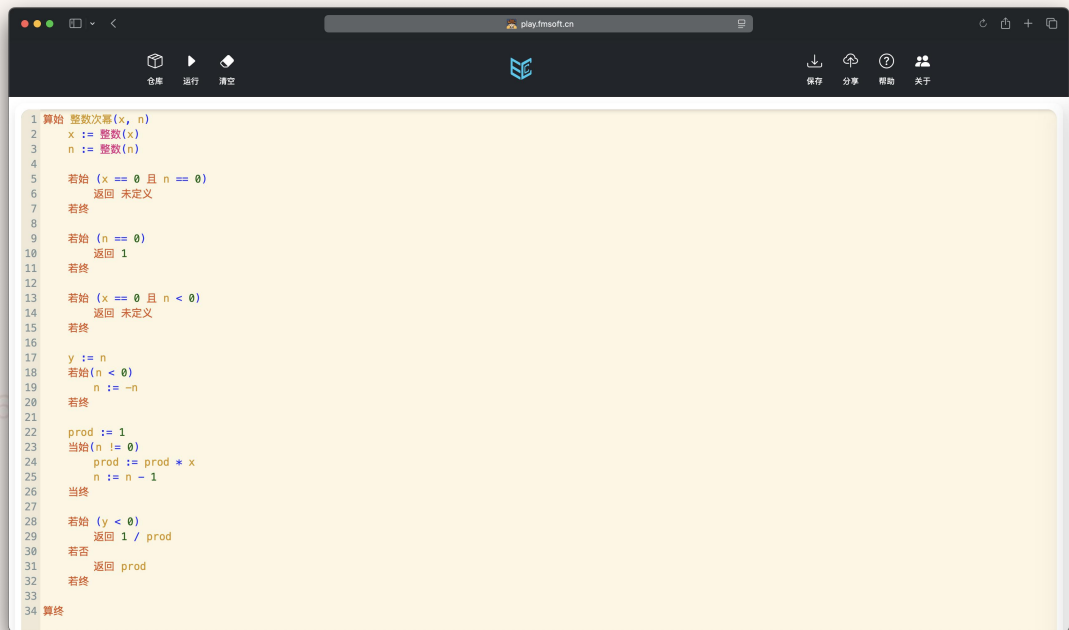
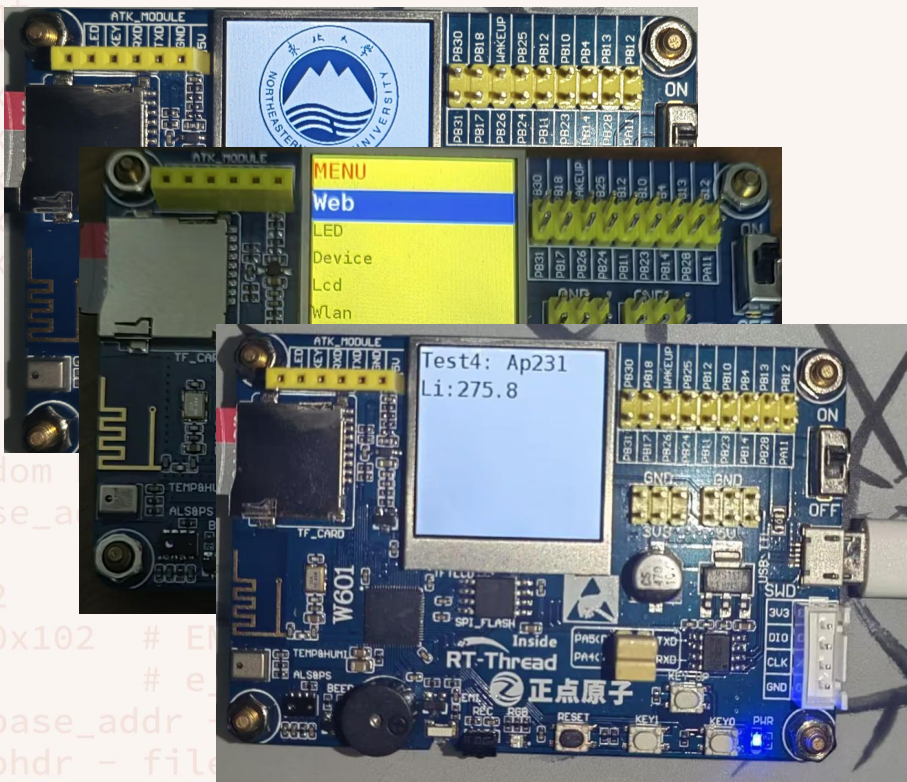
```
filestart:
```

```
# e_ident
.ascii "
.byte 0x
.byte 0x
.byte 0x
.byte 0x
.rept 7
.byte 0
.endr
```

```
# a random
.set base_a
```

```
.short 2
.short 0x102 # E
.word 1 # e
.dword base_addr
.dword phdr - file
.dword 0
.word 0x41
.short ehsize
.short phentsize
.short 1
.short 0
.short 0
.short 0
.set ehsize, . - filestart
```

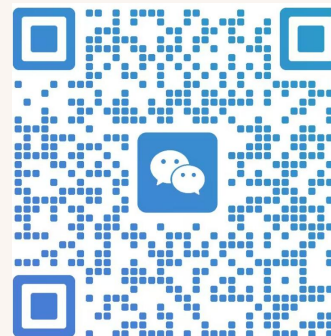
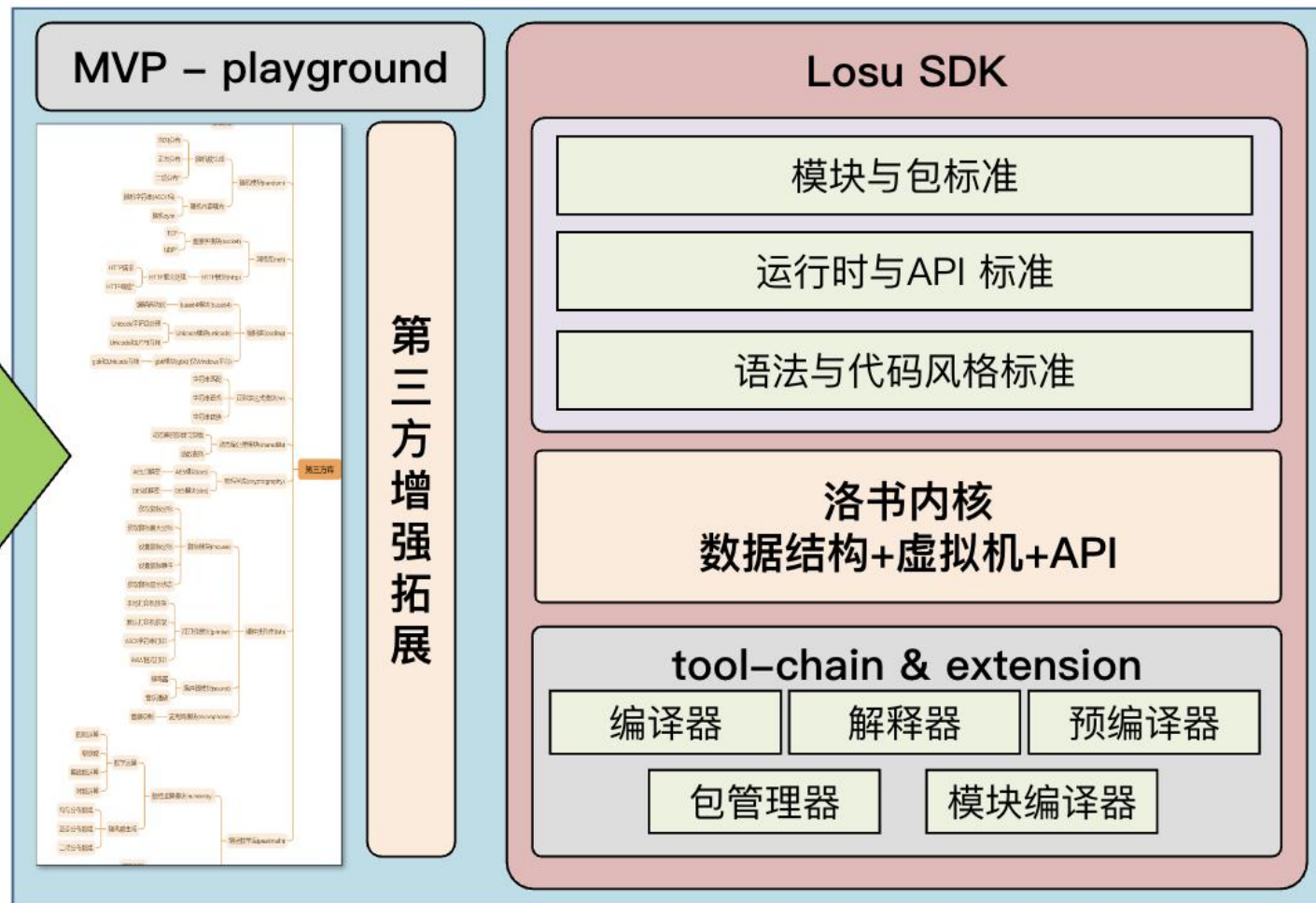
```
phdr:
```



脚本开发，可以用于各类IoT场景
定制化 DSL，适用于专用领域

龙架构 LoongArch
Biweekly
双周会

MMP: 最小可市场化产品



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
```

```
.repl
```

```
.byte
```

```
.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
```

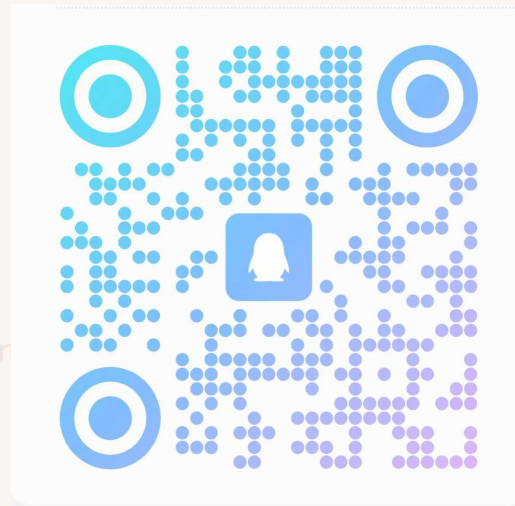
```
.short 0 # e_shstrndx
```

```
.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