

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
```

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

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

```
.short 0x0000 # EM_EBC
```

```
.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
```

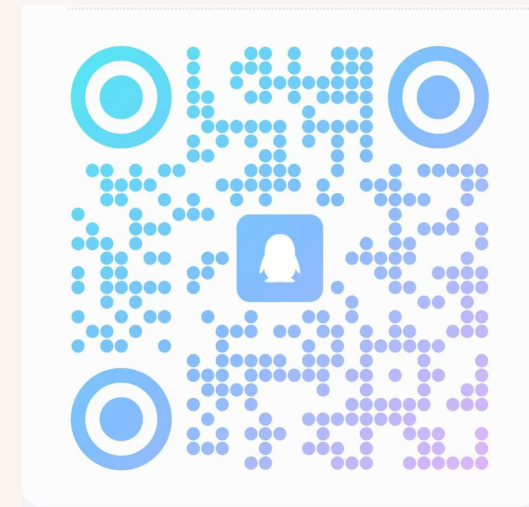
```
.short 0 # e_shentsize
```

```
.short 0 # e_shnum
```

```
.short 0 # e_shstrndx
```

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

```
phdr:
```



龙架构双周会

2025 年 2 月 16 日 · 第 5 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, 0x270000
```

```
.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

上游工作动向

龙架构 LoongArch
Biweekly
双周会

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

Firefox: Skia

```
file_start:
# e_ident
.ascii "\177ELF"
```

- 构建 Firefox 过程中，Skia 构建失败

- 向量数据类型的隐式转换 (-flax-vector-conversions) 问题

- [先前的补丁](#) (CL 908137) 仍然在等待审查

- 几天前 xtex 由于不知情上述变更，[撞车了](#)，已经协调

- xtex 保留[最后一个变更](#)，xen0n 将 xtex 其余的修改更新入 CL 908137

- 截至 20250215，（终于）有了进展，但仍未合并

```
.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:
```

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

Firefox: libyuv

- 构建 Firefox 过程中，libyuv 链接时缺 LSX/LASX 相关符号
 - 龙芯 12 月[提交的](#)改法过于简单粗暴：直接禁用了 SIMD，会有性能问题
 - [声称的](#)理由是 Skia 版本老、libpng 缺文件，以及 ABI1.0 默认不开 LSX
 - 都是可解决或已解决的，而原作者选择了先解决眼前问题，把重构留给了后人
 - libyuv 上游的 GYP 项目定义缺文件，还缺 CFLAGS 控制
 - [上游补丁甲](#)、[乙](#)
 - libyuv 的 LSX & LASX 文件还存在重复定义，故不能被置于同一编译单元
 - 其他架构的优化代码都各自只位于一个文件，故不存在该问题
 - Firefox 等项目采用统一源码构建（unified sources build），受影响
 - xen0n [临时修复了](#)
 - 需要在上述 libyuv 变更主线化后将其 vendor 入 Firefox 仓库
 - 再将 LSX/LASX 开关信息接入 libyuv 的构建定义
- 截至 20250215，libyuv 上游有进展，但仍未合并

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

Firefox: libpng & libvpx & WebRTC

- 构建 Firefox 过程中，libpng 链接时[缺 LSX 相关符号](#)
 - 先前[202412 临时方案](#)中的修复不完整
 - xen0n [补足](#)先前遗漏的 loongarch 子目录，将其[包含进](#) Firefox 构建系统
- 为 libvpx [启用](#) LSX 优化（来自 2022 年 7 月份的请求）
 - xen0n [更新了](#)配置，[启用了](#)优化
- 在 LoongArch 上[默认启用](#) WebRTC
 - 最早可[追溯至](#) AOSC 贡献者[刘子兴](#)的补丁
 - libwebrtc 使用 Chromium GN 构建系统，需要转一道集成进 Firefox
 - 需要[告诉](#) Firefox 的 GN 处理器：存在 linux loong64 这个组合
 - 然后为 libwebrtc [更新](#)自动生成的 moz.build 文件
 - 然后[启用](#) WebRTC 仅需一行代码
- 截至 20250215，均在等待代码审查或合并

```
# a random value to avoid collisions with even 64KiB-page kernels
```

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

```
.short 0x102, # FM_LOONGARCH
```

```
.word 1, # FM_LOONGARCH
```

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

```
.dword 0, # e_shoff
```

```
.word 0x41, # e_shsize
```

```
.short ehsize, # e_shsize
```

```
.short phents, # e_phents
```

```
.short 1, # e_phsize
```

```
.short 0, # e_phsize
```

```
.short 0, # e_phsize
```

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

```
phdr:
```

Ollama 适配

- DeepSeek 热潮持续；目前一般都使用 [ollama](#) 项目在本地部署模型
- 使用未修改的上游源码直接构建，会失败在 ggml C 代码
 - 需要添加 cgo directive 启用 LASX
 - 直觉是向 CFLAGS 添加 -mlasx，[但 Go 不让](#): @ideal 同学报告
 - xen0n 使 Go [放行了](#) LoongArch 特殊 CFLAGS
 - Workaround: 暂时可以以 -march=la464 起到 -mlasx 的作用，但不优雅
 - 后续计划：确定上述 Go 变更是否能进入 Go 1.24.x，跟进适配 ollama
 - 目前 Go 1.24.0 几近发布，不合适塞进去。需要看能否 backport 到 Go 1.24.1
- @ideal 同学也在 ggml 上游[修复了](#) CMake 的架构检测逻辑

GCC 优化

• 针对点乘和类似模式的向量操作优化（预计开会时已合并）

```
long
dotprod (int *x, int *y)
{
    long ans = 0;
    for (int i = 0; i < 4; i++)
        ans += 11 * x[i] * y[i];
    return ans;
}
```

```
vld    $vr1,$r4,0
vld    $vr0,$r5,0
vslti.w $vr2,$vr1,0
vslti.w $vr3,$vr0,0
vilvh.w $vr4,$vr2,$vr1
vilvh.w $vr5,$vr3,$vr0
vilvl.w $vr1,$vr2,$vr1
vilvl.w $vr0,$vr3,$vr0
vmul.d  $vr0,$vr0,$vr1
vmadd.d $vr0,$vr5,$vr4
vhaddw.q.d $vr0,$vr0,$vr0
vpickve2gr.du $r4,$vr0,0
jr      $r1
```

```
vld    $vr1,$r4,0
vld    $vr2,$r5,0
vmulwev.d.w $vr0,$vr2,$vr1
vmaddwod.d.w $vr0,$vr2,$vr1
vhaddw.q.d $vr0,$vr0,$vr0
vpickve2gr.du $r4,$vr0,0
jr      $r1
```

GCC 优化 (续)

- 16字节原子操作的代码生成 (自测中, 预计开会时会发出)

```
.section ".blob", "aw", @progbits
file_start:
# e_ident
.ascii "\177ELF"
.byte 0x7f # ELF_MAGIC
.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
.set base_addr, 0x200000
```

```
.short 2 # ET_EXEC
.short 0x102 # EM_LOONGARCH
.word 1 # e_version = 1
.dword base_addr + entry - filestart
.dword phdr - filestart # e_phoff
.dword 0 # e_shoff
.word 0x41 # objabi v1,
.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:
```

```
1:
ll.d $r16,$r18,0
bne $r16,$r12,2f
dbar 0x700
ld.d $r17,$r18,8
bne $r17,$r13,2f
move $r19,$r14
sc.q $r19,$r15,$r18,0
beqz $r19,1b
b 3f
2:
dbar 0b10100
3:
```

GCC 修复

- `#pragma GCC target "..."`
 - 据说是因为把架构相关优化代码拆到多个文件里很麻烦
 - 龙芯的工程师之前已经实现
 - 但没有重设 `__loongarch_asx` 之类的特性判别宏，所以即使用了 `#pragma GCC target ("lasx")` 也不能用 `__lasx_xvsrarni_h_w` 等函数
 - 现已由龙芯工程师修复
- 杂项
 - GCC 15 新出现的测试失败统计和修复
 - `vsr{a,l}ri` 去 UNSPEC 化
 - “夸大” `ldx/stx` 的成本
 - ...

其他上游进展

- OpenSSL 误读 -march=la64v1.0

- 用于生成汇编代码的 Perl 脚本中错误将所有带有 . 的参数均作为输出文件传参，导致 -march=la64v1.0 被错误当作输出文件

- 已由 xry111 [修复](#)并合入 3.2/3.3/3.4 分支

- LLVM LLD R_LARCH_CALL36 松弛 (relaxation) 优化

- 如跳转目标在 B/BL 范围 ($\pm 128\text{MiB}$) 内，可做如下变换：

- pcaddu18i X, %call36(sym); jirl zero, X, 0
→ b sym # R_LARCH_B26

- pcaddu18i X, %call36(sym); jirl ra, X, 0
→ bl sym # R_LARCH_B26

- xen0n 24 年 4 月就弄好了，但不会写测试，咕到现在，[刚弄完](#)

其他上游进展 (续)

- Linux 内核: USB 唤醒支持 (讨论停滞中)

- 针对先前用户报告的无法使用 USB 键盘唤醒龙架构设备的问题，陈华才已提交上游补丁供审阅 (已在多平台测试可用)
- 初步测试结果: 非 EHCI/xHCI 设备无法用作唤醒
- 经过进一步测试 (尤其是对比带有 UHCI 及 OHCI 平台的设备)，发现 UHCI 的确无法用于唤醒 (ThinkPad X200s)，但 OHCI 无此问题 (AM3/RS780/SB710 主板)；很可能属于 7A 硬件问题
- UHCI 在内核源码中已有注释，的确不支持唤醒
- 详见 [drivers/usb/host/uhci-hcd.c:328](#)

```
.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

龙架构 LoongArch
Biweekly
双周会

Arch Linux for Loong64

- 单点故障：主要维护者个人设备故障影响开发
- devtools-loong64 (by [wszqkzqk](#))
 - 更改CFLAGS/CXXFLAGS,默认Code Model为medium
 - 防止某些静态库/对象没有手动覆盖导致大软件包链接失败
 - 合理性
 - Rust已默认Code Model为medium
 - 确认至少bfd、mold已支持relax，不会影响生成文件的性能
 - 减少补丁维护量
- <https://aur.archlinux.org/packages/devtools-loong64>



龙架构 LoongArch
Biweekly
双周会

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

Arch Linux for Loong64 (续)

```
fi  
# e_ident  
.ascii "\177ELF"
```

- Linux内核配置文件清理 (by [wszqkzqk](#))

- 历史 (<https://wszqkzqk.github.io/2024/09/12/loongarchlinux-kernel/>) :

- 初期，维护者不熟悉kernel config，但是必须要维护kernel
- kernel config要支持loong64，也要能够同步Arch Linux上游更改
- 根据Arch Linux上游的config，在loong64下生成一份defconfig
- 借用其他支持loong64发行版的config，追加Arch Linux上游config，生成defconfig
- 生成两defconfig间diff，提取出匹配 `/^+CONFIG_.*=[my]$/` 的行
- 去除开头"+"并放入文件，每次构建时追加在Arch Linux上游的config后
- 手动清理

- 包含大量冗余的额外config (≈1200行)

- 近期清理了大部分冗余config，且不影响正常使用 (剩余184行)

- pkg.tar.zst体积减小 ≈ 10 MB

- 仍待继续清理

Arch Linux for Loong64 (续)

- 大型软件包变动
- Electron: 仓库已包含electron 30/31/32/33/34 (by wszqkzqk, NakanoMiku)
 - electron 32/33改为Local Patch维护, 不再采用类似darkyzhou的模式
 - 由“Patch electron的Patch流程”改为在整合子模块源码后直接应用补丁
 - 减少冲突/更符合Arch Linux流程 (其他electron版本一直是Local Patch形式)
 - electron 32小版本升级(32.3.0)新问题 (自身GUI无法复现, 需要Code - OSS)
 - Code - OSS: reason: 'crashed', code: '139'
 - sandbox问题, 但系统调用看起来没有问题
 - 修复: 放行了sched_getaffinity (可能有更好的办法)
 - <https://github.com/lcpu-club/loongarch-packages/pull/417>
 - electron 33(随机出现)/34 rollup问题
 - 注释掉terser()的方法失效, 参考archriscv的方法更改npm依赖, 已解决
 - <https://github.com/lcpu-club/loongarch-packages/pull/404>

Arch Linux for Loong64 (续)

- 大型软件包变动
- Firefox 135/136更新：移除全部树外补丁 (by wszqkzqk)
 - 上游已有修复（未必合理），本地仅保留对构建配置的少量修改
- Chromium 133更新 (by wszqkzqk)
 - 类似electron32的sandbox问题
 - [0208/180845.635771:ERROR:exception_snapshot_linux.cc(427)] invalid magic number 0x41535801
 - 同样的修复：放行了 sched_getaffinity
 - <https://github.com/lcpu-club/loongarch-packages/pull/425>
- Ollama (by wszqkzqk)
 - 开启了LASX
 - 手动指定ARCH=loong64（否则会使用 \$(uname -m)）
 - 启用CPU Target

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

Arch Linux for Loong64 (续)

```
file {
```

```
# e_ident
```

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

```
.byte 0x02
```

```
.byte 0x01
```

```
.byte 0x01
```

```
.byte 0x00
```

```
.byte 0x00
```

```
.rept 7
```

```
.byte 0
```

```
.endr
```

- 上游化修复

- deepin-pw-check (by Pluto Yang)

- 追踪同步上游库的API修改，修复构建（上游尚未响应）

- <https://github.com/linuxdeepin/deepin-pw-check/pull/44>

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

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

```
.short 2
```

```
.short 0x102
```

```
.word 1
```

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

```
.dword phdr - filestart
```

```
.dword 0
```

```
.word 0x41
```

```
.short ehsize
```

```
.short phentsize
```

```
.short 1
```

```
.short 0
```

```
.short 0
```

```
.short 0
```

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

```
phdr:
```

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

Arch Linux for Loong64 (续)

```
fi
```

```
# e_ident
.ascii "\177ELF"
.byte 0x02
```

```
.byte 0x01
```

```
.byte 0x01
```

```
.byte 0x00
```

```
.byte 0x00
```

```
.rept 7
```

```
.byte 0
```

```
.endr
```

```
# a random base
```

```
.set base, 0x200000
```

```
.short 2
```

```
.short 0x102
```

```
.word 1
```

```
.dword 1
```

```
.dword phdr - filestart
```

```
.dword 0
```

```
.word 0x41
```

```
.short ehsize
```

```
.short phentsize
```

```
.short 1
```

```
.short 0
```

```
.short 0
```

```
.short 0
```

```
.short 0
```

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

```
phdr:
```

- 其他变动

- 日常更新与维护 (by Pluto Yang, wszqkzqk)

- 较多，包括工具链、DE、基础库等

- 新增linux-lts软件包 (6.12) (by wszqkzqk)

- <https://github.com/lcpu-club/loongarch-packages/pull/407>

- 禁用archinstall的refactor镜像搜索以支持loong64平台 (by wszqkzqk)

- <https://github.com/lcpu-club/loongarch-packages/pull/421>

- 补丁集GitHub仓库新增基于GitHub Workflow的简单PR检查 (by wszqkzqk)

- <https://github.com/lcpu-club/loongarch-packages/pull/431>

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

Arch Linux for Loong64 (续)

```
file:
# e_ident
```

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

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

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

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

```
.byte 0x00 # EABI_VERSION
```

```
.byte 0x00 # ET_NONE
```

```
.rept 7
.byte 0
.endr
```

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

```
.set base_addr, 0xffffffffffffffff
```

```
.set base_addr, 0xffffffffffffffff
```

```
.set base_addr, 0xffffffffffffffff
```

```
.short 2 # e_type = EXEC
.short 0x102 # e_machine = M68K
.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
```

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

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

```
phdr:
```

• 问题交流

• Chromium PGO

- 开启PGO后即使覆盖Code Model为medium仍然relocation R_LARCH_B26 overflow

- 检查生成的LLVM字节码.o文件，确实覆盖了Code Model为medium

- 尝试传递 --no-relax，同样无效

• 硬件资源条件/CI设施

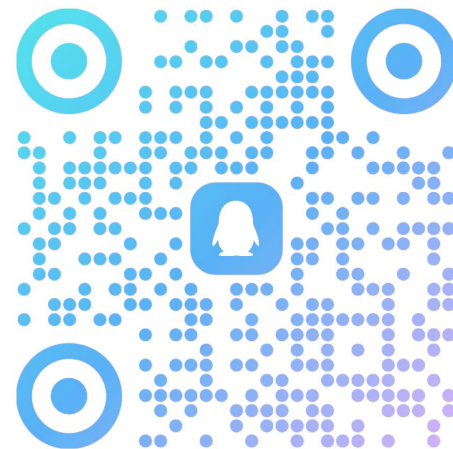
安同 OS

- 春节期间发布了一月发行更新
 - 包含 3C6000 支持及 LoongGPU 驱动
- Linux 6.14 内核可供尝鲜
 - 可通过 `oma install linux+kernel+rc` 直接安装
 - 属于超前预览且存在可用性风险，请量力而为
- Mesa 25 RC3 已开放测试
 - Intel 独立显卡用户反馈性能已非常接近 Windows!
- 用户反馈及关切
 - 正在修缮佳能打印机的驱动支持及部分行业软件的兼容性
 - x86 运行时及 Box64 ...
 - 私密吗喽，马上马上!



AOSC 社区频道

群号: 875059676



扫一扫二维码，加入群聊

龙架构 LoongArch
Biweekly
双周会

安同 OS

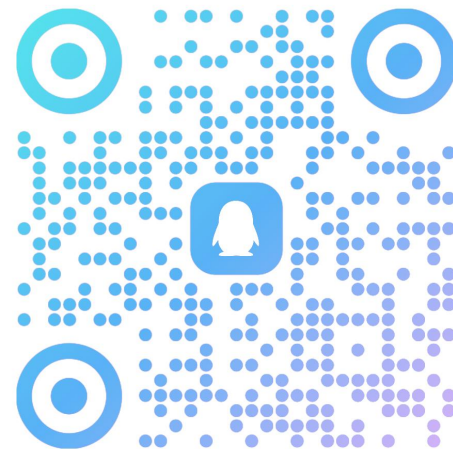
• 用户反馈及关切（续）

- 计划探索 4K/16K 分页双内核发行的可行性
 - 来自 Box64 维护者的请求
- llama.cpp 已在稳定源，ollama 即将打包
- Chatbox、阿里云 OSS Browser 等软件的适配已由贡献者 darkyzhou 接手
 - 希望近期可以完成移植，方便更多有需要的用户
- Wine 10 更新已在日程中，将增强 32 位侧的功能
 - NTSYNC 集成亦在计划中
- libvirt 10.5.0 + QEMU 9.2 在 3C6000 上启动虚拟机时卡死
 - 正在调查中



AOSC 社区频道

群号：875059676



扫一扫二维码，加入群聊

龙架构 LoongArch
Biweekly
双周会

安同 OS

- 用户反馈及关切 (续)

- LoongGPU 在 6.12/6.13/6.14 内核上无法启动非图形会话 (multi-user.target)

- 暂时定位到是 AOSC 补丁集导致的问题
- 有待进一步二分

- Ren'Py 引擎支持

- 已在日程上

- 其他计划?

- 目前观察到 deepin/kernel 新到一批补丁 ([#599](#))

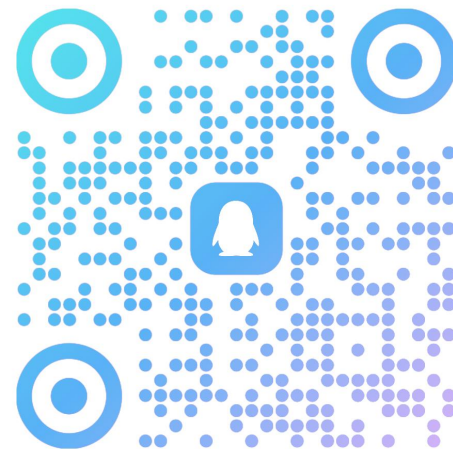
- 其中包含 I2C 输入设备的支持

- 预期可以修复联想 N60d G1d 等笔记本的触摸板
- 已排入下一个稳定内核更新



AOSC 社区频道

群号: 875059676



扫一扫二维码, 加入群聊

龙架构 LoongArch
Biweekly
双周会

诚迈COS

- 2025年1月24日发布新版本 名称由"诚鸿OS操作系统"改为"诚迈COS操作系统" 英文"ArcherMind COS 社区版 V25" (基于OpenHarmony5.0)
- 系统logo更改
- 开机动画修改
- 文管支持U盘挂载访问, 文件复制黏贴,
- U盘当前挂载路径为 主目录/External/ 目录下
- 新增画图软件
- 支持物理键盘启用中文输入法
- 修复系统其他bug



3A6000 优利龙社区PC 适配诚迈COS

诚迈COS

• 镜像下载网站www.openloongson.org

• 优利龙多系统社区PC已经适配该系统

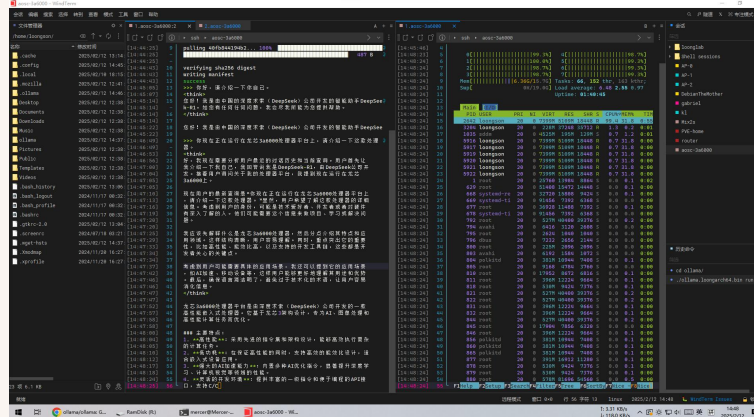
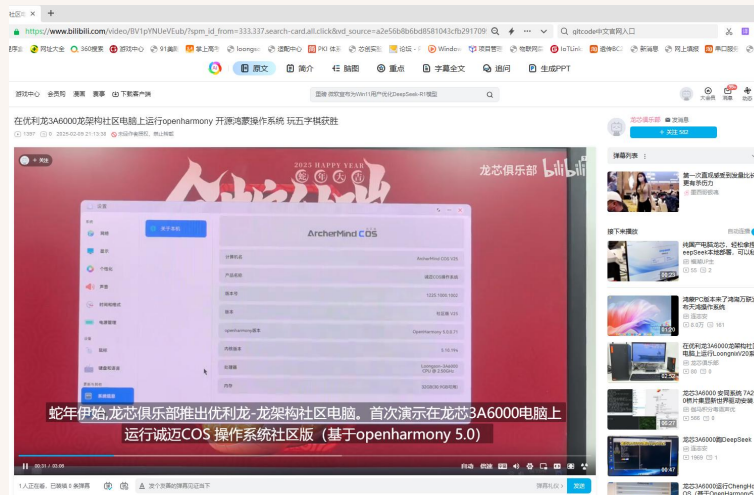
标准版预装诚迈COS（基于OpenHarmony5.0）、
Loongnix、安同等多个社区操作系统。

B站演示视频：右上图

• DeepSeek移植

龙芯3A6000上安同、统信等系统上已经实现
DeepSeek的适配，右下图。准备DeepSeek向诚迈
COS移植的社区项目

联系合作:龙芯俱乐部 微信



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

龙架构 LoongArch
Biweekly
双周会

龙架构悬赏任务

- [龙架构悬赏任务\(LoongTask\)](#), 是龙芯爱好者社区发布的悬赏任务, 所有开发者均可以个人身份或团队参与
- 相比于已自由软件为主的[英雄帖](#), 悬赏任务中会出现一些商业性质的项目
- 部分商业项目只会要求适配ABI1.0(旧世界)的需求, 但项目运维人员往往会在验收标准中添加ABI2.0(新世界)的适配要求以及在社区编写相关教程
- 悬赏任务不可重复接取避免重复劳动, 一旦有个人/团队接取任务后即会被锁定, 但需要按照任务说明定期同步进度, 否则就可能会被认定为放弃任务
- 部分偏向社区的长期悬赏任务可能会要求在双周会时同步任务进度

悬赏任务

国家中小学智慧教育平台客户端

下载链接:

https://download.ykt.cbern.com.cn/pc_ver/%E6%99%BA%E6%85%A7%E4%B8%AD%E5%B0%8F%E5%AD%A6%20Setup%201.3.6.exe.

任务要求: 上述链接中的 exe 文件由 Electron 打包而成。现需将该文件解包适配龙架构后重新打包发布, 使龙芯平台能够原生使用该软件。

奖励: 3A6000 龙芯笔记本或者 3A6000E 主机 (二选一)

具体悬赏任务细则见: <https://gitea.whlug.cn/LoongFly/LoongTask/issues/1>

社区基础设施进展说明

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

```
file_start
```

```
# e_ident
```

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

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

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

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

```
.byte 0x00
```

```
.rept 7
```

```
.byte 0
```

```
.endr
```

- 由于基础设施用的是3C6000
- 目前集群使用的是PVE+Ceph的方式搭建, 系统搭建已经完成
- 仅支持LA464, 导致虽然使用的是3C6000主机, 但在系统里识别的是3A5000

```
# a random
```

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

- 部分GUI安装界面会出现卡死的现象, 而且卡死的位置比较随机, 目前暂未定位到问题所在, 如果社区这边有比较了解PVE使用的希望能联系我协助去定位一下相关问题

```
.short 2 # EM_LOONGARCH
```

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

- 目前社区专线网络的需要准备的相关资料已提供给运营商审核, 正在等待运营商部署网络

```
.word 1 # objabi v1, soft-float
```

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

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

```
.dword 0 # e_shoff
```

```
.word 0x41 # e_ehsize
```

```
.short ehsize # e_phentsize
```

```
.short phentsize # e_phnum
```

```
.short 1 # e_shentsize
```

```
.short 0 # e_shnum
```

```
.short 0 # e_shstrndx
```

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

```
phdr:
```

抽奖环节

论坛抽奖以及双周会抽奖

```
.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:
```

龙架构 LoongArch
双周会 Biweekly

```
.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
```

专题报告

```
# 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:
```

龙架构安卓进展 (Android On LoongArch, AOL)

官方文档: <https://github.com/android-la64/docs>

- aosp12, 基于android-12.0.0_r1 (旧世界)
 - 完成最基础的移植
 - 工具链
 - Linux内核
 - Bionic
 - ART C++解释执行器
 - 采用amd520 gpu进行图形驱动
- aosp15, 基于android-15.0.0_r1 (新世界)
 - 上游多数包已支持LoongArch
 - 工具链/内核/strace etc.
- 系统配置位于device/loongson/loongsonboard中



工具链 && 图形

工具链

- aosp12
 - 基于LLVM15
- aosp15
 - 基于上游的LLVM19
 - 工具链、NDK通过所有测试
 - Bionic仍有少量BUG

图形

- aosp12
 - 7A2000适配：需打通安卓图形栈
 - 安卓桌面集成
 - 与Loongnix共用内核，在新tty中可进入安卓图形界面
- aosp15
 - Doing：打通图形栈，启动到桌面
 - Doing：cuttlefish模拟器

安卓运行时进展

工具链

- aosp12

- C++解释器
- mterp汇编解释器：74%通过率 (688/928)
- JIT支持: `art/test.py -r --target --no-prebuild --ndebug --no-image --64 --jit` 通过所有测试
- 目前仅支持基础整数/浮点指令
- Doing: AOT通过部分测试
- TODO: 完善 & 优化

- aosp15

- C++解释器 All tests passed. (https://github.com/android-la64/platform-art/tree/a15_larch_dev)
- Doing: nterp汇编解释器
- TODO: JIT/AOT
- 二进制翻译
- Doing: 借鉴berberis开发Arm->LoongArch上的动态库翻译器

VSCodium 插件生态适配

• 现状:

VSCodium 已原生支持龙架构，许多常见的编程语言的插件无需适配即可安装使用
但存在一些例外：rust-analyzer、csharp、codelldb 等

• 限制:

只要是自带二进制的插件都无法在龙架构 VSCodium 上安装

微软的和 Open VSX 插件市场的架构列表不包含龙架构，开发者无法上传专用二进制

• 解决:

短期: fork 一个不自带二进制的「通用」版（不限平台），如 rust-analyzer-no-server

长期: 联合 VSCodium 其它架构维护者向 Open VSX 倡议添加平台支持（要求用户量）

• 联络:

GitHub 组织: <https://github.com/loongcodium>

移植反馈/相关讨论: [loongbbs 帖子](#)、[GitHub Discussion](#)

```
.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
```

```
.end
```

问答环节

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

```
.set base_addr, 0x20000000
```

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:
```

龙架构 LoongArch
Biweekly
双周会

预收集提问：LoongGPU

1. 请问 LoongGPU 新世界驱动什么时候支持 Wayland？什么时候提升 OpenGL 版本？
2. LoongGPU 驱动有计划推进进 Debian 的 non-free 仓库吗？
3. GPU 驱动更新周期有计划吗？春节前出的好像只是技术预览版。
4. 请问 LG200 的 VPU 参数如何？会支持 VA-API 吗？
5. 龙芯已经在开发 2K3000 上的 GPU (LG200) 驱动了吗？

预收集提问：其他

1. 目前龙架构 dpkg 发行版对龙架构有 loongarch64 和 loong64 两种不同的称呼，对用户安装软件包造成了一定困扰，请问会不会有统合方案？
2. 当前的龙芯 3C6000 单路主板不能用 PCIe 4.0，是什么问题导致的？
3. LATX 目前开源进展如何，是否有确定具体开源时间？
4. 目前 LATX 项目尚未有官方的 logo 设计，先前已有社区小伙伴（特首）给 LATX 提交过设计稿，但后续结论似乎是希望通过公司寻找美工进行正式设计，这方面有推进计划吗？

```
.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
```

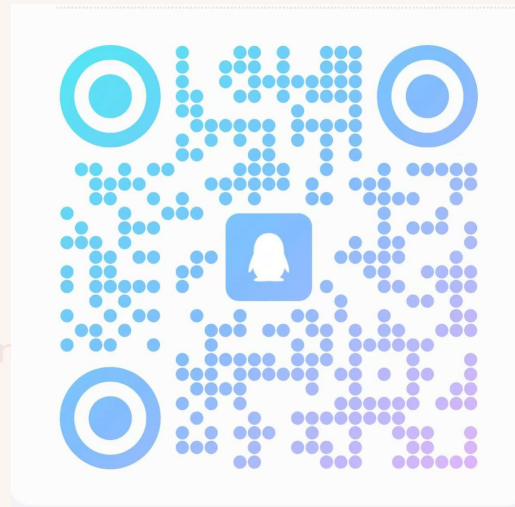
```
.short 0 # e_shstrndx
```

```
.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