

# 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

```
.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 0x02 # EM_LOONGARCH
```

```
.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
```

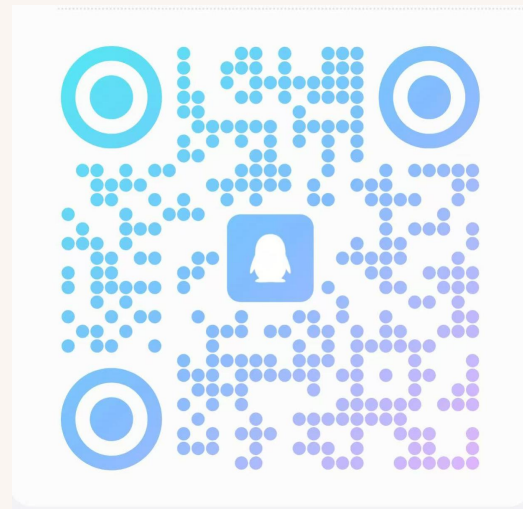
```
.short 0 # e_shstrndx
```

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

```
phdr:
```

# 龙架构双周会

2025年8月17日·第18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
```

# 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

# GCC

- chenglulu 修复了 -mtune=la664 导致性能劣化的问题 ([PR 120476](#))
  - 查明原因是在 movgr2cf 和 movcf2gr 的代价都较小时，编译器估算寄存器压力时会很野兽地将通用寄存器和浮点标记寄存器视为同一类寄存器，从而低估寄存器压力
  - 通过 TARGET\_COMPUTE\_PRESSURE\_CLASSES 指定估算寄存器压力时的寄存器分类即可解决此问题
  - 已经回合到所有支持 -mtune=la664 的 release 分支，建议使用 -mtune=la664 的发行版立刻回合该修复（或更新 GCC 到对应 release 分支的最新状态）并重新构建性能敏感的软件包

# GCC

- Yang Yujie 完成了 C23 `_BitInt` 的实现 ([PR 117599](#))
- ABI 文档尚未更新，但根据 Yang Yujie 在提交记录中的说法，在 64 位龙架构上 `_BitInt(N)` 的 ABI 如下规定：
  - N 不大于 64 时，`_BitInt(N)` 符号扩展/零扩展（根据该 `_BitInt` 是否有符号）到 `M = N.next_power_of_two()` 位，然后视为 M 位的基本整数类型处理
  - N 大于 64 时，`_BitInt(N)` 符号扩展/零扩展到 `(N + 63) & ~63` 位，然后视为 `_Alignas(16) struct {i64 x[(N + 63) / 64];}` 处理

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

# GCC

```
file_start  
# e_ident  
.ascii "\177ELF"  
.byte 0x04 # EI_CLASS  
.byte 0x02 # EI_DATA  
.byte 0x01 # EI_CURRENT  
.byte 0x00 # EI_OSABI  
.rept 7  
.byte 0  
.endr
```

- xry111 回合 [PR 120983](#) 修复到 releases/gcc-15 分支

- 触发该问题的 ext-dce 修复 ([PR 120242](#)) 已经回合到了该分支，导致 15.2 版本跑龙架构测试集时触发了该问题

- 实际代码遇到该问题的概率极低

```
# 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 xry111

龙架构 LoongArch  
Biweekly  
双周会

# Linux 内核 (loongarch 列表)

- Huacai Chen

- 将 CONFIG\_GCC\_PLUGIN\_STACKLEAK 更名为 CONFIG\_KSTACK\_ERASE ([第 1 版](#))
- 更正行为, 在保存 FPU 上下文前保存 LBT 上下文 ([第 1 版](#))

- Song Gao

- 使用 kvm\_get\_vcpu\_by\_id() 替换 kvm\_get\_vcpu() ([第 1 版](#))

- Haoran Jiang

- 修复 eBPF 程序中返回指针值 (return pointer value) 不正确的问题 ([第 1 版](#))

- Xianglai Li

- 修复虚拟处理器 (vCPU) 在未启用 (Offline) 的情况下占用极高的问题 ([第 1 版](#))

- Bibo Mao

- 为 EIOINTC 添加多中断引脚路由支持 ([第 4 版](#))
- 增强处理器间中断 (IPI, Inter-Processor Interrupt) 及 LBT 启用功能 ([第 3 版](#))
- 为 pch\_pic 新增多种 MMIO 访问大小支持 ([第 1 版](#))

# Linux 内核 (loongarch 列表)

- Kanglong Wang

- 通过优化重定位时 PLT/GOT 所需表项数量的计算提高模块载入速度 ([第 1 版](#))

- Ming Wang

- 将 COMMAND\_LINE\_SIZE 定义的大小提升为 4096 字节 ([第 1 版](#))

- Ruoyao Xi

- 更正 7A 桥片 PWM 的默认时钟频率：原定义 50kHz，应为 50MHz ([第 1 版](#))

- Tiezhu Yang

- 修复 objtool 在启用 LTO 时的警告 ([第 1 版](#))

- Binbin Zhou

- 2K0500/2K1000 NAND 控制器支持 ([第 2 版](#))

- 龙芯 2K0500 BMC 支持 ([第 9 版](#))

# Linux 内核 (loongarch 列表)

• Yao Zi

- 2K0300 SoC 时钟支持 ([第 3 版](#))
- 2K0300 SoC 引脚控制器支持 ([第 1 版](#))
- 2K0300 SoC 复位控制器支持 ([第 1 版](#))
- 2K0300 GPIO 控制器支持 ([第 1 版](#))

```
# 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
```

本页预定讲者 梓瑶

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
```

# 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

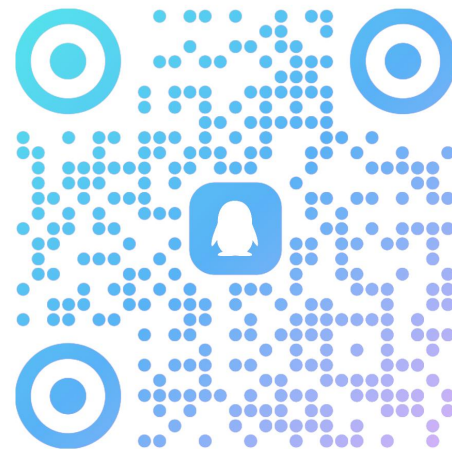
# 安同 OS (AOSC OS)

- Linux 6.16 内核更新可供测试
  - 开始提供 16KiB（默认）及 4KiB 内核分页双内核
    - 后者可便利 Box64 开发测试及 AMD (ROCm)/Intel (ICR) 用途
    - 已知 4KiB 分页内核下 VSCode 及 Chromium 稳定性存在问题
  - 另包含数个平台支持修复
    - 3B6000 及 3C6000 家族 KVM 稳定性修复（来自商业发行版内核）
    - 龙芯 3A5000/3A6000 笔记本 LoongGPU 背光支持修复
- 目前已更新到 6.16.1，欢迎各位测试：  
**oma topics --opt-in linux-kernel-6.16.0**
- 背光修复（须搭配内核更新使用）：  
**oma topics --opt-in loonggpu-kernel-dkms-1.0.1\_alpha\_lnd25.5-4**
- 2K0300（无 SIMD）系统移植已接入 BuildIt! 自动化维护框架，系统镜像即将上线社区门户
- Mesa 25.2 更新已推送，已知 Intel 独显 Vulkan 支持存在问题，待查
- Core 13 即将推送测试，将针对 -mtune=la664 性能劣化修复安排全系统重构



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

### • **BusyBox 1.37.0 安全更新**

- 修复一处 TOCTOU（检查到使用时间）竞态条件漏洞（严重性：高）

### • **Thunderbird 141.0**

- 修复 Mozilla 基金会第 2025-61 号安全公告中披露的 16 个安全漏洞（含 6 个高危漏洞）

### • **AIDE 0.19.2**

- 修复一处输出无害化处理不当漏洞，漏洞编号 CVE-2025-54389（严重性：中）
- 修复一处空指针解引用漏洞，漏洞编号 CVE-2025-54409（严重性：中）

### • **uv 0.8.6**

- 修复一处利用解析实现差异导致某些组件包安装器解压有害 ZIP 内容的漏洞，漏洞编号 CVE-2025-54368（严重性：中）

### • **7-Zip 25.01**

- 修复一处解压时可能导致任意文件写入并可进而触发代码执行的漏洞，漏洞编号 CVE-2025-55188（严重性：低）

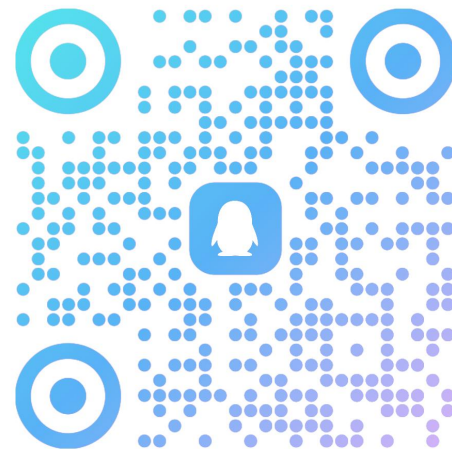
## • 建议关注公众号“安同开源”或社区主页 (aosc.io) 新闻

本页预定讲者 网管



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：笔记本背光问题调查

- 在使用 7A2000 内置的 LG110 显卡作为显示输出的龙架构笔记本上，使用安同 OS 时，背光亮度无法调节
- 根据《龙芯 CPU 统一系统架构规范》，使用 LG110 作为显示输出时，显示屏背光的亮度控制必须使用 7A2000 桥片上的 PWM3
- 容易发现，安同 OS 的内核未启用 7A 的 PWM 支持，这显然是导致问题的原因之一，但是启用以后背光还是不工作

# 安同 OS：笔记本背光问题调查

- 根据《龙芯 VBIOS 生成工具使用手册》，背光所用 PWM 的配置信息会被保存在 VBIOS 中，包括所用的 PWM 控制器编号，及其极性和工作频率
  - 我也不知道为什么在手册定死用 PWM3 的情况下还要在 VBIOS 存一个 PWM 控制器编号
- VBIOS 需要通过 GPU 设备的 PCI 地址空间读取，那么当然只有 GSGPU 驱动程序能读取上述配置信息
- 所以一定是 GSGPU 驱动程序去配置 PWM3 的输出翻转和工作频率，下面就需要读 GSGPU 代码...

# 安同 OS：笔记本背光问题调查

- 可以发现 GSGPU 试图使用 `lg_pwm_request(audev->ddev->dev, "pwm3", 3, "Loongson_bl")` 从 PWM 子系统获取 `pwm_device` 对象，这个辅助函数的定义如下：

```
static inline struct pwm_device *lg_pwm_request(struct device *dev,
const char *pwm_name,
int pwm, const char *lable)
```

```
#if defined(LG_PWM_REQUEST) e_entry
return pwm_request(pwm, lable);
```

```
#else
return pwm_get(dev, pwm_name);
```

```
#endif
```

# 安同 OS：笔记本背光问题调查

- 在 Loongnix、UOS 等内核中，pwm\_request 存在，所以相当于调用了 pwm\_request(3, "Loongson\_bl")
- 在 AOSC 内核中，pwm\_request 不存在，所以相当于调用了 pwm\_get(dev, "pwm3")

```
# 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:
```

# 安同 OS：笔记本背光问题调查

- 为什么要弃用 (v3.6) 和删除 (v6.4) pwm\_request?
  - 请问“3”这个数字代表什么？
  - 代表内核发现的第 4 路 PWM 输出.....
  - 那么如果内核在发现 7A 的 PWM 前先发现了另一个 PWM 芯片..... 是真的有可能爆炸（物理）的
- 为什么 pwm\_get 不工作？
  - 阅读内核文档和代码可以发现，pwm\_get 要成功，则要么 dev 对应的 ACPI 节点包含一指定对应 PWM 设备的属性，要么 "pwm3" 是通过设备树指定或 pwm\_add\_table 注册的 ID
  - GSGPU 根本没有 ACPI 节点，龙芯桌面设备也没有设备树，内核和 GSGPU 模块也没有任何代码注册过 "pwm3" 这个 ID

# 安同 OS：笔记本背光问题调查

- 那么我们可以在某个地方调用 `pwm_add_table` 注册这个 ID 吗？
- 很遗憾，这个函数设计上是给板级初始化代码使用的，因此并未导出供模块使用，我们无法在 GSGPU 模块中调用它
- 所以必须改动内核代码，在内核代码中搜索 `pwm_add_table`，发现一些 x86 设备在 ACPI 初始化代码中将 LPSS 的 PWM 注册为 `pwm_soc_backlight`，查提交记录发现这是为了允许 i915 驱动调用它以实现一些 x86 设备的背光控制，和我们的需求完全一致——(北美信创)——

# 安同 OS：笔记本背光问题调查

- 首先模仿 x86 的写法，在 ACPI 初始化代码中，如果扫描到 LOON0006:03 这个设备，就调用 pwm\_add\_table，将 gsgpu\_backlight 这个 ID 和该设备对应
- 然后修改 GSGPU 模块，如果发现 VBIOS 中使用的不是 PWM3 就直接报错，否则就用 pwm\_get 查找 gsgpu\_backlight 对应的设备
- ~~不沿用 "pwm3" 这个 ID 纯粹是它看上去太唐子~~
- 这样就好了.....吗？不，现在 GSGPU 模块又说找不到 GPIO 设备了

# 安同 OS：笔记本背光问题调查

- 初步调查发现找不到 GPIO 的原因是目前主线内核在对 GPIO 编号时没有处理 ACPI 表中的 gpio\_base 属性，所以 GPIO 的编号和《龙芯 CPU 统一系统架构规范》不一致，而 GSGPU 驱动是按后者写死了编号
- 不知如何解决该问题
  - 《规范》对 GPIO 的处理已有朝令夕改令人无所适从的先例：之前双周会报道触摸板问题时提到的被删掉的 gsi\_idx\_map
  - 上游讨论明确反对此种 GPIO 编号分配方式，建议使用动态分配
- 所以暂时把使用 GPIO 的代码都注释掉了，这样应该只会废掉省电功能（即应该完全关掉背光甚至 LCD 屏的时候只能把亮度拉到最小并显示黑屏），不会影响亮度调节

# 安同 OS：笔记本背光问题调查

- 然后亮度可以调了，但是不太正常，拉到比较低的亮度屏幕就会闪烁乃至黑屏
  - 之前试图通过人工操作 PWM 设备调整亮度时就已经发现此现象，还发现把 VBIOS 指定的 PWM 周期放大 1000 倍好像就没问题了
- 审查 pwm-loongson.c，发现其中有一个表示 PWM 设备时钟频率的常量
  - `#define LOONGSON_PWM_FREQ_DEFAULT 50000 /* Hz */`
- 如果你真的有使用 PWM 设备的经验就会发现这个频率低到反常识了……
- 翻 7A 手册发现时钟频率其实应该是 50 MHz，解释了之前发现的 1000 倍差距

# 安同 OS：笔记本背光问题调查

• 问题暂时解决了，下面是一些碎碎念：

- 这个问题是“新旧世界”问题吗？

- “新旧世界”是一对本来就模糊不清，后来更是被人用得十分混乱的概念，需要澄清和纠正

- 世界是复杂的，这是不以主观意志为转移的

- 哪怕是在所谓“旧世界”系统的开发阶段，pwm\_request 也已经早就被标明弃用了

- 弃用的 API 一定要弃用

• 后续：ACPI 初始化代码能推上游吗？

- 上游不会接受专用于树外模块的支持代码

- 考虑同时给上游的 7A2000 DC 驱动增加背光控制，技术上是可行的，但需要克服非技术困难

# Linux From Scratch

- 已完成 GCC-15.2、Binutils-2.45、Glibc-2.42 的集成
  - Binutils 回合了链接器松弛的符号大小修复
- x86 版本已经切出 12.4-rc1 分支，预计会在 9 月 1 日发布 12.4 版本，将尽量同步发布龙架构的 12.4 版本

# Arch Linux for Loong64

- Chromium 139 更新 (by wszqkzqk)

- 将用于 Loong64 平台 compiler-rt 路径的修复补丁合并到了 Arch Linux 上游的构建文件中

- <https://github.com/lcpu-club/loongarch-packages/pull/685>

- Pixi 0.52.0 更新及上游完全支持 (by wszqkzqk)

- Pixi 在其 0.52.0 版本中已经完全支持了 Loong64 架构

- 简化补丁

- 仅需处理由于缺乏预编译 Python 站点包导致的测试失败问题

- <https://github.com/lcpu-club/loongarch-packages/pull/693>

# Arch Linux for Loong64

- 新增**Electron 37** (37.2.6-1) 支持 (by wszqkzqk)

- 额外修复 electron\_runtime\_api\_delegate.cc 中 std::string 比较样式导致的构建问题

- 暂禁用xnnpack (tensor lite的可选神经网络库)

- <https://github.com/lcpu-club/loongarch-packages/pull/688>

- binutils: 向后移植松弛化后符号大小的修复

- <https://github.com/lcpu-club/loongarch-packages/pull/689>

- glibc 构建 (by wszqkzqk)

- 禁用 Loong64 上尚未支持的 sframe 以通过构建

- <https://github.com/lcpu-club/loongarch-packages/pull/690>

# Arch Linux for Loong64

- 跳过 python-redis 构建测试问题 (by wszqkzqk)
  - 添加了补丁以跳过 Loong64 上由于**浮点精度**问题而失败的测试，确保 python-redis 能够正常构建和使用
  - <https://github.com/lcpu-club/loongarch-packages/pull/687>
- 纯 Python 包
  - 浮点误差导致结果不匹配，原因待查明 (libm? 向量?)
- Nodejs构建时check要求调整 (by wszqkzqk)
  - nodejs 24的v8的wasm存在严重可用性问题
  - 恢复check要求 (仅忽略已知无影响项目)，避免升级到问题版本
  - 版本滞留nodejs 23 (v8上游暂无处理)
  - <https://issues.chromium.org/issues/429974680>

# Arch Linux for Loong64

- devtools-loong64 开发工具调整 (by wszqkzqk)

- 增加了 -N 标志到 patch 命令中，避免反向补丁阻塞拉取，提升开发工具的鲁棒性

```
.section ".blob", "aw", @progbits
file
# e_ident
.ascii "\177ELF"
.byte 0x02 # ELF_CLASS64
.byte 0x01 # ELF_DATALE
.byte 0x01 # EV_CURRENT
.byte 0x00 # FT_03
.byte 0x00 # FT_12
.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:
```



```
.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, 0x200000
```

## 社区事务

```
.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

# Loong 1-2-3

- LoongArch 兼容性数据库正式改为与域名相同的名称：Loong 1-2-3
- 网站现在开始存放龙芯主要产品的芯片参数（仅收录 LoongArch 指令集产品，MIPS 时期产品不考虑收录）
- 加入 CPU 对比页面，现在支持对至多 4 款 CPU 产品进行对比
- 目前仍长期接受硬件信息

### Loong 1-2-3

搜索文档 CtrlK

主页 兼容性列表 更多文档 芯片参数 参与贡献 语言 设置 刷新

- CPU
  - CPU 对比
  - 龙芯3号
    - 3A6000
      - 3A6000-HV
      - 3A6000-BLL
      - 3A6000-LL
      - 3A6000-i
      - 3A6000M
    - 3B6000
      - 3B6000-8(暂定)
      - 3B6000-12(暂定)
      - 3B6000M
    - 3C6000
      - 3C6000/S
      - 3C6000/D
      - 3C6000/Q
    - 3A5000
    - 3C5000
  - 龙芯2号
    - 2K3000

		
<b>3C6000/S</b> <a href="#">删除</a>	<b>3C6000/D</b> <a href="#">删除</a>	<b>3C6000/Q</b> <a href="#">删除</a>

基本信息			
产品系列	3C6000	3C6000	3C6000
市场定位	服务器	服务器	服务器
CPU 规格			
核心数	16	32	64
线程数	32	64	128
微体系结构	LA664	LA664	LA664
最高频率	2.2 GHz	2.1 GHz	2.0 GHz
一级指令缓存	64 KB	64 KB	64 KB
一级数据缓存	64 KB	64 KB	64 KB
二级缓存(合计)	4 MB (256 KB * 16)	8 MB (256 KB * 16 * 2)	16 MB (256 KB * 16 * 4)

# Conda 在龙架构上的移植

## 历史回顾

- 2022年4月：龙芯平台开源软件移植愿景清单[提到 Conda](#)
- 2022年10月：liuxiang [试图推动](#) conda-forge 的龙架构移植
- 2023年3月：zhangna 提交了 **lief** 的龙架构支持
- 2024年2-4月：zhoumingtao, Shi Pu-jin, zhao jiazhong, Wojciech Muła 等完善了 **simdutf**的龙架构支持

## 目标

- 在龙架构上使用 Conda 作为包管理器并使用 conda-build 编译软件包，实现开发环境的构建

## 进展

- 在先前基础上，给 Conda 包管理器 mamba 的[添加了龙架构支持](#)
- 完成了龙架构 Conda 环境的构建

## 后续计划

- 推进龙架构支持上游化并提供一键式安装包

# 2K0300 DDR 驱动的许可证问题

- 2K0300 SoC 的 U-Boot 移植目前已在 [Gitee 开源](#)
- 然而 DDR 初始化相关的代码（drivers/ram/loongson）并未完全开源，初始化逻辑以一个二进制静态库提供（libmem\_config\_new\_abi.a）
- U-Boot 项目的许可证为具有传染性的 GPL-2.0-or-later，发布在此基础上二次开发的二进制而不提供 DDR 部分的源代码存在法律风险
- 相关代码是否有开源计划？
- 如果没有开源计划，需要其他方案才能让第三方放心地分发 U-Boot 二进制
- 希望知情人员可以提供明确指引

# 龙芯杯决赛：社区宣讲、展示与交流会！

• 龙芯杯决赛将于 8 月 23-25 日在东北林业大学举办

• 恭喜各位优秀的同学们入围决赛！

• 期间，龙芯爱好者社区将组织社区宣讲、交流及展示活动

• 安排在 8 月 24 日团体答辩赛答辩后，欢迎各位参赛学生及周边院校的同学参加！

• 另有各类最新龙芯设备（3B6000、3A6000 笔记本与 2K3000 开发板）展示

• 更有抽奖环节！

• 宣讲环节邀请来自社区的龙架构开发者前来分享社区工作体验和心得体会，目前已有：

• 白铭骢：龙芯社区生态建设与行动指南

• xen0n：龙芯的“世界线”历史

• 根据日前分发给参赛同学的问卷，同学们对发行版、内核、工具链、x86 转译与游戏应用相关话题最感兴趣

• 如果您希望参与宣讲，请速联系白铭骢！

• 我们计划为中选讲者提供差旅报销

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
```

```
.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 - # 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
```

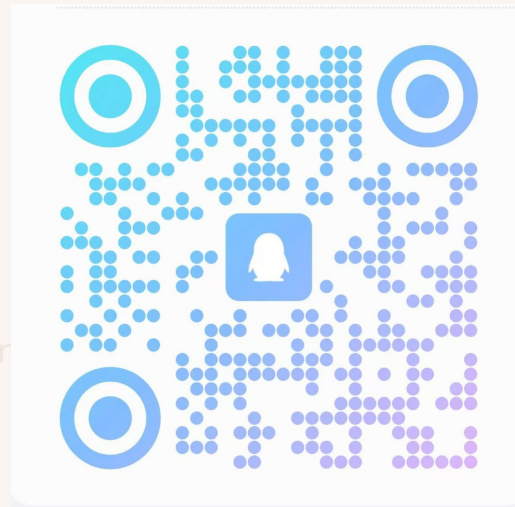
```
.short 0 # e_shstrndx
```

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

```
phdr:
```



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