

deepin



动手，拒绝焦虑！

文科生的 Linux 内核贡献之旅

白铭骢

以用为本 拥抱创新

www.deepin.org

今日要点

- 课题主旨
 - 开源软件和开源社区的建设源自于每个人的行动
 - 开源贡献的形式多种多样，且大多源自于日常遇到的问题和需要
- 我的案例
 - 从未接受过系统性计算机教育和培训，经验几乎完全来自十余年的社区工作
 - 具体到内核，我的贡献经验不多于一年
- 视角问题
 - 我的主要工作经历在社区运营而非技术研发，但我的参与“资历”确实较长
 - 参考价值（远远）低于启发价值
 - 无意证明自己的参与深度抑或经历的可复制性
 - 我只想鼓励各位放下恐惧与焦虑，拥抱行动与实践

自我介绍

- 白铭骢
 - 安同开源社区贡献者、安同 OS 维护者
 - 外号：“特首”、果冻、白总（别）、...
- 学业与“专业训练”经历
 - 中国现代史、俄语专业毕业
 - 2011 创办安同开源社区，期间组织安同 OS 维护工作
 - 补丁整理、代码审阅、本地化、...
 - 2024 年学业中断偶然入行
 - 发行版与社区运营

Linux 内核

- 万物之源
 - 由 Linus Torvalds 发起的自由软件内核项目
 - 各“Linux 操作系统”、Android、Chrome OS 的核心组件
 - 也是几乎所有国产操作系统使用的内核
- 参与方
 - 各硬件与软件厂商、支持全职开发者的基金会和组织
 - 驱动维护者、安全与代码研究人员、文档与本地化工作者
 - 以及发现问题后尝试研究并提交修复、改进补丁的业余爱好者
- 臭名昭著的“殿堂”
 - 缘由：具有核心地位的开源项目，往往令人心生畏惧
 - 个人：Linus Torvalds 及部分从说话直接到苛刻待人的维护人员
 - 恶果：对项目参与者的能力甚至道德要求产生近乎神化的认识

概念与关键词

- 内核上游
 - 今天讨论的参与场所是由 Linux 基金会运营、Linus Torvalds 直接管理的 Linux 内核源码树
- 子系统
 - 简单地说, Linux 各“组件”或维护方向
 - 如: PCI 子系统、USB 子系统、DRM/GPU 子系统
 - 有时精确到厂商驱动, 如 AMD 公司职工维护的 amdgpu 驱动
- 邮件列表
 - Linux 内核开发中一切补丁提交、评审及其他沟通工作的场所
 - 一般基于子系统划分板块 (subspace)

概念与关键词

- 维护者
 - Linux 各子系统负责评审、合入补丁的人员
 - 也有负责发布职责的人员，如负责主线版本发布的 Linus Torvalds
 - 负责 LTS 及稳定 (stable) 分支评审和发布管理的 Greg KH 和 Sasha Levin
- 主线
 - 广义：前面提到的上游源码树中的代码（包括各维护分支）
 - 狭义：版本管理意义上，首个分支版本（如 6.14）及其测试版本 (-rc)
 - 主线化：将本地的内核修改（补丁）提交子系统、分支（如 stable）邮件列表
 - 经由相关子系统或分支维护者评审通过合入后，修改即“主线化”了
- 回合 (Backport)
 - 将基于某一更新分支（往往是 master 及 stable 主线）的更改移植到维护中的其他分支（如从 master 回合至 linux-6.6.y 分支）

如何参与内核贡献？

- 问题驱动（常见思路）
 - 我希望解决 Linux 内核中存在的什么问题？
 - 抑或：我希望通过参与内核锻炼什么能力、获得什么资历？
- 寻找方向：可“深”可“浅”
 - 如同开源软件生态中的分工一样，Linux 内核也有许多贡献渠道
 - 除了“给内核写驱动”这种创造性工作之外，实际上还有许多参与方式
 - 尤其是对于技术上尚未掌握“核心技术”的朋友们：
 - 子系统文档审阅校对、翻译
 - 针对信息、输出错误，及编译器警告和错误的修复
 - 针对机型特定问题提交（尤其是内核中已经有记录在案、有规律的）规避
 - 挑战自我：针对硬件驱动实现原理，根据规格文档提交修复

文科生的贡献之旅

- 从去年八月开始...
 - Loongarch: KVM: Add KVM hypercalls documentation for LoongArch
 - 为龙架构 KVM Hypercalls 新增内核文档（提供语法修缮建议）
 - drm: amdgpu: fix comment about amdgpu.abmlevel defaults
 - 修正 amdgpu.abmlevel 参数的描述说明（-1 对应“自动”而非“禁用”）
 - ASoC: amd: yc: fix internal mic on Xiaomi Book Pro 14 2022
 - 修复 Xiaomi Book Pro 14 2022 麦克风不可用的问题
 - iommu: intel: apply quirk_iommu_igfx for 8086:0044 (QM57/QS57)
 - 修复 Intel QM57/QS57 芯片组设备在启用 Intel VT-d 后出现图形卡顿和随机重启复位的问题（联想 ThinkPad X201/X201s）
 - platform/x86: hp-wmi: mark 8A15 board for timed OMEN thermal profile
 - 修复 HP OMEN 8 (2022) 上功耗释放错误受限的问题

文科生的贡献之旅

- scripts: install-extmod-build: add missing quotation marks for CC variable
 - 修复 v6.14 开始 install-extmod-build 脚本未正确为 CC 变量加引号的问题
- platform/x86: thinkpad_acpi: Add battery quirk for ThinkPad X131e
 - 修复联想 ThinkPad X131e 未能正确报告电池电量的问题
- drm/xe/regs: remove a duplicate definition for RING_CTL_SIZE(size)
 - 修复 Intel xe 驱动中对 RING_CTL_SIZE(size) 的重复定义
- MAINTAINERS: remove undeliverable entries in RADEON and AMDGPU DRM DRIVERS
 - 删除 radeon 及 amdgpu 驱动中已不可达的维护者邮箱地址
- drm/xe: enable driver usage on non-4KiB kernels
 - 修复 Intel xe 驱动在非 4KiB 内核分页平台（如龙架构）上不可用的问题

文科生的贡献之旅

- 这些补丁修了什么？如前面提到的几个路径
 - 子系统文档审阅校对、翻译
 - 针对信息、输出错误，及编译器警告和错误的修复
 - 针对机型特定问题提交（尤其是内核中已经有记录在案、有规律的）规避
 - 针对硬件驱动实现原理，根据规格文档提交修复
- 我是如何找到这些贡献机会的？
 - 被动信息：关注 Linux 内核邮件列表中我感兴趣的部分
 - 发行版维护过程中主动或被动发现的问题
 - 在龙架构和非主力机器上安装时遇到的问题（最常见的一类）
 - 用户使用需求（如在龙芯上使用 Intel 独显进行视频剪辑、游戏娱乐）
 - 测试内核更新时发现的问题
 - 新购入的设备

贡献门槛

- 第一步：找到兴趣点或需要解决的问题（问题意识）
- 第 2-N 步：学习掌握基本技能、耐心学习流程、练习与试错
- 所需技能
 - 对日志的解读和代码定位（基础 C 语言能力）
 - 查阅邮件并定位引入过程、原因和可能后果（英语阅读能力、基础 Git 命令）
 - 在必要时进行二分排查（Git 二分命令 - git bisect）
 - 英文与正式邮件书写（收件人、抄送、回复格式、.mbox 导入等）
- 流程与规范资料
 - [提交补丁：如何让你的改动进入内核（内核上游文档）](#)
 - 提交内核补丁的两个途径
 - 使用 [b4 代理](#)或使用 [git send-email（简要文档）](#)
- 时间关系，今日不作演示，仅展示思路和成品（欢迎私下联系）

令人胆颤？

- 其实还好！
 - 摸着石头过河：无需一下掌握所有的技能
 - 开卷考试：邮件列表沟通（包括评审）节奏往往较慢，不用急
 - 尤其对于业余新手来说，更不用急于完成编写提交流程
- 演示一下
 - 我们来预演一个华硕 F441U 的一则稳定问题
 - Realtek RTL8723BE 网卡频繁报 PCIe AER（高级错误报告）
 - 频率极高以至于影响正常使用，甚至可能导致死机
 - 理想的入门案例：
 - 涉及编码量较少，着重于信息收集
 - 属于机型特定规避，内核中有较为成熟的机制可以直接利用
 - 驱动维护较为活跃，维护者较为友好

问题历史：上游介入前

- 这一问题实际上早在 2016 年就有用户报告，为华硕 Skylake 机型
 - [\[SOLVED\] RTL8723BE WiFi&BT Card and Intel Skylake problem](#)
 - 真的已经“解决(solved)”了吗，非也！
 - 用户一开始根本无法使用该网卡：网卡被 asus-nb-wmi 驱动错误屏蔽
 - 而后，用户报告网卡速率异常低，但并未在意
- 2016 - 2023 年：许多关于性能的零星报告，均为华硕 6 - 7 代机型
 - 用户开始发现内核日志 (dmesg) 中有大量 PCIe AER 报错，每秒数条
 - 亦有用户指出机器性能异常低，不止是网卡
 - 有社区好友指出可以通过禁用 PCIe ASPM（主动状态电源管理）解决问题
 - 即 pcie_aspm=off 内核参数
 - 在绝大多数情况下被当作了最终解决方法
 - 真的足够好吗？这基本相当于禁用了整台机器的 PCIe 电源管理！

问题历史：上游介入与用户报告

- 2023 年，社区好友 Jian-Hong Pan 在[内核问题追踪系统](#)报告了该问题
 - 还是华硕的 Skylake/Kaby Lake 机型
 - 完全没有得到任何答复
 - 最终自己[编写并提交了补丁](#)，在所有 Skylake/Kaby Lake 桥片上禁用了 RTL8723BE 无线网卡的 ASPM 功能
 - 经历三版补丁审阅，未能与维护者就实现方式达成一致，最终搁置
- 2025 年初，用户安装安同 OS 时报告稳定性问题
 - 无法完成安装（进度不动或死机），查阅内核日志发现大量 PCIe AER 报错（结合 `lspci -tv` 命令可知错误来自 0000:00:1c.5 - 对应 RTL8723BE 网卡）
 - 查看 `dmidecode`（DMI 信息解析工具）及 `lspci`（PCI 设备查看工具），发现属于上述典型华硕 Kaby Lake + RTL8723BE 机型
 - 发现上述补丁，但亦认同修复方式错误

行动与实验：ASPM 问题规避

- 根据 Jian-Hong Pan 收到的反馈
 - 对两代处理器的所有 PCIe 桥片与 RTL8723BE 的组合进行规避过于武断
 - 那么多使用 RTL8723BE 的设备，都有问题吗？
 - 实际上我们是不确定的
- 那么我们要如何实现，才更有可能得到上游认可呢？
 - 根据问题历史
 - 我们只确定特定的华硕 Skylake/Kaby Lake 机型搭配该款网卡会导致问题
 - 缩小规避范围
 - 与其去匹配所有两代桥片的 PCI 设备 ID，我们应该匹配网卡的 OEM 设备 ID
 - 让我们来看看 lspci 输出

行动与实验：ASPM 问题规避

```
03:00.0 Network controller [0280]: Realtek Semiconductor Co., Ltd. RTL8723BE PCIe  
Wireless Network Adapter [10ec:b723]
```

```
Subsystem: Lite-On Communications Inc Device [11ad:1723]
```

```
Flags: bus master, fast devsel, latency 0, IRQ 17, IOMMU group 13
```

```
I/O ports at c000 [size=256]
```

```
Memory at ef100000 (64-bit, non-prefetchable) [size=16K]
```

```
Capabilities: <access denied>
```

```
Kernel driver in use: rtl8723be
```

```
Kernel modules: rtl8723be
```

网卡设备 ID: 10ec:b723

通用 ID, 所有 RTL8723BE 网卡均使用该 ID

OEM ID (或称子系统 ID) : 11ad:1723

机型专属 ID, 特定机型或网卡 OEM 版本使用的 ID
可以尽可能地匹配到对应的机型

行动与实验：ASPM 问题规避

- 定位网卡驱动的 ASPM 设置 (drivers/net/wireless/realtek/rtlwifi/rtl8723be/sw.c)

```
static void rtl8723be_init_aspm_vars(struct ieee80211_hw *hw)
{
    [...]
    /* This setting works for those device with
     * backdoor ASPM setting such as EPHY setting.
     * 0 - Not support ASPM,
     * 1 - Support ASPM,
     * 2 - According to chipset.
     */
    rtlpci->const_support_pciaspm = rtlpriv->cfg->mod_params->aspm_support;
}
```

行动与实验：ASPM 问题规避

- 继续挖掘默认值

```
static struct rtl_mod_params rtl8723be_mod_params = {  
    [...]  
    .aspm_support = 1,  
    [...]  
};
```

- 这告诉我们什么？
 - 该网卡驱动默认启用 ASPM (1)，而非根据芯片组支持探测 (2)
- 下一个问题，“根据芯片组支持探测”的逻辑是怎样的？

行动与实验：ASPM 问题规避

寻找 `const_support_pciaspm` 的赋值来源 (`drivers/net/wireless/realtek/rtlwifi/pci.c`)

```
static void _rtl_pci_update_default_setting(struct ieee80211_hw *hw)
{
    [...]
    /*Set HW definition to determine if it supports ASPM. */
    switch (rtlpci->const_support_pciaspm) {
    [...]
    case 2:
        /*ASPM value set by chipset. */
        if (pcibridge_vendor == PCI_BRIDGE_VENDOR_INTEL)
            ppsc->support_aspm = true;
        break;
    [...]
}
```

行动与实验：ASPM 问题规避

- 我们目前掌握了什么信息？
 - RTL8723BE 网卡默认开启 ASPM，不论用户在使用什么芯片组（桥片）
 - Realtek 无线网卡的驱动编写者认为搭配 Intel 芯片组是一定可以开启 ASPM 的
- 哟，这不就有问题了吗？我们目前调试发现的问题
 - RTL8723BE 在华硕 F441U 这台 Intel 机型上开启 ASPM 会导致 AER 报错
 - 这与上面二者的定义均相悖
- 我们该怎么办？
 - 以严谨为原则最小化规避该问题
 - 我们目前知道问题的根源是网卡，因此应该从网卡设备上规避
 - 写代码！
- 另注：记住上面的信息，这是我们后续编写正文需要的证据

行动与实验：规避代码

- 根据第一点认识我们知道，默认开启 ASPM 不可取
 - 因为我们手上的机型开启网卡 ASPM 会导致问题
 - 根据现有惯例，ASPM 的机型 + 网卡型号匹配特判并非在具体型号的驱动内完成
 - 而是在 rtlwifi 驱动的共享 PCI 支持代码 (pci.c) 中，如下例

```
pcie_capability_read_word(rtlpci->pdev, PCI_EXP_LNKCTL, &init_aspm);  
if (rtlpriv->rtlhal.hw_type == HARDWARE_TYPE_RTL8192SE &&  
    ((u8)init_aspm) == (PCI_EXP_LNKCTL_ASPM_L0S |  
                       PCI_EXP_LNKCTL_ASPM_L1 | PCI_EXP_LNKCTL_CCC))  
    ppsc->support_aspm = false;
```

“如匹配到 RTL8192SE 及支持 PCIe ASPM L0s/L1 及 CCC 支持的平台，则禁用 ASPM”

行动与实验：规避代码

- 根据这个例子，我们将对 rtlwifi/pci.c 进行修改
 - 前面我们知道，华硕特定机型的 RTL8723BE 网卡带有特殊 OEM/子系统 ID
 - 而前面提到的数个报告中的子系统 ID 均与手上 F441U 的相同 (11ad:1723)
 - 那么补丁的内容便是 “为子系统 ID 为 10ec:1723 的 RTL8723BE 禁用 ASPM”

```
static void _rtl_pci_update_default_setting(struct ieee80211_hw *hw)
{
    [...]
+   if (rtlpriv->rtlhal.hw_type == HARDWARE_TYPE_RTL8723BE &&
+       (rtlpci->pdev->subsystem_vendor == 0x11ad &&
+        rtlpci->pdev->subsystem_device == 0x1723))
+       ppsc->support_aspm = false;
    [...]
}
```

行动与实验：上机测试

- 发布补丁前，先测试
- 测试结果
 - 该补丁确实禁用了这一 OEM 型号上 RTL8723BE 网卡的 ASPM
 - 机器正常启动，网络功能及性能正常，且没有卡顿及死机的情况
 - 其余设备的 ASPM 正常启用，节电功能正常
- 修复完美否？
 - 很可惜，不完美
 - 启动时仍有 rtl8723be 驱动触发的 AER 错误
 - 不同的是，这个错误在 rtl8723be 完成载入后就停止了
 - 很可能还有其他上游设备作怪，抑或 rtlwifi 的共享代码并没有及时禁用 ASPM
 - 暂且当个测试修复吧！

行动与实验：补丁“论文”编写

- 前面提到，内核对补丁质量有一定要求
 - 代码：符合 80 列宽、硬 Tab、条件表达式对齐等要求
 - 对于本例，还需要附加注释说明（可略加修改作为 Git 提交正文）
 - Git 提交内容：
 - 标题：标明子系统、驱动信息，简要说明修复主旨
 - wifi: rtlwifi: disable ASPM for rtl8723be with subsystem ID 11ad:1723
 - 正文：说明修复原理（可复用代码注释）
 - RTL8723BE 网卡默认开启 ASPM，不论用户在使用什么芯片组（桥片）
 - Realtek 无线网卡的驱动编写者认为搭配 Intel 芯片组是一定可以开启 ASPM 的
 - RTL8723BE 在华硕 F441U 这台 Intel 机型上开启 ASPM 会导致 AER 报错
 - 匹配华硕 F441U 上网卡的子系统 ID 11ad:1723 并禁用 ASPM 以规避问题
 - 测试结果
 - 末尾（签准区）：签署、参考链接、合作者信息、修复记录、Stable Cc 等

行动与实验：补丁“论文”编写

- 补丁生成的大致过程

```
git add drivers/net/wireless/realtek/rtlwifi/pci.c
# 将该文件添加到提交内
git commit -s
# 编写提交信息，附带签署信息
git format-patch -1 --rfc
# 用最近提交的一笔提交生成补丁（并标记为征求意见稿，RFC）
```

- 这些命令的使用方式较为容易摸索，留作习题（大雾）；时间关系暂不展示
- 来看看完成编写、签署和引用的补丁生成出来是怎样的
- ...
- 倒数第 N 站：提交补丁（同样留作习题）

功德“圆满”后

- 补丁提交给对应列表后，事情还没结束！
 - 上游维护者可能提出修改意见或质疑
 - 请积极答复，补充必要的信息，或作出必要修改
 - 如有疑问，请不要犹豫并提出来
 - 积极沟通可能未必得到有用甚至友好的答复
 - 可悲的现状，但请保持自信和积极态度
- 恭喜你成为光荣的 Linux 内核贡献者！

今天我讲了什么（希望你们理解到的）

- Linux 内核如同其他开源项目一样，亟需外部参与
 - 硬件种类千千万，就算是神也不可能一直兼顾
 - 硬件支持随着时间推移劣化是正常现象
 - 硬件实现并非一定完美，代码层面上的规避可提升用户体验
 - 您的补丁可能让许多人受益，或让他们使用 Linux 工作时心情更为愉快
 - 内核贡献可大可小，人人为我，我为人人
- 内核贡献的参与深度极深
 - 今天讲述的例子亦有一定深度，但也只是基于现有机制的补充修缮工作
 - 但这不代表工作量就一定小：问题历史调查、人际沟通也是一门功课！
 - 该学的还是得学，但不用学精了再动手
 - 学以致用，事不宜迟！

行动派们，后会有期！

- 今天的分享作为教程是不及格的
 - 最近工作变动，精力有限
 - 课件有许多思考、验证和设计不周到的地方，还请谅解！
- 一起行动起来！
 - 我希望带动各路有意改善 Linux 维护质量的人参与上游开发
 - 内核贡献涉及许多知识和技能点，其中也包括非技术性的
- 欢迎与我联系，我将尽力提供帮助和指导
 - 微信：mingcongbai
 - QQ：2879378668
 - Telegram：@JeffBai
 - 邮件：jeffbai@aosc.io

