内核贡献指南

如何向 Linux 内核提交你的补丁

程子丘

2023 年 3 月 26 日

操作系统讨论班
华科大操作系统内核贡献团队
Security Pride, HUST

警告！

由于 CTEX 引擎可能产生的各种字符替换问题，请不要复制该文件中的任何命令到你的终端中。

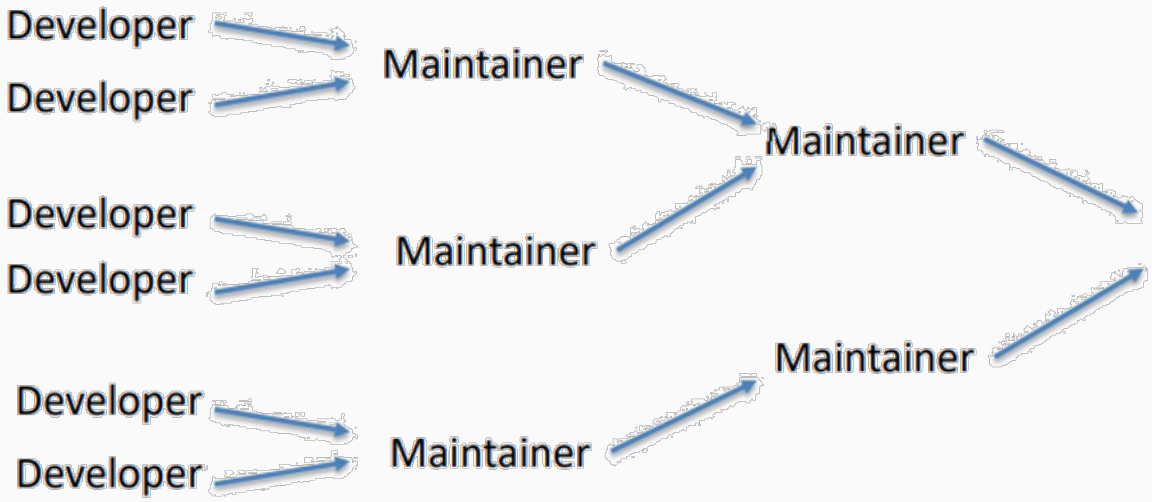
如需要复制命令，请使用 commands.txt 文件。

背景

内核代码仓库如何维护？

通过 Linux Kernel Mailing List（LKML）[1]

分支合并过程：



[1] http://vger.kernel.org/vger-lists.html#linux-kernel

环境

必备工具：

- 文本编辑器（VsCode, Sublime Text, [Neo]Vim, ...）
- Git [2]
- Perl
- 电子邮件客户端（git-sendmail [3]）

编译环境：[4]

- Linux 平台 [5]
- GNU 编译工具链，包括编译器、汇编器、链接器（build-essensial）

---

[2]一个交互式的 Git 学习沙盒：`https://oschina.gitee.io/learn-git-branching/`

[3]使用 git-sendmail 发送补丁邮件参见 `https://mudongliang.github.io/2018/03/20/setting-up-git-send-email-with-gmail-to-send-linux-kernel-patch.html`

[4]此处只列举部分，具体环境可以参考 Documentation/process/changes.rst

[5]如果没有 Linux 环境，可以使用 Windows Subsystem for Linux 进行替代

制作补丁

获取内核源代码

从主线克隆整个 Git 仓库

```
git clone https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git
```

获取源代码后，可以在源代码目录下的的 Documents/process/submitting-patches.rst 中获得贡献指南文档。

做出修改

根据具体的 Bug 原因，对代码进行修改。

可参见第6页中的例子。

在提交 Patch 之前，可以新建一个分支，避免影响以后主分支的 pull。

```
git checkout -b ${BRANCH_NAME} [6]
```

[6] 从此页开始，所有命令行中的变量均需替换为具体值。

编码规范

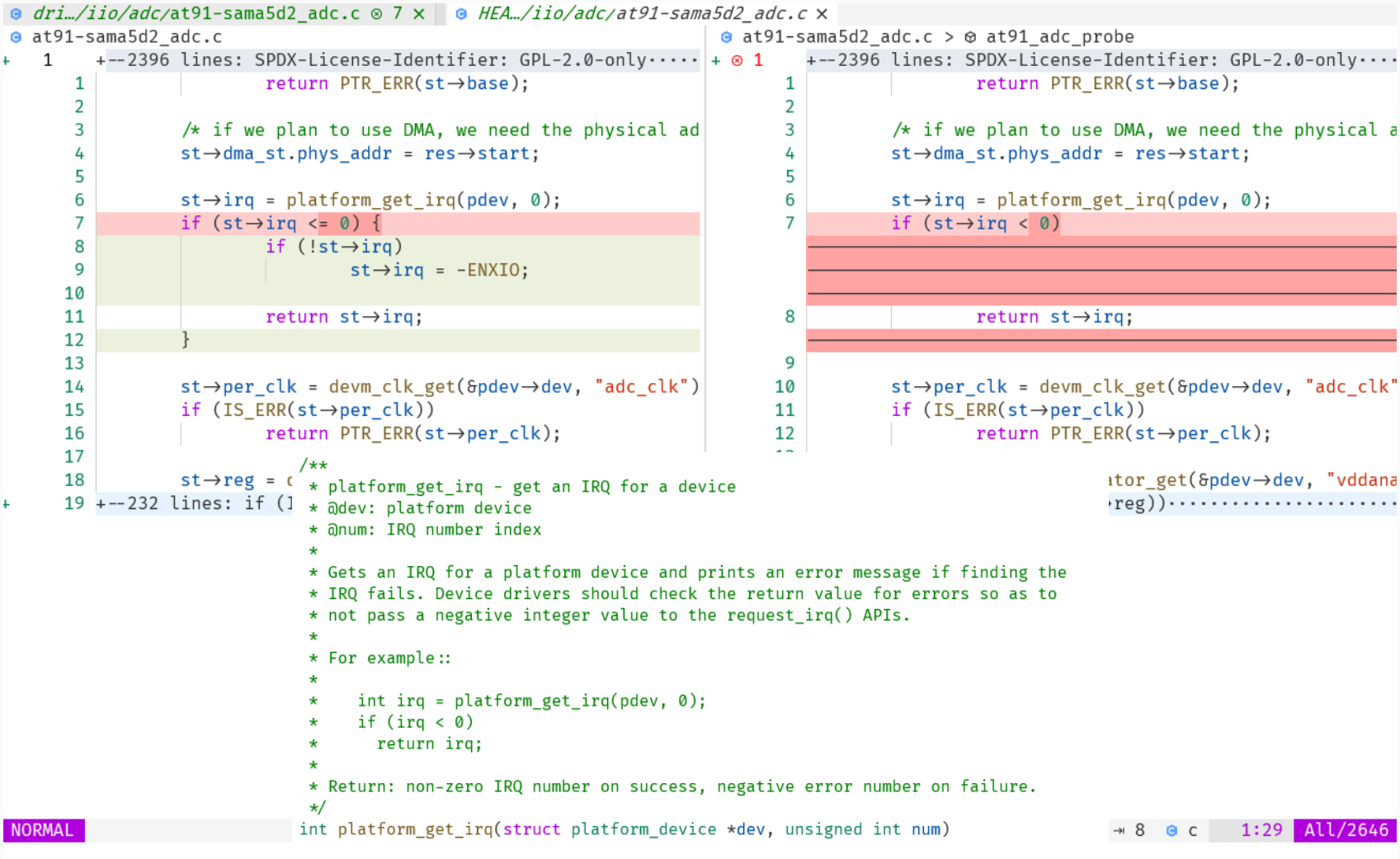
在修改内核代码前，强烈建议阅读内核编码规范。

文档在内核代码库的 Documentation/process/coding-style.rst 中。

★ 其中最需要注意的是

- **缩进**：内核编码规定使用制表符（Tab）缩进，缩进宽度为 8。
- **最大宽度**：每行最多 80 列，以确保在老式终端机中代码不需要自动换行（word wrap）也能显示完全。

一个修改的例子


dri…/iio/adc/at91-sama5d2_adc.c ⊗ 7 ×
HEA…/iio/adc/at91-sama5d2_adc.c ×
at91-sama5d2_adc.c
at91-sama5d2_adc.c > at91_adc_probe
+--2396 lines: SPDX-License-Identifier: GPL-2.0-only
return PTR_ERR(st->base);
/* if we plan to use DMA, we need the physical ad
st->dma_st.phys_addr = res->start;
st->irq = platform_get_irq(pdev, 0);
if (st->irq <= 0) {
if (!st->irq)
st->irq = -ENXIO;
return st->irq;
}
st->per_clk = devm_clk_get(&pdev->dev, "adc_clk")
if (IS_ERR(st->per_clk))
return PTR_ERR(st->per_clk);
st->reg = d
+--232 lines: if (!
/**
* platform_get_irq - get an IRQ for a device
* @dev: platform device
* @num: IRQ number index
*
* Gets an IRQ for a platform device and prints an error message if finding the
* IRQ fails. Device drivers should check the return value for errors so as to
* not pass a negative integer value to the request_irq() APIs.
*
* For example::
*
*   int irq = platform_get_irq(pdev, 0);
*   if (irq < 0)
*   return irq;
*
* Return: non-zero IRQ number on success, negative error number on failure.
*/
int platform_get_irq(struct platform_device *dev, unsigned int num)
NORMAL
8 c 1:29 All/2646

编译内核

在提交之前，可以通过编译内核判断补丁是否能正确编译。[7]

1. 生成一个 config
   - 使用现有的 config
     cp XXX.config .config
   - 自己配置 config，加上修改过的部分
     make menuconfig (config xconfig gconfig) [8]
2. 开始编译

```
make -j${THREADS}[9]
```

[7] 编译内核的具体步骤，可以参见fedora21-compile-linux-kernel.html
[8] menuconfig 需要安装 ncurses，xconfig 需要安装 qt，gconfig 需要安装 gtk
[9] 此处 THREADS 变量为并发线程数

Commit!

在完成修改后，可以采用 diff 工具检查你的修改。

```
git diff
```

检查没有问题后，开始 commit！

```
git commit -asev
```

规范的 Commit message

subsystem: summary phrase

详细描述，如需分段，段与段之间需要一个空行

Signed-off-by: Author

patch 签名标签及用途

- **Signed-off-by:** 表示签名者参与了补丁的开发，或者在补丁的交付路径中。
- **Acked-by:** 表示没有直接参与补丁的准备或处理，但希望表示并记录他们对补丁的认可。这通常是受影响代码的 maintainer 既没有贡献也没有转发补丁时使用。
- **Co-developed-by**：当几个人在一个补丁上工作时，它用于给共同作者提供属性。其后必须紧跟 Signed-off-by。
- **Reviewd-by**：表示一种意见声明，即补丁是对内核的适当修改，没有任何严重技术问题。此标签用于向审阅者表示信任，并告知维护人员对补丁所做的审阅程度。
- **Fixes**：指明一个 commit，标记表示该补丁修复了先前提交中的问题。它用于轻松确定错误的来源，这有助于审查错误修复。此标记还帮助稳定内核团队确定哪些稳定内核版本应该接收您的修复。

```
Fixes: 54a4f0239f2e ("KVM: MMU: make kvm_mmu_zap_page() ...")
```

一个例子

```
iio: adc: at91-sama5d2_adc: remove dead code in `at91_adc_probe`

From the comment of platform_get_irq, it only returns non-zero IRQ
number and negative error number, other than zero.

Fix this by removing the if condition.

Signed-off-by: Cheng Ziqiu <chengziqiu@hust.edu.cn>
Reviewed-by: Dongliang Mu <dzm91@hust.edu.cn>
```

驱动中的 subsystem 规范写法：

subsystem: area: driver: change

生成并提交补丁

准备建立你的 patch 文件：Subject 前缀

当发送邮件时，需要为邮件主题添加一个前缀，这个前缀保存在你的 patch 文件中，格式为

[PATCH (vP) (M/N)] [10]

以下前缀是规范的：

[PATCH]

[PATCH v2]

[PATCH 2/5]

[PATCH v2 01/27]

[10]括号内的内容表示可以省略，P 为补丁版本号，M 和 N 表示补丁系列中有 N 个补丁，该补丁为第 M 个。M 和 N 高位需要用 0 填充，以便其字典序为其真实顺序。

华中科技大学邮箱配置指南

在使用 git-send-mail 发送邮件前，需要配置好你的 SMTP 设置。

1. 设置邮箱密码：通过统一身份认证进入学校邮箱，并设置一个邮箱账户密码。
2. 修改你的.gitconfig [11]

```
[sendemail]
    smtpEncryption = ssl
    smtpServer = mail.hust.edu.cn
    smtpUser = YOUR_HUST_EMAIL_ADDRESS
    smtpServerPort = 465
```

[11]当然，在.gitconfig 中不进行任何配置并在发送时指定命令行参数也是可行的。

建立你的 patch 文件

通过 git format-patch 建立你的 patch 文件，其默认 subject 前缀为 [PATCH]。

```
git format-patch -1
```

如果是第二次提交 Patch，需要自定义你的 subject 前缀和 changelog。

```
git format-patch –subject-prefix=”PATCH v2” HEAD...master
```

正确的 patch 格式 [12]

```
<commit message>
...
Signed-off-by: Author <author@mail>
---
V2 -> V3: Removed redundant helper function
V1 -> V2: Cleaned up coding style and addressed review comments

path/to/file | 5+++--
...
```

[12] submitting-patches.rst:704

一个 patch 例子

```
From: Cheng Ziqiu <chengziqiu@hust.edu.cn>
Date: Thu, 9 Mar 2023 22:42:17 +0800
Subject: [PATCH v3] iio: adc: at91-sama5d2-adc: remove dead code in
 `at91-adc-probe`

From the comment of platform-get-irq, it only returns non-zero IRQ
number and negative error number, other than zero.

Fix this by removing the if condition.

Signed-off-by: Cheng Ziqiu <chengziqiu@hust.edu.cn>
Reviewed-by: Dongliang Mu <dzm91@hust.edu.cn>
---
v2 -> v3: Change subject to make it regular.
v1 -> v2: Change commit message from SoB to Reviewed-by.

 drivers/iio/adc/at91-sama5d2-adc.c | 6 +-----
 1 file changed, 1 insertion(+), 5 deletions(-)

diff --git a/drivers/iio/adc/at91-sama5d2-adc.c b/drivers/iio/adc/at91-sama5d2-a
index 50d02e5fc6fc..168399092590 100644
......
```

内部 Review 邮件列表

邮件列表链接：

`https://groups.google.com/g/hust-os-kernel-patches`

在发送邮件时，在 Cc 中添加下面邮件地址

hust-os-kernel-patches@googlegroups.com

检查并发送

在发送 Patch 之前，检查你的 patch 文件

```
./scripts/checkpatch.pl ${PATCH_FILE}
```

可以通过下面的脚本得到相关 maintainer 信息：

```
./scripts/get_maintainer.pl ${PATCH_FILE}
```

发送你的 Patch

然后通过`get_maintainer.pl`获取 maintainer，并发送你的 patch 给 maintainer。[13]

```
git send-email –to-cmd=”./scripts/get_maintainer.pl –nogit
–nogit-fallback –norolestats –nol”
–cc-cmd=”./scripts/get_maintainer.pl –nogit –nogit-fallback
–norolestats –nom” ${PATCH_FILE}
```
[14]

[13]关于 Patch 如何发送，可以参考`https://mudongliang.github.io/2021/06/21/git-send-email-with-cc-cmd-and-to-cmd.html`

[14]此处参数也可以采用.gitconfig 进行配置

疑问？

飞书


慕冬亮邀请你加入

仅支持区号为 +86 的手机号加入以下企业

华科大操作系统内核贡献团队

二维码失效时间：永久

请在手机应用商店搜索“飞书”并下载