



Welcome
欢迎

Writing Better
Commit Messages 101
如何高效撰写
Git Commit

杨欣辉
cyan@cyano.uk

白铭骢
jeffbai@aosc.io

/ CIP 什么是 Commit?

- 动词：将若干项修改归集在一起的动作。
- 名词：将若干项修改归集后形成的记录。
- 体现方式：每个 Git Commit 可视作一个补丁 (Patch) 。

* 修改指对 Git 树内的文件进行的修改。

* “归集”指的是 `git add` 操作。

/ CIP Commit 的数据结构

```
$ git show HEAD
```

```
commit cf2ba92481d56ab923ccae96f29764e871979451
```

```
Author: Hans-Peter Nilsson <hp@axis.com>
```

```
Date:   Mon Feb 27 20:44:46 2023 +0100
```

```
testsuite: No xfail infoleak-vfio_iommu_type1.c bogus for  
default_packed
```

There are no messages about padding for targets that don't pad, i.e. default_packed. Noticed for cris-elf, verified for pru-elf at gcc-testresults@.

/ CIP Commit 的数据结构

```
$ git show HEAD
```

```
diff --git a/gcc/testsuite/gcc.dg/plugin/infoleak-vfio_iommu_type1.c
b/gcc/testsuite/gcc.dg/plugin/infoleak-vfio_iommu_type1.c
```

```
index 51ad5db2bab..af320b6b17e 100644
```

```
--- a/gcc/testsuite/gcc.dg/plugin/infoleak-vfio_iommu_type1.c
```

```
+++ b/gcc/testsuite/gcc.dg/plugin/infoleak-vfio_iommu_type1.c
```

```
@@ -37,8 +37,8 @@ int vfio_iommu_type1_get_info(unsigned long arg)
    info.cap_offset = 0;
}
```

```
- /* The padding bytes (20-23) are uninitialized, but can't be written
```

```
- back, since minsz is either 16 or 20. */
```

```
- return copy_to_user((void *)arg, &info, minsz) ? -14 : 0; /* { dg-bogus "exposure" ""
{ xfail *-*-* } } */
```

```
+ /* The padding bytes (20-23, but applicable just for targets with padding) are
```

```
+ uninitialized, but can't be written back, since minsz is either 16 or 20. */
```

```
+ return copy_to_user((void *)arg, &info, minsz) ? -14 : 0; /* { dg-bogus "exposure" ""
{ xfail { ! default_packed } } } */
```

```
// TODO: false +ve due to not handling minsz being either 16 or 20
```

```
}
```


/ CIP Commit 的数据结构

- Commit ID (SHA-1 Hash)
- Commit 作者 (通过 git config 命令设置)
- Commit 时间戳
- Commit 标题
- Commit 详尽描述 (可选)

/ CIP 你是谁？

通过如下命令即可设置 Commit 作者信息（Git 身份）。此后，在本机中所有仓库的 Commit 均会以该身份提交。请注意，该设置仅对当前用户有效。

```
git config [--global] user.name "Your Name"
```

```
git config [--global] user.email "your@email.tld"
```

如果某个仓库需要由他人提交，可为该仓库单独设置（不使用 `--global` 开关）Git 身份。

/ CIP 不成文的规定

- 一般来说，Commit 消息的格式如下：

Message title

↓

Long message body (Optional)

- 如果修改非常简单，无需使用长消息，用简短的标题描述即可
- 注意：Commit 标题应以动词的一般现在式开头，清晰有效地概括更改内容和目标

/ CIP 撰写 Commit 消息的目的

- 简而言之，记录修改，供自己和团队参考
- 描述此次修改的目的（修复、更新和实现功能等）、内容和效果（可将有关的问题 / 工单编号包含其中）
- 长久来说，Commit 消息就像开发日志一样，用于索引项目的演变和趋势

/ CIP Commit 消息标题示例

- 按修改类型（新功能、 Bug 修复等）：
 - feat: implement Git snapshot update method
 - fix: set VER
 - update: adapt method() to upstream changes

/ CIP Commit 消息标题示例

- 按修改的模块（或文件）：
 - completions/bash: fix completions for instance names
 - config: adapt to toml API changes
 - tree-wide: update clap to 4
- 注意，每个项目应该确定一种标题写法，并保持统一

Src: <https://github.com/AOSC-Dev/ciel-rs/commits/master>

/ CIP 复习：共同点

- Commit 消息标题一定以修改的模块或修改类型开头
 - 每个项目必须统一开头规则
- 而后紧跟着的部分必须以动词开头
- 明确描述该 Commit 所做的事情或目的
- 标题不应长于 80 个字符（最好短于 60 个）

/ CIP The Good

- - 2eb942a8c6 celluloid: rebuild for mpv 0.35.1
 - bf8e0ddb86 haruna: rebuild for mpv 0.35.1
 - 037b00b9a6 mpv: update breakage
 - e8829db6c8 mpv: update to 0.35.1
 - 317bf36d00 yt-dlp: update to 2023.02.17
 - 408d864784 dpkg: update to 1.21.18
 - 4210c0b55b aoscdk-rs: update to 0.5.5
 - 326e6f6133 thermald: update to 2.5.2
 - 4609170e21 fish: update to 3.6.0
 - 2679e5defe bcache-tools: fix typo `Bcacheefs` => `Bcache`
 - 22ff2672fa systemd-boot-friend: update to 0.25.1
 - cb6bc2c09c systemd-boot-friend: update to 0.25.0
 - 38995f7946 pushpkg: update to 0+git20230220
 - fc1fab15de nvidia: upgrade to 525.89.02 to support (SHAME!) 4070Ti
 - ced1ea7a6d nvidia+390: 390.157, disable kernel / environment CC version check
 - 05ea9c5b2b glibc: introduce trigger for ldconfig
 - c81475af30 llvm-runtime+wasi: adapt to new build system
 - f50b14180b llvm-runtime+wasi: update to 15.0.7
 - 83d468dcde llvm: use non-architectural-specific directory for ...

<https://github.com/AOSC-Dev/aosc-os-abbs>

/ CIP The Good

- C
 - e0324e2629e c++: allocator temps in list of arrays [PR108773]
 - 4214bdb1d77 testsuite: add various -Wanalyzer-null-dereference false +ve test cases
 - ec4bc86b439 middle-end: On emergency dumps finish the graph generation.
 - 8e26ac4749c AArch64: Fix codegen regressions around tbz.
 - 96abc822246 libstdc++: Implement LWG 3820/3849 changes to cartesian_product_view
 - 065c93b89c5 libstdc++: Implement LWG 3796 changes to repeat_/chunk_by_view [PR109024]
 - 95827e1b9f7 libstdc++: Make views::single/iota/istream SFINAE-friendly [PR108362]
 - c6232ba229a Fix PR 108980: note without warning due to array bounds check
 - 3df9760d566 libstdc++: extraneous begin in cartesian_product_view::end [PR107572]
 - f366fdfeec0 libstdc++: Really fix symver for __gnu_cxx11_ieee128::__try_use_facet [PR108882]
 - afe1f0c251d c++: CTAD for less-specialized alias template [PR102529]
 - 9e6170098d5 c++: fix alias CTAD [PR105841]
 - 30556bf81f4 c++: hide __is_deducible for GCC 13
 - 148cbb153da c++: add __is_deducible trait [PR105841]
 - d11e088210a Enable UTF-8 code page on Windows 64-bit host [PR108865]
 - a6457974a1f LRA: For clobbered regs use operand mode instead of the biggest mode
 - ace65db9215 middle-end/108995 - avoid folding when sanitizing overflow
 - bad177e8487 range-op-float: Fix up reverse binary operations [PR109008]
 - 288bc7b5d17 libgomp: Fix default value of GOMP_SPINCOUNT [PR 109062]

<https://gcc.gnu.org/git/?p=gcc.git;a=log>

/ CIP The Bad

```
cyan@Raptor [ dont-do-this@master ] $ git log --decorate --oneline
bca6614 (HEAD -> master) PLZ WORK
5751779 why it does not work
515fa25 Add more code
f1f5452 Update
8c77b2a fix bad code
edad355 update
9a8e119 new
cyan@Raptor [ dont-do-this@master ] $
```


/ CIP 如何定义 “不好”

- 模糊不清：Commit 标题过于简要，无从得知作者更新或修复的内容
- 效率低下：内容不描述更改（如 "PLEASE WORK", "WHY IS THIS NOT WORKING" 等毫无意义的字句）
- 规矩很简单：确保一系列 Commit 连串起来可以描述开发更改历史，维护良好的 Commit 记录通过少许修改即可成为“更改日志 (Changelog)”一类面向用户的文档



/ CIP 基本编写策略

- 每个 Commit 应描述针对特定模块或文件的修改
- 对于管理操作（如修改仓库布局），模块名称应为 “tree-wide（全树）”，“misc（杂项）” 抑或 “meta（统括）”
- 如果修改的文件很多，且包含不同类型的更改，请使用 `git add` 命令逐一添加同一个模块的修改，按照模块分别提交
 - 如果同一文件中有多钟不同修改，亦可使用 `git add -p` 逐步添加更改
 - 如此一来，发现问题时就可以通过 `git reset` 撤销更改，而不是采取“改回去”的操作

/ CIP 基本编写策略

- 如果有一系列 Commit 连续修改同一个文件，请使用变基 (Rebase) 功能将这些 Commit 合为一个 (Squash)
- 在验证修改前不要着急将修改推送至远端仓库
- 在推送前整理本地 Commit
 - 尤其注意检查 Commit 历史，确保内容整洁有效

/ CIP 特殊情况

- 修改了大量文件
 - 如果 Bug 修复或新功能实现等涉及大量文件修改，应将这些文件分别提交，并在长消息中解释本提交属于一系列大更改
- 修改了一个文件，但其中修改类型多于一种
 - 按照修改类型分别提交该文件的不同部分
 - 使用 `git add -p` 命令，将文件中不同部分的修改分开提交



Thanks for listening!

Stay Tuned...