# 我和 LLDB 那些事

主讲人：杨靖华

# 三个方面

- 前言
- 和编译器打交道
  - 和 Debugger 打交道
- 和社区开发者打交道
- 和标准打交道

- 可能不会有第二个人遇到的问题：如何在 x86 上跨架构调试一个调试器。
- 如果不犯什么“原则性”错误，开发者对你还是很友好的。
- 可能很难读，资料也很少，得有耐心

# 0. 一个全自动 CI 的重要性

```
[ 2270s] Summary of Failures:
[ 2270s]
[ 2270s]  51/160 b_ulimit                  FAIL            5.71s   exit status 2
[ 2270s]  54/160 b_ulimit/shcomp           FAIL            7.11s   exit status 2
[ 2270s]  70/160 directoryfd               FAIL            6.45s   exit status 2
[ 2270s]  76/160 directoryfd/shcomp        FAIL            5.61s   exit status 2
[ 2270s]  88/160 exit/shcomp               FAIL            7.00s   exit status 1
[ 2270s]  91/160 exit                      FAIL            7.86s   exit status 1
[ 2270s] 124/160 options/shcomp            FAIL          110.97s   exit status 2
[ 2270s] 125/160 options                   FAIL          114.89s   exit status 2
[ 2270s] 138/160 substring/shcomp          FAIL           27.08s   exit status 1
[ 2270s] 159/160 special-dev-paths         FAIL            1.68s   exit status 1
[ 2270s] 160/160 special-dev-paths/shcomp FAIL             2.02s   exit status 1
[ 2270s]
[ 2270s] Ok:                 149
[ 2270s] Expected Fail:      0
[ 2270s] Fail:               11
[ 2270s] Unexpected Pass:    0
[ 2270s] Skipped:            0
[ 2270s] Timeout:            0
[ 2270s]
[ 2270s] Full log written to /home/abuild/rpmbuild/BUILD/ksh-2020.0.0/riscv64-openEuler-linux-gnu/meson-logs/testlog.txt
```

- 图片来源：https://build.tarsier-infra.com/package/live_build_log/Factory:RISC-V/ksh/Roll/riscv64

# 对于 Open Build Service 的想法

- 做到了合格 CI 应有的水平，在多人协作中已经提供了较大便利

- 与 gitee 不绑定，不能自动拉取
- 将源码打包为 tar.gz 增加了修改的难度
- 学习成本较高，可能忘记关键步骤导致浪费时间

# 和编译器打交道

- LLDB 作为横跨几年时间长度的项目，其环境必然有所改变，所以 LLVM 会给出“推荐”软件环境：

| Package | Version |
| --- | --- |
| CMake | >=3.20.0 |
| GCC | >=7.1.0 |
| python | >=3.6 |
| zlib | >=1.2.3.4 |
| GNU Make | 3.79, 3.79.1 |

图片来源：https://llvm.org/docs/GettingStarted.html#software

## C++17 library features

| C++17 feature | Paper(s) | GCC libstdc++ | Clang libc++ | MSVC STL | Apple Clang | Intel Parallel STL | Sun/Oracle C++ Standard Library | Embarcadero C++ Builder Standard Library |
|---|---|---|---|---|---|---|---|---|
| std::void_t | N3911 | 6 | 3.6 | 19.0 (2015)* | Yes | N/A | | 10.3 |
| std::variant | P0088R3 | 7 | 4 | 19.10* | 10.0.0* | N/A | | 10.3 |
| std::make_from_tuple() | P0209R2 | 7 | 3.9 | 19.10* | Yes | N/A | | 10.3 |
| std::has_unique_object_representations | P0258R2 | 7 | 6 | 19.11* | Yes | N/A | | 10.3 |
| std::gcd() and std::lcm() | P0295R0 | 7 | 4 | 19.11* | Yes | N/A | | 10.3 |
| std::not_fn | P0005R4 P0358R1 | 7 | 3.9 | 19.12* | Yes | N/A | | 10.3 |

图片来源：https://en.cppreference.com/w/cpp/17

# IDE 的重要性

- 增/改大量代码
- 减少人为失误
- 简化的交叉编译流程
- 更易用的调试器

# 可能要去 “迁就” 一下 buildbot

The error appears to be:

```
../../lldb/source/Plugins/Process/Utility/RegisterInfos_riscv64.h(67,5): error: constant expression evaluates to
-1 which cannot be narrowed to type 'uint32_t' (aka 'unsigned int') [-Wc++11-narrowing]
    DEFINE_GPR64(pc, LLDB_REGNUM_GENERIC_PC),
```

So I'm not sure that patch will fix everything but let's see.

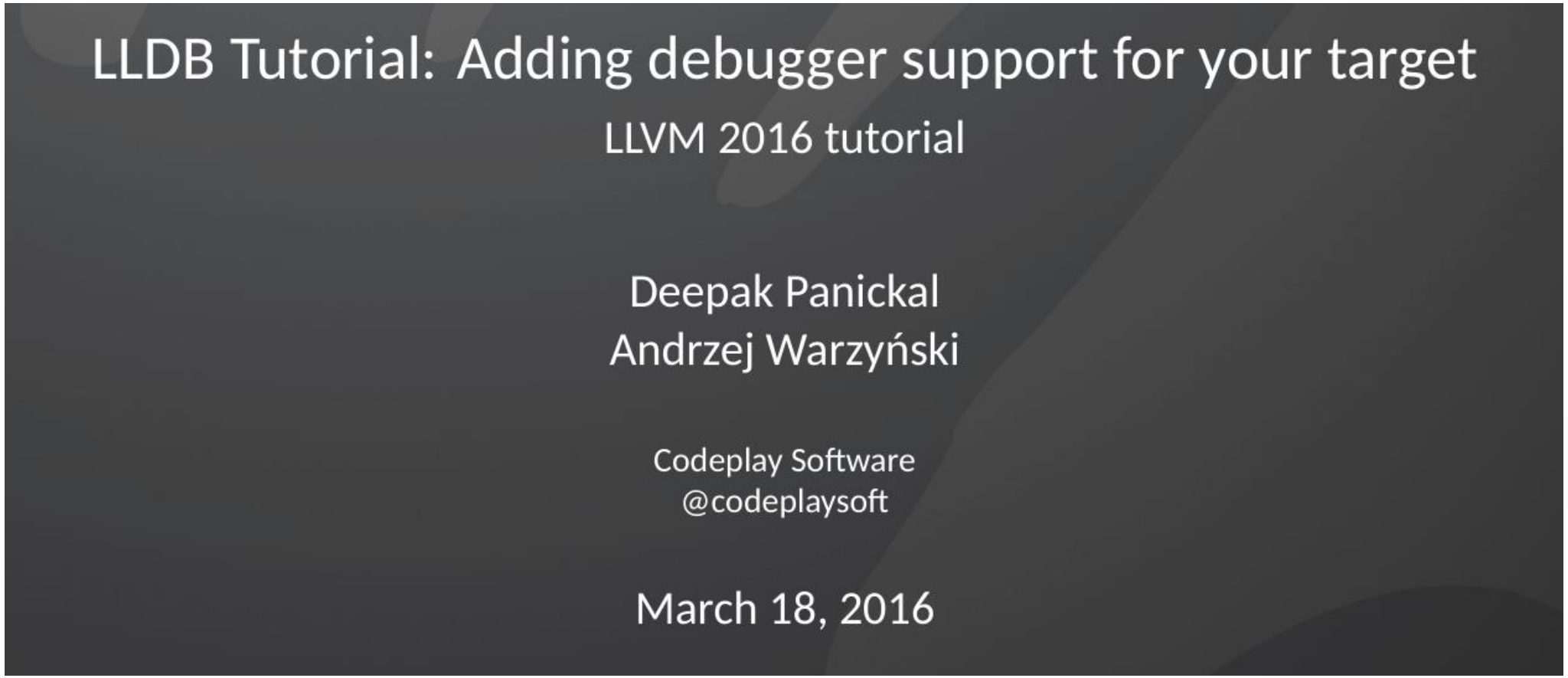
```
125
126   // mock pc regnum
127   dwarf_gpr_pc = UINT32_MAX,
128 };
129
```

```
125
126   // mock pc regnum
127   dwarf_gpr_pc = 11451,
128 };
129
```

# 编译成功了，然后呢

- Segmentation Fault
- 'A' packet returned an error: -1
- (program seems 寄了)

# 使用 lldb-server 获得更多的 debug 信息




内容来源：https://llvm.org/devmtg/2016-03/Tutorials/LLDB-tutorial.pdf

# 在 2023 年还在使用 printf 调试大法

- 暴论：不能自动化的 log 全是 printf

- 在使用 lldb-server 获取全部 log 信息后，发现 'A' packet returned an error: -1 之前进行了 ThreadResume 和 wait，说明 ThreadResume 失败且没有进行 log 输出。

# [LLDB] Handle possible resume thread error

Closed Public

Authored by **Emmmer** on Aug 16 2022, 4:05 PM.

## Details

**Reviewers** DavidSpickett

**Commits** rG8ed3e75c96d9: [LLDB] Handle possible resume thread error

### SUMMARY

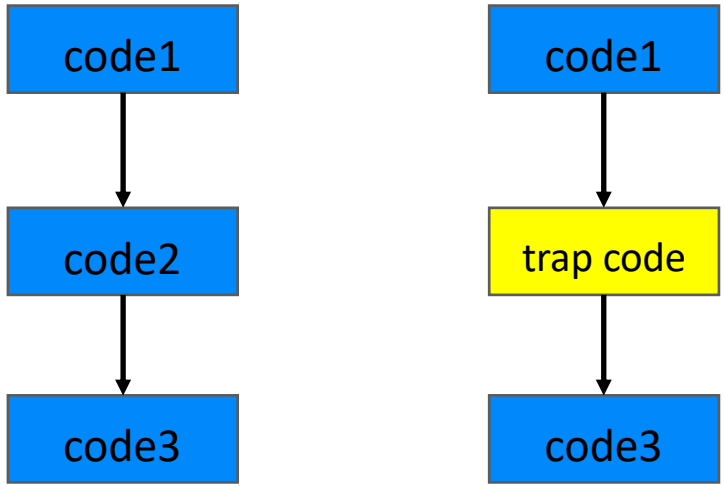
In this switch case we didn't handle possible errors in `ResumeThread()`, it's hard to get helpful information when it goes wrong.

# trap code 陷阱

- 我们能同时在两个地方找到 trap code 的定义，分别为
  - **NativeProcessProtocol.cpp**
  - **Platform.cpp**
- 一处用来“set breakpoint”，一处用来“resume breakpoint”
  - set breakpoint: `(lldb) b main`
  - resume breakpoint: `(lldb) c`

# Debug 流程示意图

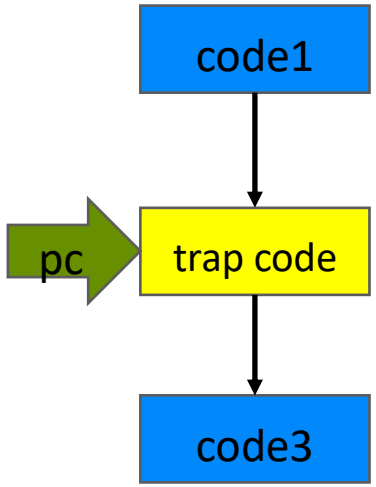
(lldb) b code2



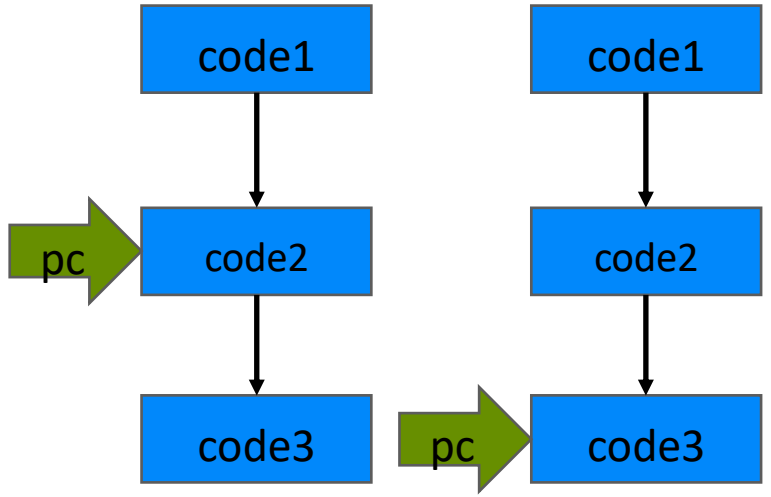




(lldb) r




(lldb) c



# LLDB 单步调试流程

- 在单步调试中，PTRACE_SINGLESTEP 会向内核申请每一条单步指令的执行
- 如果 Hardware single-step 被宿主支持，则我们能直接获得结果
- 如果宿主不支持 Hardware single-step，LLDB 会启动一个模拟器，即 EmulateInstruction{TARGET} 去执行并且获取结果

# 差点变成灵异事件的 Initialize()

- 在 **SystemInitializerLLGS.cpp** 中存在一些预定义头文件，一些模块的功能需要开发者通过加入一些宏定义在这里 Initialize（注册）plugin 才能被调用。

- 都怪 software single-step

# LLDB 不足之处

- 有些代码过于臃肿
- 部分功能结合较为复杂，比较难梳理
- Log 全部是手动输出，不利于 Debug
- 给出的报错不够直观，如 'A' packet returned an error: -1 对用户来说是无效信息

# 和社区开发者打交道

——以 https://reviews.llvm.org/D140092 为例

## [NFC][LLDB] Using namespace llvm in EmulateInstructionRISCV

Closed Public

Authored by **Emmmer** on Dec 15 2022, 7:58 PM.

**Details**

**Reviewers** DavidSpickett

**Commits** rG260ba2f22422: [NFC][LLDB] Using namespace llvm in EmulateInstructionRISCV

**SUMMARY**

The `EmulateInstructionRISCV` uses a lot of types and functions in `llvm` and `lldb`, this change is to make the code look cleaner.

PS: This patch should be merged before **D140032**.

# 对 maintainer 的要求做出修改

**Harbormaster** completed remote builds in **B203303: Diff 483117**. Dec 15 2022, 8:04 PM

D **DavidSpickett** accepted this revision. Dec 15 2022, 8:43 PM

LGTM

D This revision is now accepted and ready to land. Dec 15 2022, 8:43 PM

D **DavidSpickett** added a comment. Dec 15 2022, 8:43 PM

Please tag [LLDB] in the commit title.

**Emmmer** retitled this revision from *[NFC] Using namespace llvm in EmulateInstructionRISCV* to *[NFC][LLDB] Using namespace llvm in EmulateInstructionRISCV*. Dec 15 2022, 8:44 PM

# patch 已经变成 commit 后的补救

Closed by commit **rG260ba2f22422: [NFC][LLDB] Using namespace llvm in EmulateInstructionRISCV** (authored by **Emmmer**). · Explain Why

Dec 15 2022, 9:32 PM

This revision was automatically updated to reflect the committed changes.

**Emmmer** added a commit: **rG260ba2f22422: [NFC][LLDB] Using namespace llvm in EmulateInstructionRISCV**.

L

**labath** added a subscriber: **labath**.

Dec 16 2022, 2:54 AM

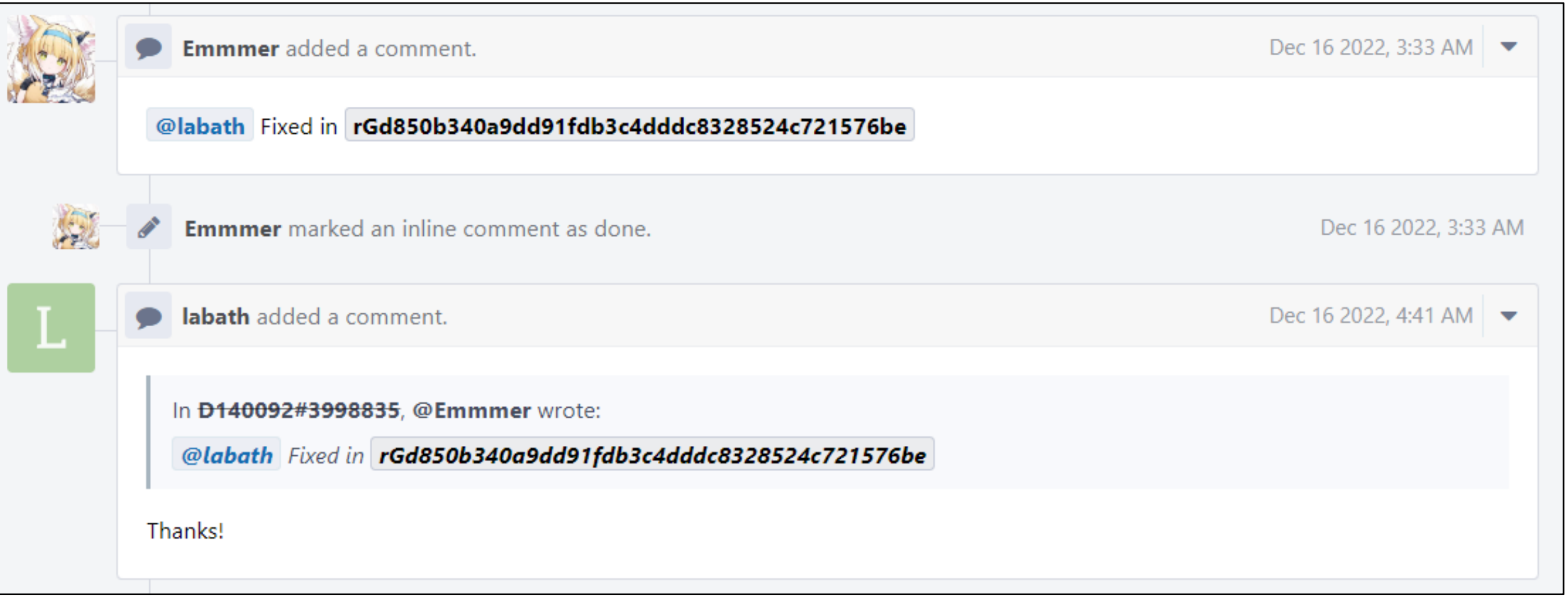
**labath** added inline comments.

**lldb/source/Plugins/Instruction/RISCV/EmulateInstructionRISCV.h**

21–22

You cannot use `using namespace` in a header file. That pollutes the lldb_private namespace for everyone who includes this file. Please revert this immediately.

# Just push it to the main branch



**Emmmer** added a comment. Dec 16 2022, 3:33 AM

@labath Fixed in **rGd850b340a9dd91fdb3c4dddc8328524c721576be**

**Emmmer** marked an inline comment as done. Dec 16 2022, 3:33 AM

**labath** added a comment. Dec 16 2022, 4:41 AM

> In ~~**D140092#3998835**~~, **@Emmmer** wrote:
>
> *@labath Fixed in* ***rGd850b340a9dd91fdb3c4dddc8328524c721576be***

Thanks!

# 补充：作为 new contributer，如何为 LLVM 提交较大 patch

1.准备 Phabricator 账号（可通过 GitHub 登录）

2.准备 arcanist （感觉……不如……Git）

3.先提出一个小型 patch，熟悉维护人员，获得社区信任

4.小型 patch 被接受后，向 Chirs 发邮件申请仓库编辑权限

5.使用提前准备的 arcanist land 你的 patch （arc land --revision 123456 --onto main）

6.做好 CI 爆炸被 revert 的心理准备

**Request to join LLVM Github org**

发件人：Emmmer<yjhdandan@163.com>

收件人：clattner<clattner@llvm.org>

时　间：2022年07月29日 12:08 (星期五)

发送状态：发送成功　查看详情

Hi Chirs,

I sincerely apply for commit permission of llvm/llvm-project because I have a patch ready to land (https://reviews.llvm.org/D130686).
I follow the instruction from (https://github.com/llvm/llvm-project/blob/main/llvm/docs/DeveloperPolicy.rst#obtaining-commit-access) and get your email, and please correct me if there is anything wrong.

My Github username: SEmmmer
My Email address: yjhdandan@163.com

Thank you in advance

## Re: Request to join LLVM Github org

发件人：Chris Lattner<clattner@llvm.org>

收件人：  Emmmer<yjhdandan@163.com>

时　间：2022年07月30日 08:51 (星期六)

翻译成中文

Great, welcome to the LLVM team! I just sent an invitation through GitHub, please make sure to read through LLVM Developer Policy, and try a test commit!

-Chris

# 和标准打交道

- 包括且不限于：
- RISC-V Instruction Set Manual: Unprivileged ISA
- RISC-V Instruction Set Manual: Privileged Architecture
- RISC-V Calling Convention
- RISC-V Debug Specification
- 以及最恐怖的：RISC-V "V" Vector Extension

# 和标准打交道

- 在尝试理解 RISC-V 标准时，可能会出现误解，可能来自于文档的表述不够清晰，也可能来自于读者对某些概念的理解程度。
- 在一些比较复杂的功能中，文档可能会比较细碎跳脱，对于母语非英语的同学可能产生阅读上的阻碍
- Calling Convention 对于 pc 寄存器只字未提，导致我在安排Dwarf文件时十分苦恼

# 参考文献


- https://www.redhat.com/zh/topics/devops/what-is-ci-cd