zCore on RISCV

Run zCore on qemu of riscv

车春池

2020.08.29

@ 华中科技大学 东十二舍 308

前言

zCore 是一个微内核，它是用 Rust 语言重新实现了 Google 开发的操作系统 Fuchsia 的微内核 zircon 的一个项目。

下一代 Rust OS： zCore 正式发布

zCore 目前只支持 x86 架构（mips ？）

目标 => 在 riscv 架构下的虚拟机 qemu 中跑 zCore

github 传送门：zCore-riscv

相关工作介绍

- 在 zCore 的开发过程中，开发者已经考虑到将来可能要支持 riscv，架构相关的代码都做了标注
- kernel-hal-bare/src/arch/riscv.rs
- 构建 zCore 的各个 Rust 库除了 rboot 外基本上都支持 riscv
- rCore 是支持 riscv 的
- 刘丰源学长曾将 zCore 移植到了 mips 架构上

实现方案

- 基于 OpenSBI 搭建运行环境 zCore-riscv（zCore 在 x86 和 riscv 上 bootloader 的区别）
- 一步步将原 zCore 中的模块移植到 riscv 的运行环境中
- 为 zCore-riscv 添加测试模块
- 在底层的模块中为架构相关的代码添加 riscv 支持（unimplemented!宏）
- 在裸机环境下调用 loader 层（遇到了障碍）

具体实现方法

- 基于 OpenSBI 的运行环境是参照 rCore-Toturial 来搭建的
- 原 zCore 中 kernel-hal 等模块是作为一个个 crate，被 zCore 调用，而在新搭建的运行环境中是作为一个个 mod 来处理
- 在 no_std 环境下的 Rust 项目想要使用单元测试的话，十分麻烦。因此添加了一个模块 fake_test 用于测试
- 添加 #[cfg(any(target_arch = "riscv32", target_arch = "riscv64"))] 标注
- 在 rust_main 函数中调用 loader 层封装的函数

当前遇到的障碍和解决思路

障碍：

- Fuchsia 官方目前不支持 riscv，而且将来可能也不打算支持 riscv
- 当前运行环境的缺陷

解决思路：

- 放弃对接 Fuchsia && 转战 Linux 路线
- 阅读 zCore 和 rCore 的 Makefile，理解它俩是怎么运行起来的，然后再考虑怎么改善这个简陋的运行环境

一点小困难：原 zCore 代码的更新

遇到的一些小插曲

- 重写 linker.ld
- 重写 memory 模块，hal_frame_alloc，hal_frame_alloc_contiguous，hal_frame_dealloc
- 一个小疑惑（or 惊喜？）：riscv64.img，x86_64.img（待阅读 Makefile）

后续工作的方向思路

- 为底层代码添加 riscv 支持（阅读 riscv 官方文档和 rCore 中的实现）
- 完善运行环境（阅读 rCore 和 zCore 的源码和 Makefile）（RustSBI？）
- 加深对 Rust 语言本身的认识
- 同步完善记录文档

谢谢听讲！

PS：感谢老师，学长们的付出，也感谢清华大学还有鹏城实验室提供的实习机会。