

为 rCore 实现更多 Linux 系统调用

罗峻骁 宋香君



课程设计目标

- 基础目标
 - 支持 make, 能在 rCore 中从源码安装软件
 - 完善相关系统调用以支持在 rCore 中:
 - 构建有 STL 的 C++ 项目
 - 使用离线版的 cargo 构建并运行有 Rust-std 的项目
- 扩展目标
 - 支持离线版的 git
 - 完善 cargo 的网络功能
 - 完善 git 的远程仓库功能

已有工作

- rCore 对 gcc 的支持
 - 在已经实现相当数量的 syscall 后, gcc 可以无缝移植到 rCore 中
 - ~~一天, 助教说 rCore 要有 gcc, 于是 rCore 就有了 gcc~~
- 已经实现的系统调用
- 一个文件系统 (rCore-fs)

目前完成的工作

- 将 musl-gcc 和 musl-rust 工具链安装到了 rCore 上
 - 使用的官方预编译版，或者在自己的操作系统中编译
- 测试了一些简单的带有标准库的 C++ 和 Rust 程序在 rCore 中的编译及运行
- 发现了一些可能存在问题的系统调用
 - 在运行 cargo run 时，出现了奇怪的文件描述符

```
ode: 0o666
[ INFO] [0,1] => Err(ENOENT)
[ INFO] [0,1] close: fd: 18446744073709551615
[ INFO] [0,1] => Err(EBADF)
[ INFO] [0,1] close: fd: 18446744073709551615
[ INFO] [0,1] => Err(EBADF)
```

需要解决的问题

- 完善缺失的系统调用（消警告）
 - sigprocmask
 - geteuid
 - brk
 - sigaction
 - sigacti
 - flock
 - ...

需要解决的问题

- 测试更多的 C++ 和 Rust 程序
 - 还不知道哪里有测例
 - 目前的想法：支持 make 后，找一些足够大的工程进行测试
- 有关网络的系统调用
 - 不会网络原理，看助教修得怎么样

小组分工

- 罗峻骁
 - 实现和完善系统调用
 - 完成对 cargo 离线版的支持
 - 测试 C++ 和 Rust 项目

- 宋香君
 - 实现和完善系统调用
 - 完成对 make 的支持
 - 测试 C 项目