

# 基于 RISC-V 架构与Rust语言的16KiB页面操作系统

## ——操作系统 Project 报告

## 1 项目简介

### 1.1 项目协作

12011404 叶臻铭。

12012029 邓值仁。

---

### 1.2 为什么选择用Rust语言做Project?

本次 Project 我们没有使用Lab中的C语言代码，而是根据rCore的指导重新构建出一个操作系统。选择Rust有以下几个好处：

#### 1. 生产力。

##### (a) 可拔插的语言特性。

比起C语言和C++写操作系统没有任何标准库，Rust语言在设计时便考虑了裸机环境、嵌入式设备下没有标准库的问题。使得我们在编写操作系统的时候，仍然可以使用core核心库。

此外，只要在具体平台上编写相应的语言特性，便可以逐步恢复标准库的功能。

##### (b) 优质的包管理器。

比如，

相比用C语言做这次Project

##### (c) 出色的文档

比如，

相比用C语言做这次Project

##### (d) 清晰的错误提示

比如，

相比用C语言做这次Project

#### 2. 可靠性。Rust 丰富的类型系统和所有权模型保证了内存安全和线程安全，让您在编译期就能够消除各种各样的错误。<sup>1</sup>

举个例子，

#### 3. 高性能。

Rust 和C++语言一样，提供了“零开销”的抽象。我们用Rust写操作系统，不仅不会比C语言慢，还可能因为更加清晰的类型系统和所有权机制使得导致变慢的操作在编译时就消除，从而获得更好的内存利用率。

---

## 2 应用程序与基本执行环境

### 2.1 16KiB下的QEMU

### 2.2 Rust 执行环境

## 3 16KiB下内存布局与地址空间的设计

###

### 3.1 实验：cargo bench 测试16KiB页面的性能

## 4 实现多道程序、分时多任务与进程。

## 参考文献

1. Rust 程序设计语言.” <https://www.rust-lang.org/zh-CN/> (accessed Jul. 29, 2022). [↩](#)