

北京理工大学

本科生毕业设计（论文）开题报告
基于 RISC-V 的操作系统实验集成和改进

学 院： 计算机学院

专 业： 计算机科学与技术

班 级： 07712001

姓 名： 李雯

指导教师： 陆慧梅

校外指导教师：

二〇二二 年 一 月 十 一 日

毕业设计（论文）开题报告评审表

姓名	李雯	学号	1120205029	班级	07712001	专业	计算机科学与技术 (第二学士学位)
导师	陆慧梅	校外导师（副教授）		无		校外导师单位	无
论文选题	题目名称	基于 RISC-V 的操作系统实验集成和改进					
	题目性质	软件开发（ <input checked="" type="checkbox"/> ） 理论研究（ <input type="checkbox"/> ） 工程设计（ <input type="checkbox"/> ） 技术科学研究与工程技术研究（ <input type="checkbox"/> ）					
	题目来源	结合科研（ <input checked="" type="checkbox"/> ） 结合生产实际（ <input type="checkbox"/> ） 结合实验室建设（ <input type="checkbox"/> ） 自拟题目（ <input type="checkbox"/> ）					
评审组成员	姓 名	职 称		工作单位及职务		签 字	
评审意见	<p>（含：选题意义；选题是否满足毕业要求；技术方案是否可行；进度安排是否合理等）</p> <p>论文选题针对的是基于 RISC-V 的操作系统实验集成与改进，参考 rCore 实验设置，用 C 翻译 Rust 代码，作为 rCore 操作系统实验的过渡实验，并在 QEMU 硬件仿真平台上完成实验测试。</p> <p>选题具有一定教学应用价值，需要掌握一定专业理论知识，具备实践难度，符合毕业设计要求。</p> <p>技术方案从相关理论知识入手，逐步过渡到实践实验，基本具备可行性。</p> <p>进度安排包括了实验开展、实验指导文档撰写、论文撰写、外文翻译和每周周志等过程，工作安排基本面面俱到，时间安排符合毕业设计进程表，准予开题。</p>						
成 绩							
评审组长签字：		2022 年 2 月 24 日					

注：成绩以“合格”“不合格”记；评审组长为高级职称人员。

1. 选题背景、研究现状及选题意义

1.1 国内外开源操作系统

开源操作系统是公开源代码的操作系统软件，遵循开源协议[1]使用、编译和发布。自由和开放源代码软件中最著名的是 Linux，一种类似 Unix 的操作系统。通常所说的 Linux 系统是指使用 Linux 内核的操作系统，它有很多发行版，如：Ubuntu、Manjaro、Debian、Fedora、CentOS、RedHat、openSUSE、openEuler 等。Linux 可安装在各种计算机硬件设备中（如手机、平板、路由器、视频游戏控制台、台式计算机、超级计算机等）。除此之外，国内外还有许多著名的开源操作系统，虽然它们的应用领域不如 Linux 那样广泛，但却在各自的领域发挥着不可替代的作用。

● ToAruOS[2]

由伊利诺伊大学计算机科学本科生开发的具有教学意义、完全从零开始构建、类 Unix 系统的操作系统，其核心组件都是原始的，在大约 80,000 行 C 和汇编语言中提供了完整的环境，能够托管 Python 3 和 GCC，代码托管在 GitHub[3]上。ToAruOS 可在 POSIX 和 x86 架构上运行，其最终的目标是实现一个微内核。ToAruOS 的主要功能包括对进程和线程的支持、ELF 二进制的支持、运行时加载模块、管道（Pipe）和各种类型的终端设备（TTY）的支持、虚拟文件系统的支持、EXT2 文件系统的支持、信号量支持等。

● BareMetalOS[4]

BareMetal OS 是一个为 x86-64 系统开发的开源操作系统，它使用汇编语言编写且能够使用 C/C++ 开发应用程序，其代码托管在 GitHub[5]上。该系统的开发有三个适用场景，第一个是使用高性能的计算，能够作为 HPC 集群的节点，以运行高负荷的计算任务；第二个是嵌入式应用，提供了基于 x86-64 硬件的嵌入应用开发平台；第三个是提供教育和教学使用，提供了基于 x86-64 环境的汇编语言的学习和实验的环境。目前，该系统只是尽可能提供有用的功能，还没有将其打造成通用操作系统（如 Mac OS X、Linux、Windows）的计划。

● HarmonyOS[6]

鸿蒙是华为自主开发的基于微内核的全景应用分布式操作系统，可灵活部署于不同设备，如：智能屏幕、车载终端、可穿戴设备、智能手机、计算机、智能手表、无线耳塞、平板电脑等，其支持的 RAM 大小从千字节到千兆字节不等；兼容 Linux 与

Android 应用，同时该系统应用程序开发中使用的方舟编译器（ArkCompiler）也将支持 Kotlin、Java、JavaScript、C 与 C++。

1.2 国内外教学操作系统

调研国内外高校操作系统课程与实验建设[7]，以教学为初衷的操作系统不断涌现，国内有 uCore、rCore、Tinix、MOS 等，国外有 xv6、Minix、GeekOS、Nachos、OS/161 等，这些操作系统在代码和功能实现上各有差异，共同点是开源、可扩展、可移植，因而能较好地满足操作系统实验教学的需求。根据运行时依赖的指令集架构，教学操作系统可分为两类：一类基于 x86 或 Intel I-A32 指令集架构，另一类基于 MIPS 指令集架构，以下作简要介绍。

1.2.1 基于 x86 或 Intel I-A32 指令集架构

● xv6

MIT 基于 PDP-11 上的 Unix Version 6 用 ANSI 标准的 C 语言编写了 xv6，可在 x86 指令集架构上运行，开设了课程 6.828: Operating System Engineering[8]，在 xv6 和 JOS 基础上设置七个实验，课程设计内容如表 1.1。

表 1.1 MIT 6.828 基于 xv6 课程设计项目

实验编号	实验名称	实验内容
Lab1	机器启动	使用 x86 汇编语言，QEMU 模拟器编写启动程序；加载 JOS 内核镜像
Lab2	存储器管理	为内核分配和释放物理内存空间；映射虚拟地址到物理内存中的地址
Lab3	用户模式环境	通过 JOS 内核建立数据结构来跟踪受保护的用户模式环境
Lab4	抢占式多任务	JOS 中添加多处理器支持，实施轮转（RR）调度实现一个进程创建的系统调用；添加进程通信（IPC）允许时钟中断和抢占
Lab5	文件系统再生和 Shell	实现一个简单的读、写文件系统
Lab6	网络驱动	实现一个网络接口卡驱动程序
Lab7	JOS 项目	扩展和选作项目，完成 JOS 项目

● uCore

清华大学计算机系基于 x86 架构、参考 MIT 的 xv6、采用 C 语言和 x86 汇编语言，以增量方式完成 uCore OS 的设计，开设基于 uCore OS 操作系统实验课程[9]，课程设计内容如表 1.2。

表 1.2 清华大学基于 uCore OS 的课程设计项目

实验编号	实验名称	实验内容
Lab0	Coding	熟悉开发环境的运行和调试、硬件模拟器应用
Lab1	启动操作系统	完成一个能切换到 X86 保护模式的 bootloader，为启动 uCore 准备
Lab2	物理内存管理	完成一个简单的物理内存管理系统
Lab3	虚拟内存管理	借助页表机制和中断异常处理机制，完成页面出错异常处理和 FIFO 页面置换算法
Lab4	内核线程	内核线程创建、执行、切换和基本调度过程
Lab5	用户调度	用户进程创建过程和系统调用框架的实现机制
Lab6	处理器调度	基于调度器框架实现一些调度算法
Lab7	同步互斥	理解信号量和管程机制，使用同步机制解决进程同步问题
Lab8	文件系统	基于索引节点组织方式的文件系统设计实现、文件系统抽象层的设计与设计

● Minix (Mini-Unix)

荷兰阿姆斯特大学开发设计了具有微内核结构的 Minix 教学操作系统，由一万多行代码组成，五千多行代码运行在内核模式下，其它代码运行在用户模式下。系统功能较完整，结构庞杂，用于集中式教学过程中，比较难以理解和掌握。国内选择该系统作为课程设计平台有：上海交大、南京大学、南开大学等。

● GeekOS[10]

美国马里兰大学开发设计的基于 C 语言的简洁操作系统，实验项目覆盖操作系统核心内容，难度循序渐进，可以在 Linux 环境下编程，对其功能进行扩充，也可以在 Windows 下使用 Cygwin 工具（类 Unix 模拟环境）进行开发。

1.2.2 基于 MIPS 指令集架构

● Nachos (Not Another Completely Heuristic Operating System) [11]

美国加州大学伯克利分校（University of California, Berkeley）开发设计的操作系统课程平台，其运行过程可被跟踪，作为一个用户进程运行在主操作系统之上，同时还需要运行在一个模拟 MIPS R2/3000 指令集虚拟机上。Nachos 建立在软件模拟的虚拟机上，避免了编写复杂的硬件控制程序，用 C++ 和 JAVA 语言中的类来表示各个机器模拟对象。与 GeekOS 相比，只能运行在 MIPS 处理器的模拟器上，若移植到实际硬件机器中，需要使用交叉编译器才能把代码编译成 MIPS 相应的机器代码。但是该系统存在和实际操作系统不相符合的设计，比如：内核设计不受内存容量限制，这样会对实际操作系统运行机制会造成误解。

- OS/161[12]

由美国哈佛大学（Harvard University）设计，运行在与操作系统无关的 System/161 模拟器上。它只包括一个便携系统的框架结构，与处理器和平台相关的代码分开维护，用 C 语言编写、GCC 编译。与 Nachos 有很多相似之处，需要使用交叉编译器才能把代码编译成 MIPS 相应的机器代码。

1.3 基于 RISC-V 的 uCore 操作系统课程实验研究现状

2018 年清华计算机系：清华大学 2015 级学生张蔚实现 uCore 到 RISC-V 的移植，并开设课程，以操作系统基本原理为教学引导，以简洁的 RISC-V CPU 为底层硬件基础，设计并实现一个微型但全面的操作系统—uCore。期望能够采用简化的计算机硬件为基础，以操作系统的基本概念和核心原理为实践指导，逐步解析操作系统各种知识点和对应的实验，做到有“理”可循和有“码”可查，最终让读者了解和掌握操作系统的原理、设计与实现。

2018 年南开大学募集优秀学生进行实验尝试，并于 2019 年进行试点教学。

2019 年清华大学 uCore on RISC-V 32bit 操作系统课程设计实验。

2019 年麻省理工大学开设基于 RISC-V 的 xv6 操作系统实验课程 S081。

2021 年操作系统大赛，华南理工大学软件工程专业大四在读本科生彭天祥将 RISC-V uCore 由 QEMU 移植到 K210 平台，同时为其移植一个 SD 卡驱动程序。

2. 研究方案

2.1 本选题的主要任务

本论文课题首先调研国内外开源操作系统、开源教学操作系统、基于 RISC-V 体系的操作系统课程实验等，并学习 RISC-V 手册。随后，学习基于 Rust 语言的 RISC-V 64bit rCore 操作系统的实验，用 C 完成 RISC-V 64bit rCore 操作系统的移植，完成并优化相关的操作系统实验，测试实验结果，并给予分析；撰写完成详细的实验指导手册。

2.2 技术方案的选择

首先学习教学操作系统 rCore（ON RISC-V 64）的课程实验；学习 RISC-V 手册；了解 Rust 语言的基本语法。

在基本了解 Rust 语言实现 rCore（ON RSIC-V 64）操作系统的同时，模仿 rCore 实验设置，参考 uCore 和 xv6 实验中的相关实验代码，用 C 翻译 Rust 代码，进行 rCore 实验的迁移，逐步完成环境搭建、内核启动、简单批处理、分时多任务、内存管理、进程管理的实验，并在 QEMU 硬件模拟器中实现 rCore 操作系统运行。

2.3 实施技术方案所需的条件

rCore 操作系统实验环境：在 VMware 虚拟机中安装 Ubuntu（准备虚拟环境），安装必要的开发工具（VScode、交叉编译工具链）、硬件模拟器（QEMU）、调试工具（GDB）。

2.4 存在的主要问题

目前存在的主要问题是：基础过于薄弱以至于很多东西理解很慢、进度很慢。内存管理中 RISC-V 架构特有的 SV39 多级页表硬件分页机制较难掌握理论知识和进行实践，进程管理实验中关于进程创建的函数实现细节较多，关于陷入上下文、任务上下文的处理尤其需要注意指针指向的变化，还需要理解跳表 trampoline 的存在意义和实现等。

2.5 预期能够达到的研究目标

完成 rCore 教学操作系统的实验集成和改进，参照 rCore 实验设置，参考 uCore 和 xv6 操作系统实验的代码，将基于 Rust 语言的 rCore 教学操作系统用 C 语言进行迁移，完成实验并撰写实验指导书，撰写论文，提交软件。

3. 课题计划进度表

- 12月26——1月12号(寒假前)

进行相关研究课题调研：开源 OS 国内外行业标准、规范、趋势、相关行业政策和法律；

读国内外文献、相关知识，研究分析：RSIC-V 体系的 uCore 操作系统课程实验的国内外现状，撰写开题报告。

- 1月13号——2月20号（寒假期间）

完成操作系统课程实验：uCore OS (ON RISC-V) 实验；

学习相关知识：RISC-V 指令集。

- 2月21号-3月13号（第1-3周）

完成外文翻译；完成开题及答辩，实验环境部署。

- 3月14号——4月10号（第4-7周）

用C语言实现 rCore 操作系统 实验零至二（内核启动、简单批处理、分时多任务）。

撰写基于C语言的 rCore 操作系统（RISC-V 64bit）的实验指导书。

- 4月11号——5月1号（第8-10周）

用C语言实现 rCore 操作系统 实验三、四（内存管理、进程管理）。

撰写相应章节基于C语言的 rCore 操作系统（RISC-V 64bit）的实验指导书。

- 5月2号——5月22号（第11-13周）

完善实验，完善实验指导书的撰写，着手论文工作。

- 5月23号——6月5号（第13-15周）

完善、查重论文，制作毕设研究课题 PPT，并进行毕设答辩。

4. 参考文献及链接

- [1] [Licenses by Name | Open Source Initiative](#)
- [2] <http://toaruos.org/>
- [3] <https://github.com/topics/toaruos>
- [4] [Download BareMetal OS 0.5.1 \(softpedia.com\)](#)
- [5] <https://github.com/ReturnInfinity/BareMetal-OS/>
- [6] 李艳,刘丹,田小东. HarmonyOS 特点与应用前景分析[C]//2019 四川省通信学会论文集.[出版者不详],2019:146-149.DOI:10.26914/c.cnkihy.2019.049491.
- [7] 王亚,李欢,晁妍,王先传.基于教学操作系统的课程设计分析与探讨[J].山东农业工程学院学报,2020,37(11):185-188.DOI:10.15948/j.cnki.37-1500/s.2020.11.036.

- [8] [6.828 / Fall 2018 \(mit.edu\)](#)
- [9] <https://objectkuan.gitbooks.io/uCore-docs/content/>
- [10] [Microsoft Word - GeekOS Overview \(umd.edu\)](#)
- [11] [<CS162 Subversion Quick-Start Guide \(berkeley.edu\)](#)
- [12] [The OS/161 Instructional Operating System \(os161.org\)](#)
- [13] <https://nankai.gitbook.io/uCore-os-on-RISC-V64/>