

Rlinux

2022年操作系统实验

Unix-Like Operating System on RISC-V

- 2022-09-15
- 实验课助教
 - 张文龙, Yixiaonn@gmail.com
 - 季高强, reticenceji@qq.com



浙江大学
ZHEJIANG UNIVERSITY

CONTENTES

目录

#01 操作系统实验**整体**介绍

#02 操作系统实验**安排**说明

#03 操作系统实验**内容**说明
(09.15 Lab-0介绍)



Rinix#01

操作系统实验**整体**介绍

Rinix is a Simple Re-Implementation of
Unix-Like Operating System on **RISC-V**.

实验背景介绍



GNU



1991年：*Linus Torvalds*，在学校学习操作系统的课程时候自己开发了一个简单的操作系统内核放到了互联网上(GNU: [操作系统和自由软件运动](#))

1993年：大约有100多名程序员参与了Linux内核代码。这时已经是互联网的时代，他们不用相识，不用谋面，通过网络讨论，通过网络协作，推进着Linux。

1994年：Linux1.0内核发布，Linux已经成为一个基本可用的内核。在源代码的实现中，包括了功能可用的引导程序(boot)，驱动(driver)，文件系统(fs)，网络通信(net)等

.... 从此Linux的开发进程就一直没有踩过刹车，一步步从幼苗长成参天大树。

Linux 1.0: <https://mirrors.edge.kernel.org/pub/linux/kernel/v1.0/>

Linux受GNU通用公共许可证（GPL）保护，其内核源代码是完全开放的。

Most Popular Projects



[Open Hub, the open source network](#)

Rinux 实验背景

开源？

比较容易想到的是各类开源软件。
如何打造一台开源的计算机？你可以…

然而，其计算机内部的处理器是专有的，例如现如今最常见的 **Intel_x86** 处理器和 **ARM** 处理器

- 1978年：Intel-x86架构诞生
- 1985年：ARM架构诞生

授权保密？
知识产权？

Rlinux 实验背景

RISC-V ISA: 一种开源处理器设计，它正在迅速获得关注，并有望改变计算领域。

RISCV发端于深厚的学术研究，将免费且可扩展的软硬件架构自由度提升至新的水平，将为未来**50**年的计算设计与创新铺平道路。



RISC-V 的诞生背景

- 2010年：加州大学伯克利分校的研究团队设计并推出了一套基于BSD协议许可的免费开放指令集架构**RISC-V**
- 2013年：**RISC-V**成功流片。
- 2015年：为推广RISC-V，非盈利性组织RISC-V基金会成立。
- 2018年：RISC-V基金会宣布与Linux基金会进行联合合作。
- 现在….

Rinux a simple re-implementation of unix-like operating system on **RISC-V.**

RISC-V 的未来



FOUNDING

成为资本 CHENGWEI CAPITAL



FOUNDING



FOUNDING



FOUNDING



FOUNDING





Rinix#02

操作系统实验安排说明

Rinix is a Simple Re-Implementation of
Unix-Like Operating System on **RISC-V**.

操作系统实验时间安排

Lab-0 (独立完成)

实验0 Rlinux环境搭建和内核编译 (5%)

秋学期1-3周

- 自己写的代码必须有详细的注释，每5行代码必须有注释，否则会扣分。

Lab-1 (独立完成)

实验1 Rlinux内核引导 (15%)

秋学期4-5周

- 每个实验必须写“讨论心得”（实验过程中遇到的问题及解决方法）这部分内容占本实验报告20%分数

Lab-2 (分组完成)

实验2 Rlinux时钟中断处理 (20%)

秋学期6-7周

操作系统实验时间安排

Lab-3 (分组完成)

实验3 Rlinux 简单的进程调度实现 (20%)

秋学期第8周 至 冬学期第1周

Lab-4 (分组完成)

实验4 Rlinux 虚拟内存管理 (20%)

冬学期2-3周

Lab-5&6 (分组完成)

实验5 Rlinux 动态物理内存管理与页错误处理 (20%)

实验6 Rlinux 用户模式 (**bonus +15%**)

冬学期4-8周

操作系统实验安排

课程成绩评定：

- 期末考试：**50%**
- 作业 (Homework)：**5%**
- 平时课堂练习：**5%**
- 实验报告：**20%**
- 实验验收：**20%**

实验小组分工说明：

实验Lab0-1不分组，需独立完成；

实验Lab2-6可小组分工（2人）实现，实验报告中需注明小组成员的分工情况；

实验内容成绩：实验报告20% + 实验验收20%

操作系统实验验收与提交

实验验收与答疑:

在周一第7,8节, 地点曹西503机房

实验报告提交:

1. 对于每个实验, 每位同学须提交电子文档的实验报告。实验报告(pdf格式)提交的内容包括实验过程记录及源代码。
2. 实验源代码树打包构成一个压缩(*.zip)文件(lab0除外)。

将每个实验的上述两个文件上传到“学在浙大”中
实验报告电子文档文件名: 姓名+实验编号.pdf



Rinix#03

操作系统实验内容介绍

Rinix is a Simple Re-Implementation of
Unix-Like Operating System on **RISC-V**.

实验0 Rlinux环境搭建和内核编译

01

实验目标

使用GDB在QEMU模拟器中调试运行基于64位RISCV架构的Linux内核。

实验简介

02

本次实验将帮助我们更好地了解如何使用交叉编译工具链编译Linux内核源代码，并将编译所得到的基于RISCV指令集架构的Linux内核运行在QEMU模拟器上。

本次实验还将帮助我们更好地了解掌握如何利用GDB和QEMU联合调试内核运行。

实验0 实验环境说明


实验辅助材料


 Linux基本命令 [查看文件](#)

 RISC-V手册 [查看文件](#)

 RISC-V SBI手册 [查看文件](#)

 RISC-V Privileged Spec手册 [查看文件](#)

 oslab文件下载链接

 浙江大学操作系统课程实验 学生版仓库 网址: <https://gitee.com/zjusec/os22fall-stu>

+ 新增学习活动



后续实验的内容
请以发布在“**学在浙大**”课程网站上的实验手册为准

实验内容

 实验0 RV64 环境搭建和内核编译 作业交付截止 2022.10.28 23:59 | 个人作业 (全部学生)

实验0 实验环境说明

实验0 RV64 环境搭建和内核编译

 编辑

作业属性				作业批改		成绩统计	
占成绩比例		0.0%		公布成绩时间		不公布	
开放时间		2022.09.12 12:02		作业交付截止		2022.10.28 23:59	
作业形式		个人作业（全部学生： 94人 ）		完成指标		提交作业	

评分方式 (教师评阅 100.0%)			
<div><div></div>教师评阅</div>			
占成绩比例		100.0%	

实验0：实验验收截止时间2022年10月24日，实验报告上传截止时间2022年10月28日。

实验过程中如遇仓库 <https://gitee.com/zjusec/os21fall>无法访问的情况，
请使用 <https://gitee.com/ilfth/os21fall> 进行代替（注：此仓库仅适用于Lab0）

附件名	大小	
 lab0 Rlinux环境搭建和内核编译v2.2.pdf	245 KB	 

实验0 实验环境说明

03

实验环境

我们的实验建议在Docker下，需要下载oslab.tar这一Docker镜像，下载链接在“**学在浙大**”课程网站上可以得到。

为什么使用Docker?



Os lab.tar是容器镜像，包含了Linux系统的命令程序、QEMU环境、gcc环境、GDB、部分原代码等。

尝试手动构建：<https://www.nuanyun.cloud/?p=1481>

实验0 前置实验阅读材料

Linux相关：

1. [The Missing Semester of Your CS Education >>Video<<](#)
2. [GNU/Linux Command-Line Tools Summary](#)
3. [Basics of UNIX](#)

实验环境与工具相关：

1. [Docker介绍与使用方法：https://www.docker.com/](https://www.docker.com/)
2. [100个gdb小技巧：https://wizardforcel.gitbooks.io/100-gdb-tips/content/](https://wizardforcel.gitbooks.io/100-gdb-tips/content/)
3. [QEMU模拟器的使用：https://www.qemu.org/docs/master/system/index.html](https://www.qemu.org/docs/master/system/index.html)

实验0 前置实验阅读材料

 **gitee**

开源软件 企业版 ^{特惠} 高校版 私有云 博客 **Go** 我的 ▾

搜开源

 ²⁴  + 

导入仓库

 从 URL 导入

 导入 GitHub 仓库

 导入 GitLab 仓库

Git 仓库 URL *

eg: https://github.com/randx/six.git(URL must be clonable)

仓库名称 *

归属

 iLifetruth ▾

/

路径 *

仓库介绍

0/100

用简短的语言来描述一下吧

☐ 开源 (所有人可见) ⓘ

☒ 私有 (仅仓库成员可见)

☐ 企业内部开源 (仅企业成员可见) ⓘ

选择语言

请选择语言 ▾

导入

新建仓库

发布代码片段

创建组织

开通企业版

从 GitHub / GitLab 导入仓库



在Gitee导入Github仓库

实验0 前置实验阅读材料

 iLifetruth / tutorial-unix-basics 

 代码

 Issues 0

 Pull Requests 0

 Wiki

 统计

 流水线

gh-pages ▾

 分支 2  标签 2












+ Pull Request

+ Issue

文件 ▾

Web IDE

克隆/下载 ▾

 Christopher Paciorek	fix typo	b673720	13天前	 101 次提交
 _includes	update headers for site accessibility			2个月前
 _layouts	fix header details			2个月前
 _sass	underline links			14天前
 assets	add ls image			7个月前
 .gitignore	update .gitignore			9个月前
 README.md	full revision			9个月前
 _config.yml	move logo img file			8个月前
 example.txt	setup gh-pages materials and minor revisions to content			9个月前
 index.md	fix typo			13天前

实验0 实验步骤

实验0: Rlinux环境搭建和内核编译

0 实验简介

搭建实验虚拟机、docker运行环境。通过在QEMU上运行Linux来熟悉如何从源代码开始将内核运行在QEMU模拟器上，并且掌握使用gdb协同QEMU进行联合调试，为后续实验打下基础。

1 实验目的

- 了解容器的使用
 - 使用交叉编译工具, 完成Linux内核代码编译
 - 使用QEMU运行内核
 - 熟悉GDB和QEMU联合调试
-

实验0 实验步骤

1. Docker镜像下载

2. Docker安装

3. Docker镜像载入，生成容器

4. 运行容器，开始实验步骤

- 下载Linux内核源代码
- 编译Linux内核源代码
 - 调试Linux内核
 - 完成思考题

实验0 注意事项1

请下载“学在浙大”中所提供的oslab.tar文件，进入Docker环境中开展实验。

oslab文件下载链接

🔖 新页签打开

✎ 编辑

由于网站安全设置，此链接可能无法直接预览，请在右上角点击「新页签打开」查看

开放时间

2022.09.12 12:00

完成指标

访问线上链接



🇨🇳 简体中文

下载APP

登录

张 张文龙 分享了文件

2022-12-24到期



oslab.tar
2.76 GB

预览

下载

保存到浙大云盘



用 浙大云盘 APP扫码，快速保存分享

实验0 注意事项2

按照实验手册进行实验时，请不要复制/粘贴实验手册中的命令行。而是建议在充分理解每一句命令行的含义以及参数内容的前提下，自己在终端手动输入运行。

4.4 使用QEMU运行内核

```
# pwd
/home/os21fall/src/lab0/
# qemu-system-riscv64 -nographic -machine virt -kernel path/to/linux-5.19.8/arch/riscv/boot/Image \
  -device virtio-blk-device,drive=hd0 -append "root=/dev/vda ro console=ttyS0" \
  -bios default -drive file=rootfs.img,format=raw,id=hd0
```

退出QEMU模拟器的方法为：使用 `ctrl + a` (macOS下为 `control+a`)，松开后再按下 `x` 键即可退出qemu

实验0 注意事项3

请大家根据自己的操作系统选择指定的Docker安装方式；

注：Windows下的WSL属于Linux环境，但是并不一定完全适配Docker (Windows Desktop也是基于WSL2)。



实验0 注意事项4

如果在Windows中安装Docker存在问题，建议安装原生Ubuntu 22.04.1虚拟机（如使用VMware Workstation Player）

<https://www.vmware.com/cn/products/workstation-player.html>

- 安装时设置虚拟机ubuntu的内存容量，推荐**4GB**，便于后续进行编译内核的实验；
- 安装时设置虚拟机ubuntu的硬盘容量，推荐不小于**128GB**；
- 设置虚拟机ubuntu的网络共享为**NAT**模式，使得虚拟机可以访问网络；

实验0 注意事项5

若原生Ubuntu 22.04.1虚拟机（如使用VMware Workstation Player）启动后，界面过小，且无法拖放文件。

安装完毕并进入虚拟机后，更新软件源：

```
$ sudo apt-get update && sudo apt-get upgrade -y
```

安装VMWare Tools（高版本的虚拟机环境，已经更名为open-vm-tools）：

```
$ sudo apt-get install open-vm-tools-desktop  
$ sudo apt-get install open-vm-tools
```

安装git客户端（也就是ubuntu环境中git的客户端），方便做实验：

```
$sudo apt install git-all
```

浙江大学实验报告

课程名称： 操作系统

实验项目名称：

学生姓名： 学号：

电子邮件地址：

实验日期： 年 月 日

实验0 实验报告编写

实验0 实验现场验收

一、实验内容

记录实验过程并截图，对每一步的命令以及结果进行必要的解释

三、讨论、心得（20分）

在这里写：实验过程中遇到的问题及解决的方法，你做本实验体会

实验1 Rlinux内核引导

01

实验目标

学习RISC-V相关知识，Makefile相关知识

实验简介

02

编写head.S实现RISC-V boot loader的功能，并利用
Makefile来完成对整个工程的管理

实验2 Rlinux时钟中断处理

01

实验目标

能在BootLoader中启用时钟中断，并定时打印内容

实验简介

02

学习在RISC-V上的异常处理相关机制
以时钟中断为例，编写时钟中断处理函数

实验3 Rlinux 简单的进程调度实现

01

实验目标

基于时钟中断处理，实现内核进行的调度与切换

实验简介

02

结合课堂所学习的相关内容，在上一实验实现中断的基础上进一步地实现简单的进程调度

- 非抢占式短作业优先算法
- 抢占式优先级算法

实验4 Rlinux 虚拟内存管理

01

实验目标

学习RISCV虚拟内存管理机制

实验简介

02

结合课堂学习的页式内存管理以及虚拟内存的相关知识，尝试在已有的程序上开启 MMU 并实现页映射，保证之前的进程调度能在虚拟内存下正常运行

实验5 Rlinux 动态物理内存管理与页错误处理

01

实验目标

- 目标一：了解 **Buddy System** 和 **Slub Allocator** 物理内存管理机制的实现原理。实现统一的内存分配/释放接口：**kmalloc/kfree**。
- 目标二：在 **mm_struct** 中补充实现 **vm_area_struct** 数据结构的支持，并结合后续 **mmap** 等系统调用实现对进程多区域虚拟内存的管理。
- 目标三：实现 **222: mmap**, **215: munmap**, **226: mprotect** 系统调用。
- 目标四：在 **Lab5** 实现用户态程序的基础上，添加缺页异常处理 **Page Fault Handler**，并在分配物理页或物理内存区域的同时更新页表。
- 目标五：综合上述实现，为进程加入 **fork** 机制，能够支持创建新的用户态进程，并测试运行。

实验简介

在充分理解前序实验中RISC-V地址空间映射机制与任务调度管理的前提下，进一步了解与**动态**内存管理相关的重要结构，实现基本的内存管理算法，并最终能够较为**综合**地实现对进程的内存管理与调度管理

02

实验6 Rlinux 用户模式 (Bonus)

01

实验目标

- 实现用户态进程
- 完成write、getpid系统调用
- 了解文件系统的实现，学会使用给定的文件系统读取文件内容。
- 理解ELF文件格式，实现简单的ELF Loader
- 实现exec, wait, exit系统调用。
- 使用给定的Shell，能够解析命令行输入，获取需要执行的用户态程序名，运行给定的用户态程序。

实验简介

02

- 在低地址空间映射用户态程序，并尝试编写简单的系统调用处理函数。
- 利用给定的文件系统和Shell程序，实现从命令行运行给定的用户程序。

The background is a solid dark blue. On the right side, there are several large, overlapping, light blue geometric shapes that resemble stylized mountain peaks or abstract architectural elements. These shapes are composed of straight lines and smooth curves, creating a dynamic and modern look.

THANKS

For Your Attention

Q&A