

嵌入式工作室23级暑期招新-computer architecture

计算机系统结构的定义及主要研究内容

定义：计算机系统结构，也称计算机体系结构

- 经典定义：计算机系统结构是程序员所看到的计算机属性，即概念性结构与功能特性
- 广义定义：计算机系统结构囊括计算机设计的3个方面：指令集结构、组成、硬件

摘自别人的一篇博客

万丈高楼平地起，计算机系统就像程序员金字塔的地基。理解了计算机系统的构造原理，在写程序的道路上才能越走越远。道理LZ很早就懂了，可是一直没下定决心好好钻研，或许是觉得日常工作中根本用不到这些，又或许是每次拿起书看到那些复杂的底层架构，看到存储器，寄存器，CPU，总线等等这些概念就头大。总之，由于各种各样的原因，对这块的知识一直没有认真花时间去钻研。那么你可能会问，那你写这篇博客的题目不就是准备学习这方面的知识吗？是的，LZ准备下定决心钻研了，至于原因如下：

①、经常用一些不知其所以然的技术，会感到不安

大家可以看看LZ前面写的博客，很多都是对框架的用法进行总结，至于为什么这个框架要这么实现等等而涉及的很少。原因很简单，LZ研究框架的源码很吃力。因为源码的实现往往涉及到多方面的知识，比如设计模式，比如JDK的一些高级特性等等。LZ是想做一个有梦想的咸鱼，但是每次看到很多不懂的东西，就深深的体会到现实与理想落差的痛苦，所以LZ下定决心要改变这种无力感。

②、在这个数据爆炸的年代，很多系统对于项目性能的优化有很高的要求。那么如何调优成为了程序员不可避免的问题，我们知道Java是运行在JVM上的，程序的调优那就需要对JVM有一定的了解。LZ本来想直接研究JVM的，书都找好了-----《深入理解Java虚拟机：JVM高级特性与最佳实践》。但是LZ发现很难消化，理解起来特别困难，于是间接的将LZ推上了研究计算机系统的道路。

③、再就是如果你对计算机系统的构造不太了解，可能你平时写的程序都是错误的。

比如我们一直以为两个正数的和或者积一定为正，但是用二进制补码表示的正数和或者积却不一定；程序员和编译器不能用 $(x-y < 0)$ 来代替 $(x < y)$ ，因为前者会产生溢出。甚至也不能用表达式 $(-y < -x)$ 来代替，因为在二进制补码表示中负数和正数的范围是不一致的。算术溢出是造成程序错误和安全漏洞的一个常见根源。

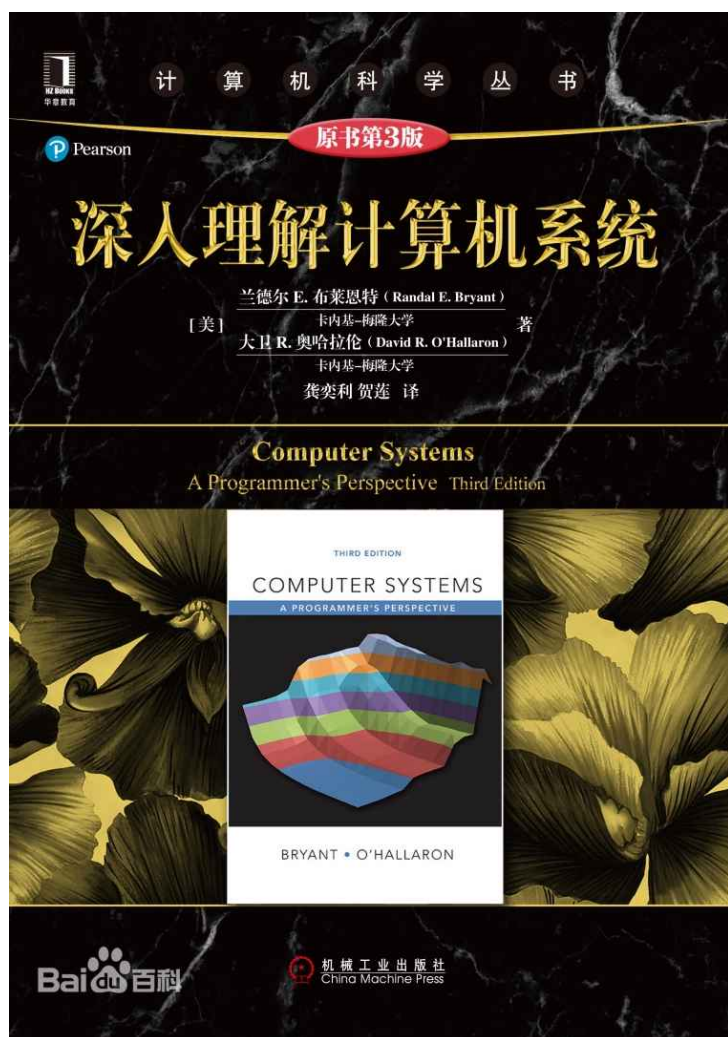
④、第四个原因，这个也是最直接的原因。不知道大家是否了解大的互联网公司面试，比如阿里巴巴，腾讯这样的公司。他们面试对于基础的东西要求的很严格，你对这些东西的了解程度将直接决定你的面试成败。

比如：讲一下JVM的结构，TCP/IP的三次握手、四次挥手，淘宝用户的数据怎么满足高并发？等等。像这些问题，如果我们不懂计算机底层知识肯定是不能过关的。

当你执行一个简单的hello.c代码时会不会好奇——一段代码是如何变成可执行程序被执行的，终端的命令又是如何被执行的呢，为什么多个应用程序可以同时执行互不干扰，需要的内存空间大于电脑实际物理内存时程序为什么仍旧可以运行……所有这些都属于计算机的体系结构

- 体系结构为工作室新开设的招新方向，欢迎对计算机体系结构有浓厚兴趣的你来挑战，学习，在编写代码时如果你曾好奇编译的过程，不如尝试自己去学习，寻找答案，然后在寻找答案的过程中你又会遇到一些新的疑问，然后继续寻找答案，最终一个个学习的知识会编织成一张网（这里不得不提到一本书）

本书主要介绍了计算机系统的基本概念，包括最底层的内存中的数据表示、流水线指令的构成、[虚拟存储器](#)、编译系统、动态加载库，以及用户应用等。书中提供了大量实际操作，可以帮助读者更好地理解程序执行的方式，改进程序的执行效率。此书以程序员的视角全面讲



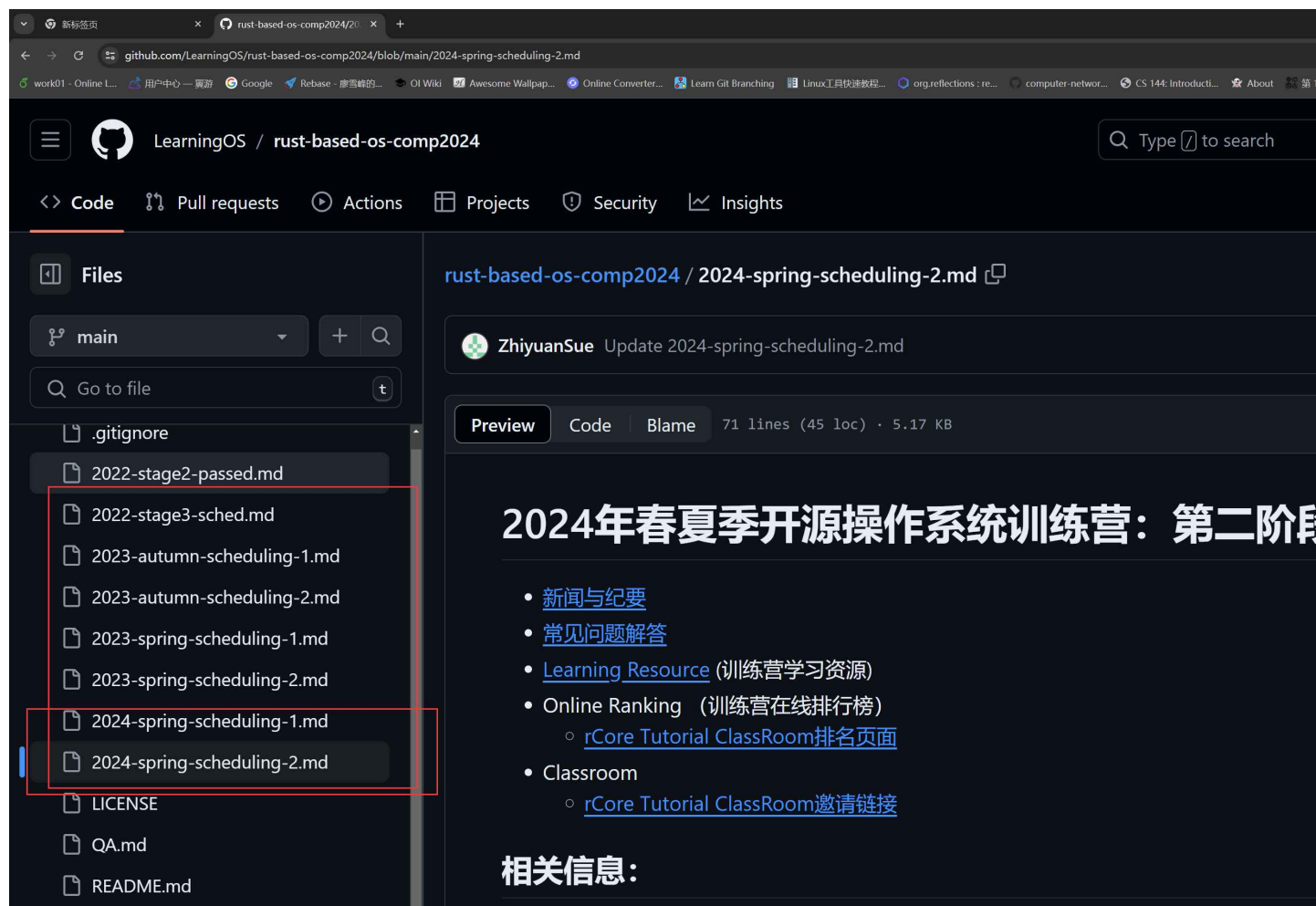
解了计算机系统，深入浅出地介绍了处理器、编译器、操作系统和网络环境，是这一领域的权威之作。（来自百度）

-  [计算机科学丛书] Randal E.Bryant_David O'Hallaron - 深入理解计算机系统（原书第3版）(2016, 机械工业出版社) - libgen.lc.pdf



- 本次招新主要为两个阶段
 - 学习Rust语言
 - 使用Rust语言完成操作系统相关实验
- 题目来自于2024年春季清华大学、CSDN、阿图教育等共同举办的**2024年春季开源操作系统训练营活动**
- 题目链接
<https://github.com/LearningOS/rust-based-os-comp2024/blob/main/2024-spring-scheduling-1.md>
- 你们需要完成的是2024-spring-scheduling-1.md & 2024-spring-scheduling-2.md

即左边红框中的部分



初级阶段

学习Rust语言（[2024-spring-scheduling-1.md](#)）,github上提供了Rust语言学习以及操作系统实验的开发和测试环境，可以进行题目查看、代码测试以及答案阅读

1. 点击红框区域配置环境



2. 详细配置步骤

Files

main

Go to file

.gitignore

2022-stage2-passed.md

2022-stage3-sched.md

2023-autumn-scheduling-1.md

2023-autumn-scheduling-2.md

2023-spring-scheduling-1.md

2023-spring-scheduling-2.md

2024-spring-scheduling-1.md

2024-spring-scheduling-2.md

LICENSE

QA.md

README.md

announce.md

excellent-students-2022summer....

excellent-students-2023winter.md

graduate-students-2023winter.md

lab3-os5-passed.md

news.md

relatedinfo.md

rust-based-os-comp2024 / 2024-spring-scheduling-1.md

Preview Code Blame 119 lines (69 loc) · 10.6 KB

Raw

codespace 不是必须的。如果是本地的ubuntu中建立开发环境，可在shell中执行 `make ubuntu_local_setenv` 来自动安装配置开发环境（执行需要 `sudo` root 权限，仅需要执行一次）。

a. 在网络浏览器中用自己的 github id 登录 github.com。

b. 接收 [Rust-lang Lab Test based on Rustlings](#) 的github classroom[在线邀请](#)，根据提示一路选择OK即可。

c. 完成第二步后，你的rustlings实验练习的 github repository 会被自动建立好，点击此github repository 的链接，就可看到你完成的实验了。

d. 在你的第一个实验练习的网页的中上部可以看到一个醒目的 `code` 绿色按钮，点击后，可以进一步看到 `codespace` 标签和醒目的 `create codesapce on main` 绿色按钮。请点击这个绿色按钮，就可以进入到在线的ubuntu +vscode环境中

e. 再按照下面的环境安装提示在vscode的 `console` 中安装配置开发环境：rustc等工具。

f. 然后就可以基于在线vscode进行测试（执行命令 `rustlings watch` ），编辑代码的循环实验过程了。

g. 如果使用本地的环境进行rustlings的练习，请按照接下来的步骤进行：首先需要安装一个Linux的环境，对于windows的用户，推荐使用wsl2，也可以使用vmware等虚拟机进行安装。如果在这一步存在问题，请联系助教。

h. 创建ssh key。在linux环境下，使用 `ssh-keygen -t rsa -b 4096 -C "你的邮箱"` 命令，创建ssh key，下面的选项全部直接敲回车即可。随后使用 `cat ~/.ssh/id_rsa.pub` 命令查看生成的公钥，并完整的复制下来。在github仓库界面点击自己的头像，选择 `settings`，进入到设置页面后，点击左侧的 `SSH and GPG keys` 选项。点击 `New SSH key` 选项，并将复制下来的内容粘贴上去，添加该ssh key的描述。随后点击 `Add SSH key`，并一路点击确认即可。

i. 在本地安装rust。进入linux环境下，参考rcore 教程 <http://rcore-os.cn/rCore-Tutorial-Book-v3/> 中，第零章操作系统概述，实验环境配置的内容，找到Rust 开发环境配置的章节，相应配置即可，你可以同时将后续需要的环境也配置好。

j. clone实验仓库到本地。在前面点击链接生成的仓库中，同样点击醒目的 `code` 绿色按钮，选择 `local` 下的 `ssh` 选项，复制下面的链接。随后回到本地linux环境下，使用 `git clone` 复制的链接的方式，将目标仓库clone到本地。随后，使用 `ls` 命令查看自己clone下来的文件夹，再使用 `cd` 命令进入到该文件夹下，使用 `cargo install --force --path .` 安装rustlings。

k. 练习rustlings。使用vscode等编辑器，进入clone下来的目录下的 `exercises` 文件夹，依次完成对应的练习。随后在rustlings的根目录下使用 `rustlings run` 练习名称 去运行对应练习，也可以使用 `rustlings hint` 练习名称 查看题解。

l. 提交。当做完部分或所有练习之后，执行 `git add; git commit -m "update"; git push` 命令，把更新提交到GithubClassroom的CI进行自动评测。你可以在github仓库页面的actions页面，看到你的CI提交结果，或者在排行榜上面查看自己的评分。

3. 进行本地Rust语言实验学习和测试

🌟

初级阶段任务要求：

● 完成所有Rust语言测试题

● 使用markdown撰写的能够展现学习过程的笔记

高级阶段

通过初级阶段的学习，相信你已经初步掌握了Rust语言，接下来，我们希望你能使用Rust语言完成第二阶段的操作系统实验（[2024-spring-scheduling-2.md](#)）

操作系统相关实验与前面类似，可以通过配置本地测试环境

1. 配置实验环境

Files

main

Go to file

.gitignore

2022-stage2-passed.md

2022-stage3-sched.md

2023-autumn-scheduling-1.md

2023-autumn-scheduling-2.md

2023-spring-scheduling-1.md

2023-spring-scheduling-2.md

2024-spring-scheduling-1.md

2024-spring-scheduling-2.md

LICENSE

QA.md

README.md

announce.md

excellent-students-2022summer....

excellent-students-2023winter.md

graduate-students-2023winter.md

lab3-os5-passed.md

news.md

relatedinfo.md

rust-based-os-comp2024 / 2024-spring-scheduling-2.md

PreviewCodeBlame71 lines (45 loc) · 5.17 KB

RawDownloadEdit

- 课程幻灯片
- 参考书
 - Operating Systems: Three Easy Pieces
 - 深入了解计算机系统
 - RISC-V Reader中文版

课程实践：rCore Tutorial Book v3

- 课程实践参考书
- 课程实践代码仓库
- 课程实践代码的API文档

基于Rust语言的rCore Tutorial实验指导

- 实验文档
- API文档
- 实验代码
- 测试用例

rCore实验讲解视频

建立基于ClassRoom实验的具体步骤

基于Rust语言的rCore Tutorial

- rCore Tutorial ClassRoom邀请链接：点击后按提示可以建立自己的rCore Tutorial实验专用仓库
- rCore Tutorial ClassRoom排名页面：可以查看自己的rCore Tutorial实验的排名情况

请根据各个实验的具体实验要求在自己的仓库中完成5个实验。请在每完成一个实验（完成编码和实验报告文档）后，请通过执行 `git push` 命令来更新自己的实验专用 repos，来通过基于GitHub Classroom的CI测试。

One More Thing：当你看到这，感觉第二阶段还没开始，还在想下一步要干啥时，我们的建议是：Just Do It NOW!

2. 进行本地操作系统实验学习和测试



高级阶段任务要求：

- 完成所有操作系统相关实验测试题
- 使用markdown撰写的能够展现学习过程的笔记

项目提交

提交内容



我们期待的提交内容：一份用 **markdown** 撰写的、**可读性强**的能够展现你完成招新项目的**完整过程**的笔记以及一个代码结构清晰、附带详细的README文件GitHub仓库

- 学习笔记.pdf**:项目完成过程的学习笔记（例如了解到的操作系统相关知识）、遇到的困难与解决方案、你的所思所想等，文件命名：学习笔记（必选）
- github.txt**:项目 github 仓库链接（必选）
- 其他能展示你能力的补充材料（可选）

提交方式



嵌入式工作室期待遇见一个有学习热情、自驱力强、有规划的你，希望你能合理安排招新题完成进度~

请在规定截止时间内将项目打包发送至 1494627110@qq.com，邮件标题为：**嵌入式工作室招新-学号-姓名-体系结构**，打包命名：**学号-姓名-体系结构.zip**

第一次提交 DDL：**2024-08-01 23:59:59**

做多少交多少：无论你完成进度，我们都希望你能够提交，这样有助于出题人更好了解大家的完成进度，适当对招新题作出一定调整~

第二次提交 DDL：**2024-09-01 23:59:59**

提示

学习过程中遇到问题

1. [2024-spring-scheduling-1.md](#)和 [2024-spring-scheduling-2.md](#)中有许多学习资料可以供大家学习参考
2. 除了提及的文档外，大家还可以积极探索我们所提供的github仓库里的其他内容
3. 灵活使用搜索引擎。
4. 学会去外网找资料。
5. 多参考别人的开源项目。

注：目前该训练营已经结束，但是评测系统仍然可用，只是排行榜可能不再有效，如果发现评测系统也失效，可以及时反馈给出题人

若有其他问题也可私戳出题人  王茜 或在飞书文档（[📖 嵌入式工作室23级暑期招新-computer architecture](#)）进行评论



暑期招新只是将你引入门，我们希望通过此次的招新学习可以激发你们对于计算机体系结构的兴趣，期待与有想法的你碰撞出思维的火花！