基于RISC-V的操作系统实验集成与改进

毕业设计总结：收获与反思

1. 我学习到的新技能（收获）
2. Linux发行版Ubuntu的VMware虚拟机安装及进一步使用，之前用的是Manjaro的VBox虚拟机安装，实践体验是Manjaro不好用，还是Ubuntu经典。不过之前只掌握一些简单的命令行使用，甚至连输入法、VScode、Python3.0、字体界面调整工具gnome等都没安装。
3. QEMU硬件模拟器的安装及使用，不仅仅能模拟RISC-V硬件平台，还支持许多其他架构如X86、ARM、MIPS等。
4. 交叉编译工具链的安装及使用，一方面也在上《编译原理》的课，理解了交叉编译的缘故。
5. 关于OpenSBI、RustSBI、SBI的初步认识和使用（预编译二进制版本）。
6. VScode中代码编写及调试，之前只用过一点VisualStudio写C++和X86汇编调试。
7. Rust代码的阅读和理解，简单了解其优势如安全性高（内存溢出、错误和空值的处理等）、清晰简洁（宏定义机制和基于 Traits 的通用范型系统）、完备易用的Cargo工具链等。
8. Makefile语法及编写、修改，主要由目标文件、先决文件、执行命令构成。
9. 链接脚本（.ld）语法及编写、修改，主要用于输入输出文件之间的映射和内存布局，理解为主。
10. 阅读RISC-V指令集手册，了解了RISC-V指令集的构成、特点、指令格式，会编写和理解一些简单的RISC-V汇编指令。
11. 大致了解和学习[rCore](https://rcore-os.github.io/rCore-Tutorial-Book-v3/index.html)、[uCore](https://nankai.gitbook.io/ucore-os-on-risc-v64/)、xv6（CSDN博客[中文实验跟做](https://blog.csdn.net/u013577996/article/details/108679997)、[英文原课程](https://pdos.csail.mit.edu/6.828/2019/index.html)）操作系统实验课程。
12. 五个实验的技能收获：内核启动（程序入口点、栈顶栈底及栈空间大小），简单批处理（应用程序设计、系统调用、陷入处理），分时多任务（关于时间的CSR调用和任务切换、上下文保存与恢复），内存管理——RISC-V架构的SV39多级页表硬件分页机制及代码实现（物理地址虚拟地址管理）、简单进程管理（创建进程和回收进程资源、用户shell实现）。
13. 利用Git上传Github代码和文档，分支管理，一些简单的Git命令，之前一直用最原始的方式上传文件到Github、一直只是下载从未上传。
14. 学会了Markdown格式文件的书写（工具是Typora+Picgo+Gitee图床），优点是快捷键、格式简洁优美、支持Github直接阅读。
15. 学会了Gitbook在线书写和发布。
16. 新工具：石墨文档（支持协同），ProcessOn思维导图（支持协同）。
17. 学会了VPN登录、搭梯子。
18. 虚拟机打包。
19. 了解了开源相关概念：开源软件、开源协议、开源组织、开源社区、开源操作系统、开源操作系统大会。
20. 一些新概念新词：SiFive、RISC-V基金会、香山处理器、鹏城实验室、欧拉操作系统、龙蜥开源社区。
21. 我可能帮助到的人群
22. 想学习rCore但无Rust语言基础的同学，用C翻译Rust代码，可作为过渡实验。
23. 对操作系统实验感兴趣的同学。
24. 复盘：一些感悟

频繁使用CSDN、Github、Gitee、cnblogs竟然是从上二学位开始的，让我看到了更开阔开源的世界：计算机软件开发（前端、后端）、硬件设计、各种编程语言（C、C++、Java、SQL、x86汇编、MIPS汇编、Verilog等），芯片设计、流片、封装测试，FPGA，CPU、处理器、超流水线、超标量流水线。之前的专业（光电信息科学与工程-光纤方向），应用光学、物理光学、非线性光学、晶体光学、激光原理、光纤通信原理，更多是公式推导（麦克斯韦方程、耦合方程）和理论（如成像、波导、薄膜等）的理解，用到软件设计的并不多（印象中有proteus、multisim、optisystem、zemax、filtmaster），不怎么涉及到太多编程代码。

操作系统、计算机体系结构、计算机组成原理、计算机网络、编译原理，是计算机专业的几门硬课，实验能帮助加深印象和深入理解，用起来的东西才不容易遗忘。

操作系统实验好难，内核栈和用户栈切换时的任务上下文、陷入上下文恢复与保存不好理解，尤其是跳板机制。做的过程难，做完了觉得实现的功能好简单。MIT的S081.xv6实验看起来不错，先挖个坑。

RISC-V和ARM架构在未来有可能替代X86架构成为主流计算机指令集架构、主流CPU架构、主流芯片设计架构，开源、精简、高效、扩展性移植性强是其优势。目前RISC-V架构的应用主要是嵌入式处理器，但嵌入式芯片的应用也很广泛。