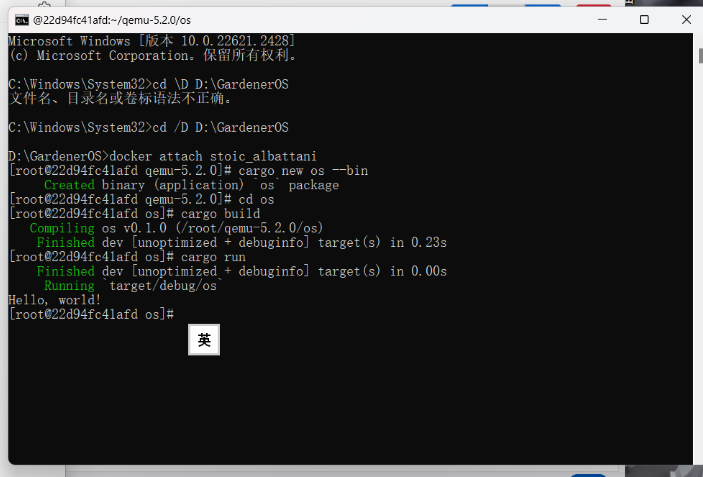
2023秋《操作系统》课程实验报告

实验一

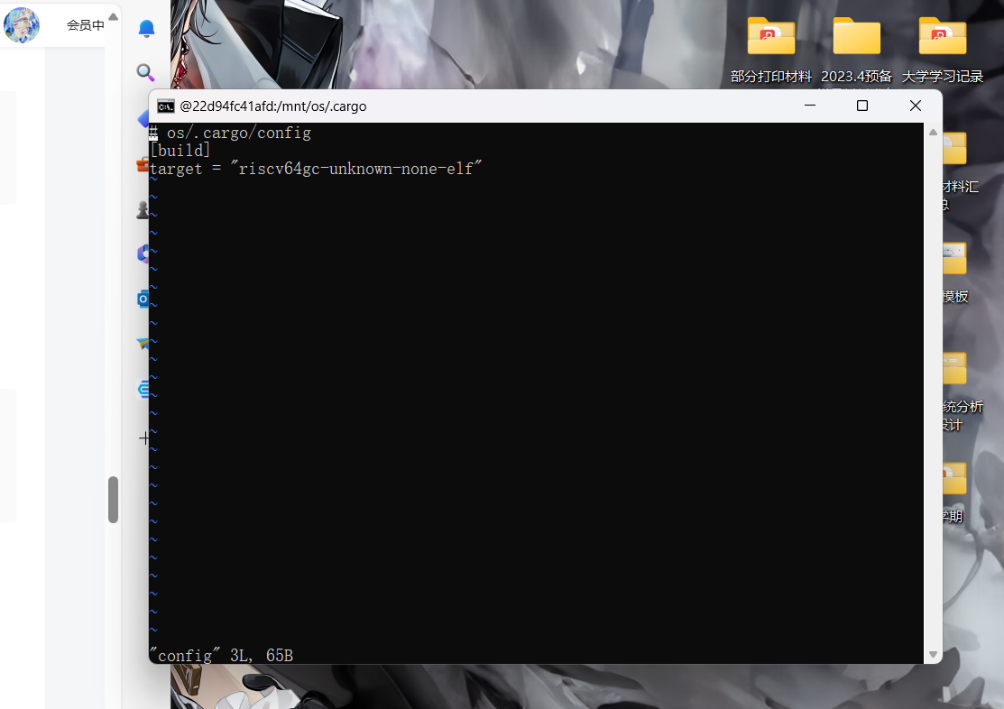
21301161 齐家昕

1. 实验步骤
2. 创建一个Rust项目，执行如下命令查看应用运行结果

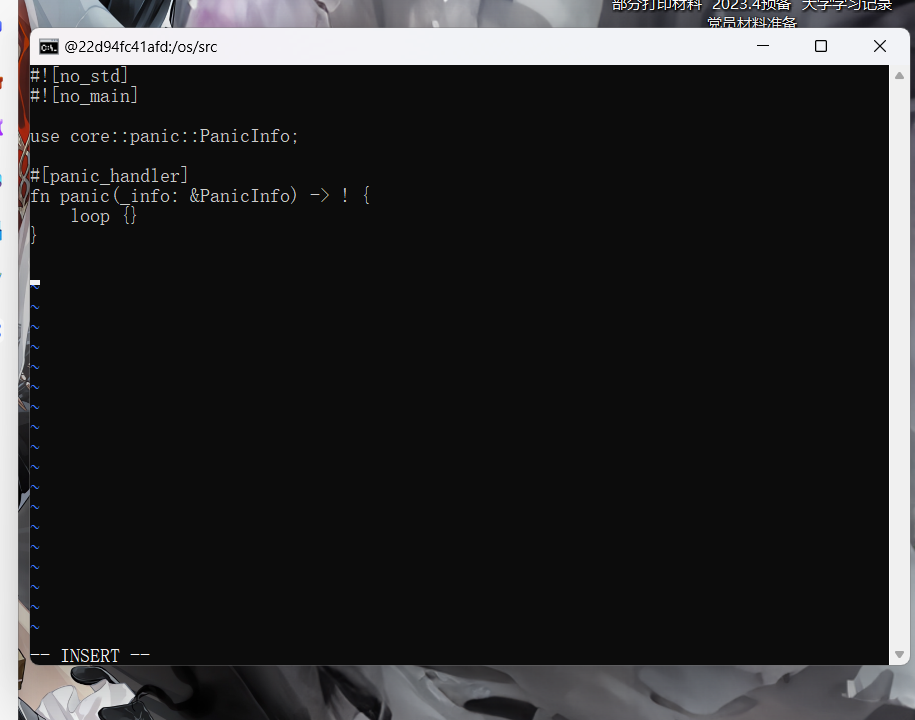


1. 移除标准库依赖

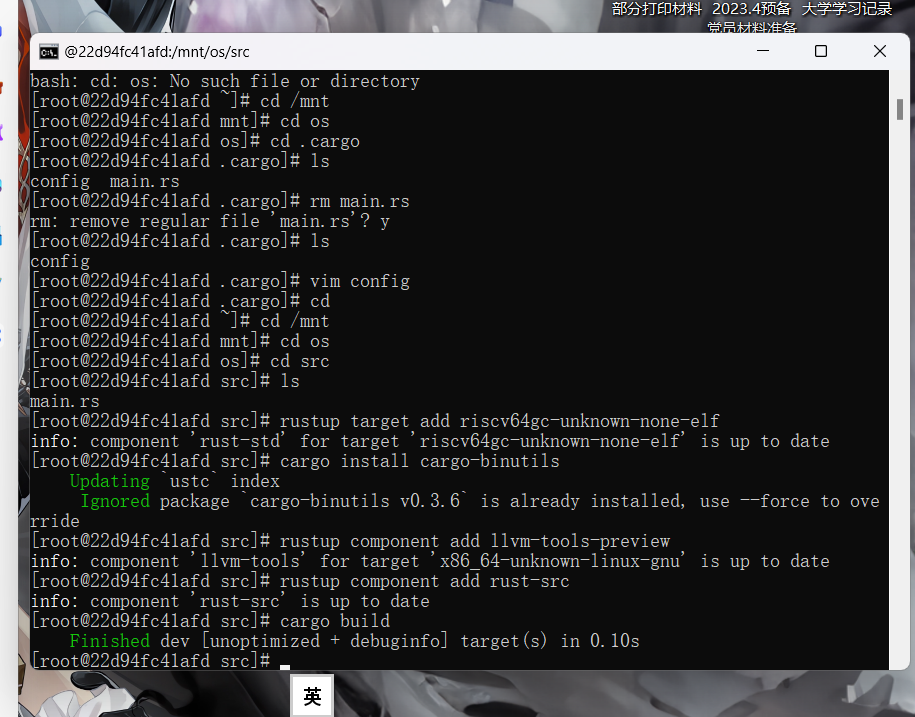
需要修改target为riscv64



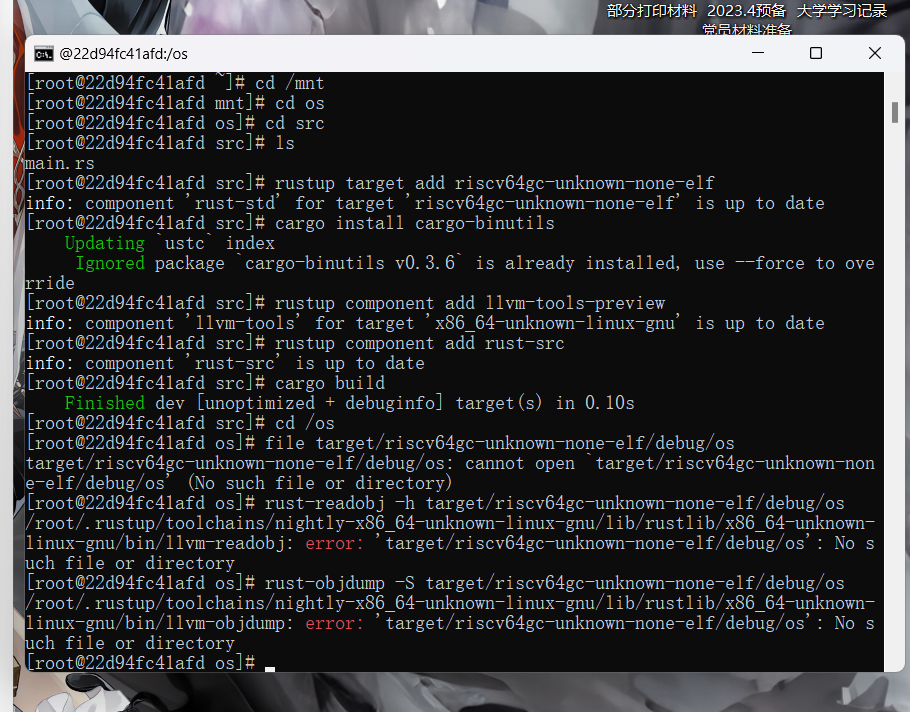
修改main.rs文件



执行cargo build进行编译

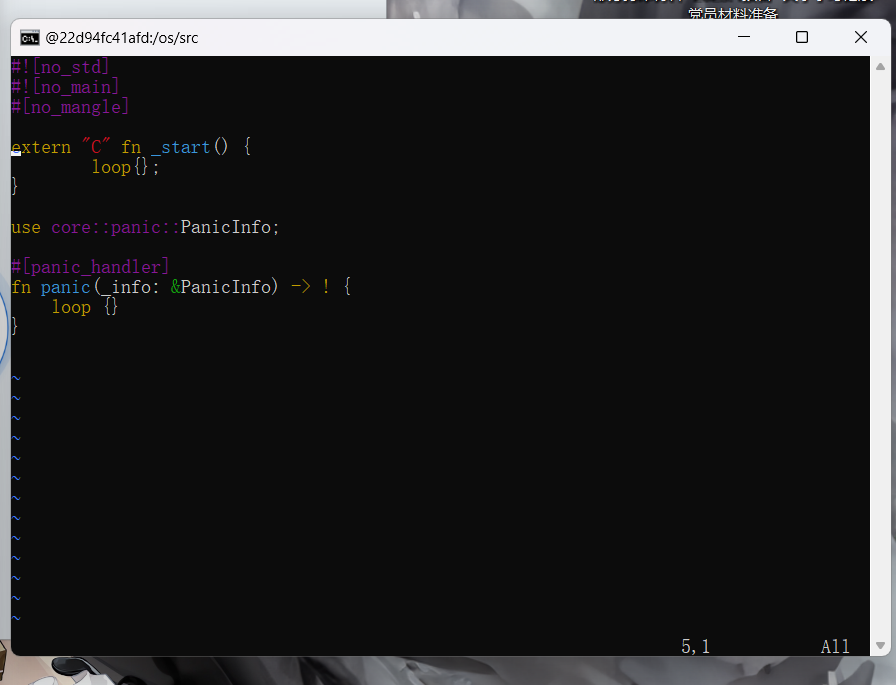


分析独立的可执行程序

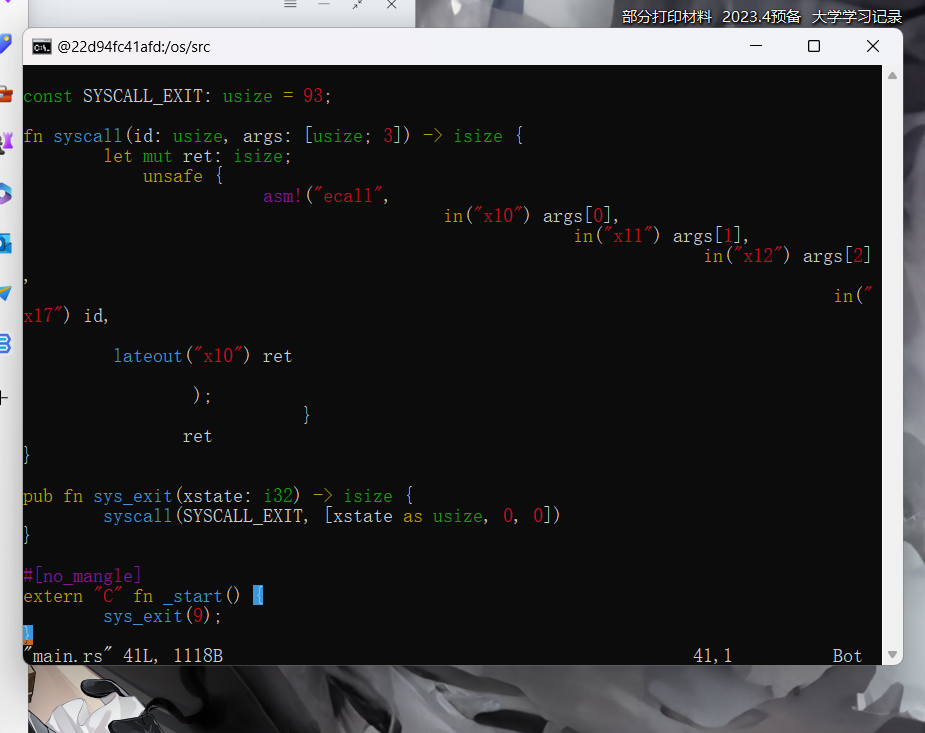


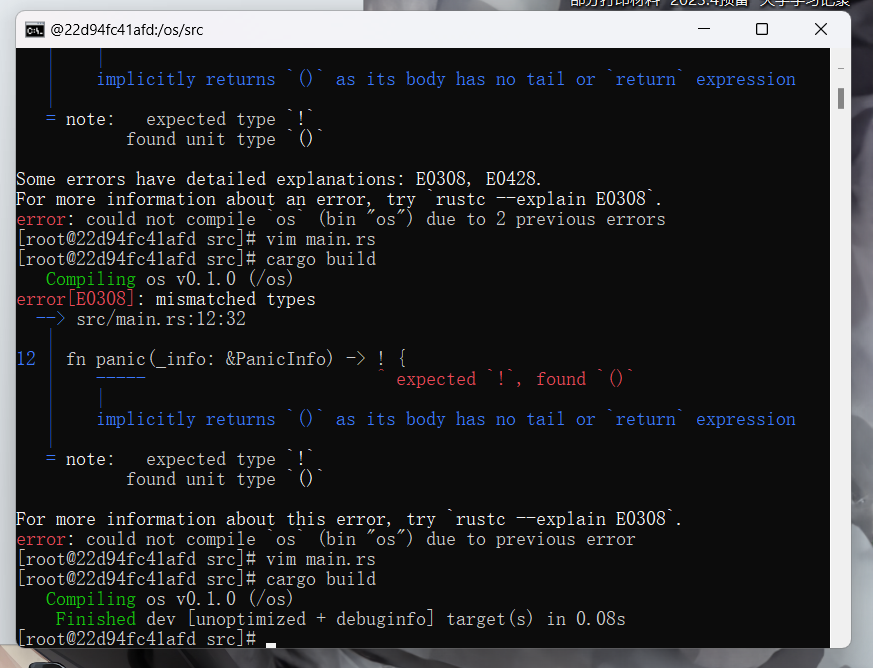
3.用户可执行的环境

增加入口函数

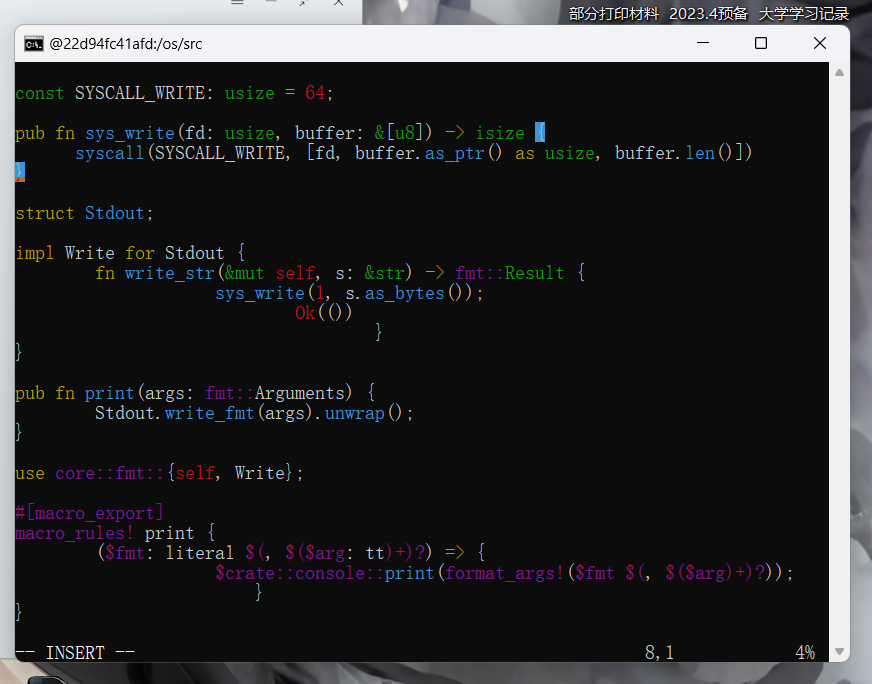


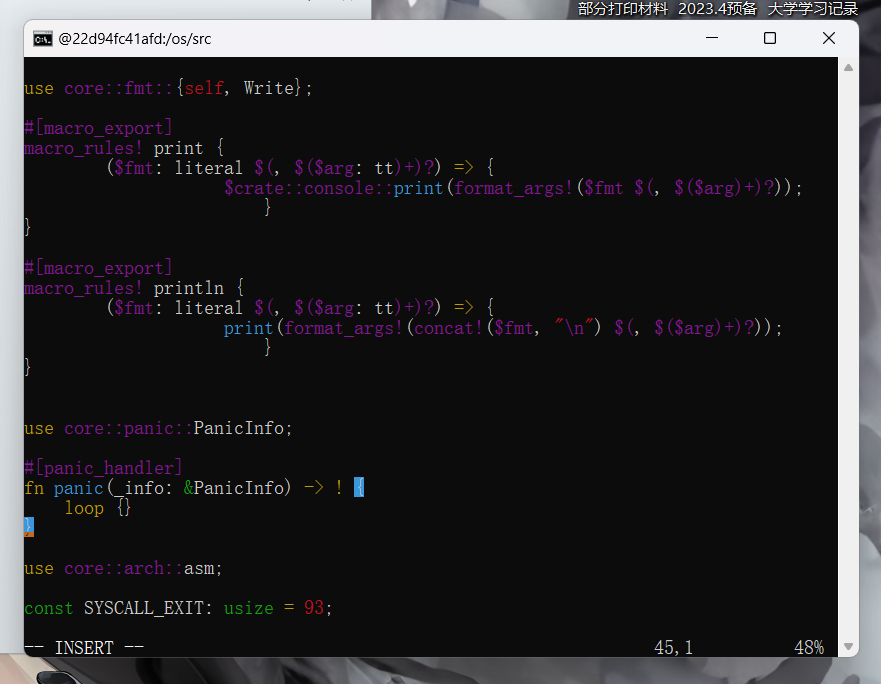
实现退出机制





实现输出支持



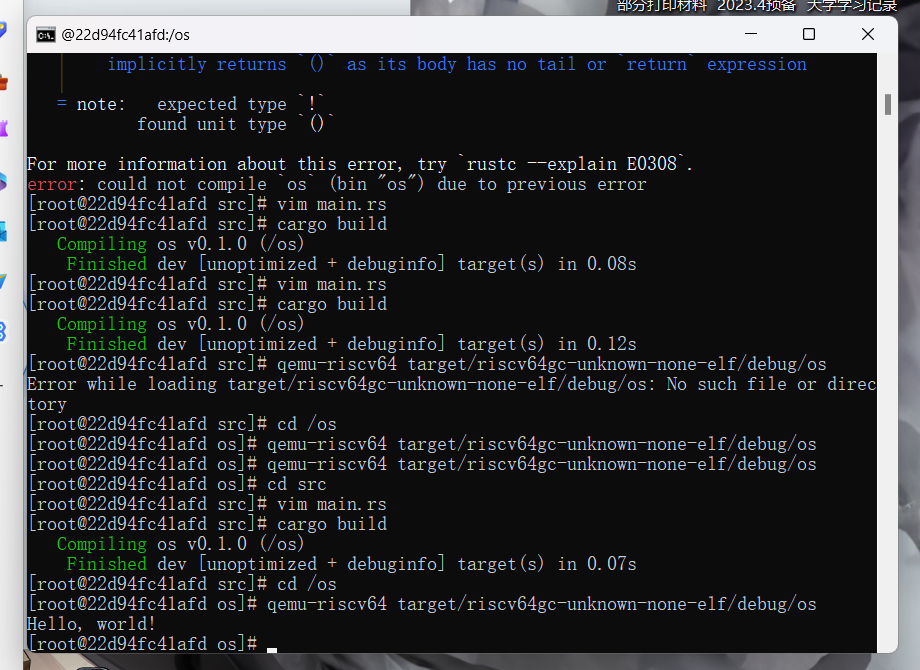


同时在入口函数\_start增加println输出



编译并通过如下命令执行，就可以看到独立的可执行程序已经支持输出显示了。

qemu-riscv64 target/riscv64gc-unknown-none-elf/debug/os



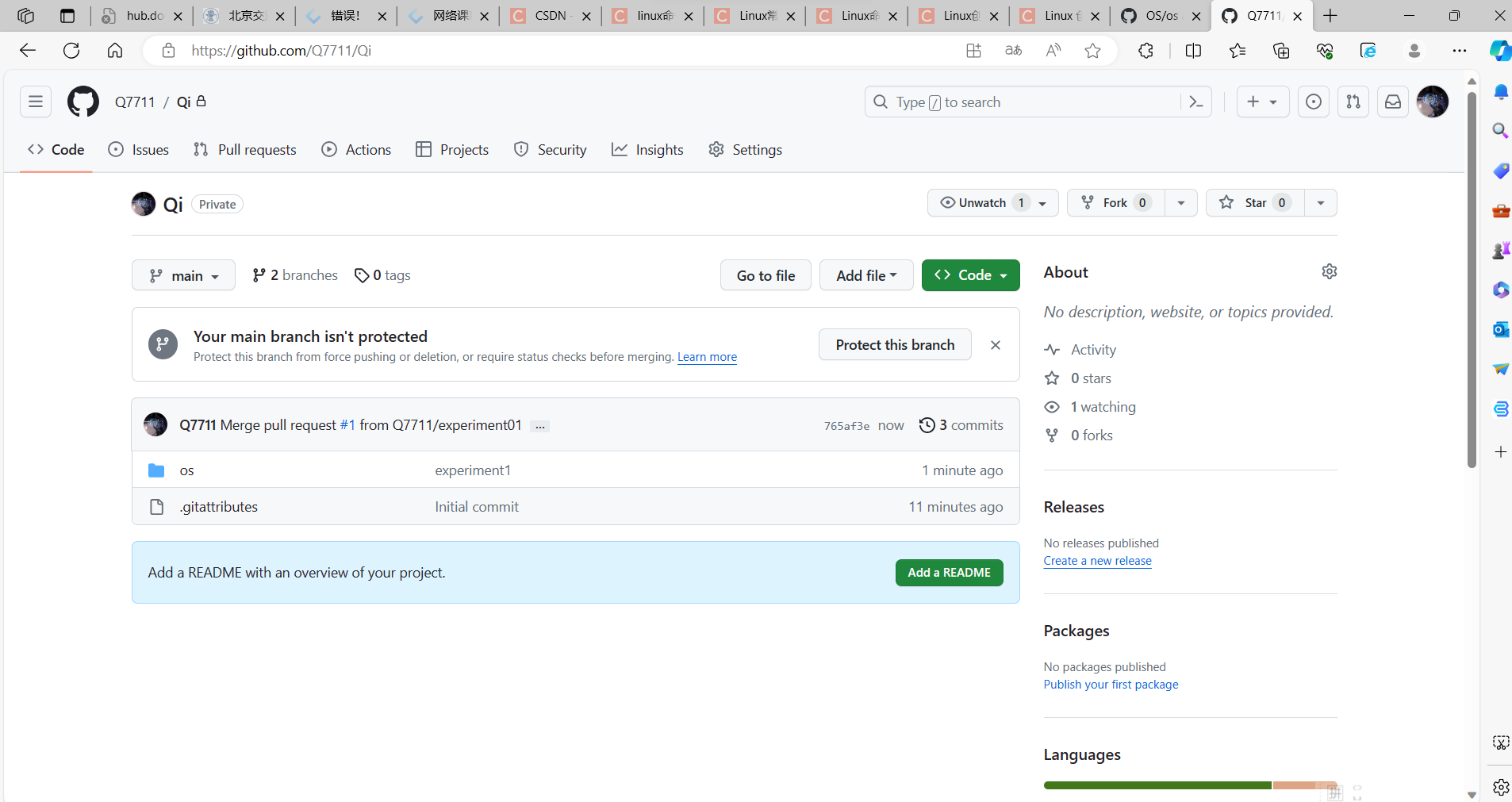
1. 思考问题
2. 为什么称最后实现的程序为独立的可执行程序，它和标准的程序有什么区别？

最后可执行的程序不依赖于rust标准库，是独立的可执行程序，需要进行手动的底层编程。在编译时通过使用 #![no\_std] 属性，告诉编译器不链接标准库。

1. 实现和编译独立可执行程序的目的是什么？

为了在资源受限或特定环境下提供更高的灵活性和控制性

1. Git提交截图



1. 其他说明

无