实验三

姓名:	张云策	学号:	3200105787	学院:	计算机科学与技术学院
课程名称:	计算机系	系统 II	_同组学生姓名:		/
实验时间:	2021.	实验地点:	紫金港机房	指导老师	j: 申文博

一、 实验目的和要求

- 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

二、实验内容和原理

2.1 实验内容

GDB + QEMU 调试 64 位 RISC-V LINUX

2.2 设计模块

三、 主要仪器设备

Docker 实验环境镜像

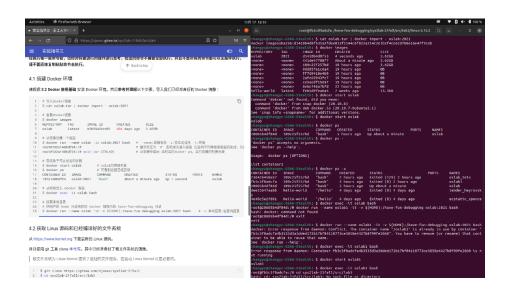
四、操作方法与实验步骤

- 4.1 操作方法
- 4.2 实验步骤
- 4.1 搭建 Docker 环境
- 4.2 获取 Linux 源码和已经编译好的文件系统
- 4.3 编译 Linux 内核
- 4.4 使用 QEMU 运行内核
- 4.5 使用 GDB 对内核进行调试

五、 实验结果与分析

4.1

导入镜像,查看镜像,创建一个名为"oslab"的 container,并且启动,从终端连接,重新将本地 home 目录映射到新建"Oslab1"容器中



(由于之前已经完成过实验,所以 docker images 会显示出现多个镜像文件, 并且再创建时会显示重名,并未利用 docker rm 命令进行删除重建)

4.2

下载 Linux 5.15.2 源码,并且利用 git clone 仓库内容,最终显示出在 src/lab3 中为三个文件: rootfs.img/Linux.tar/Linux

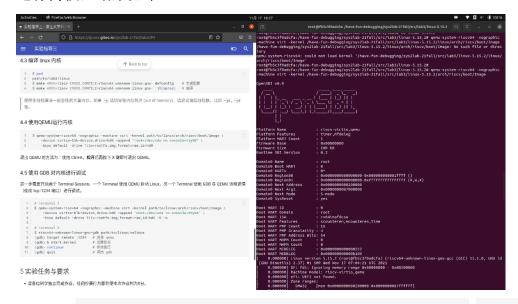
4.3

编译内核,起初出现 out of memory (出于对 PC 性能的自信,结果翻车),



4.4

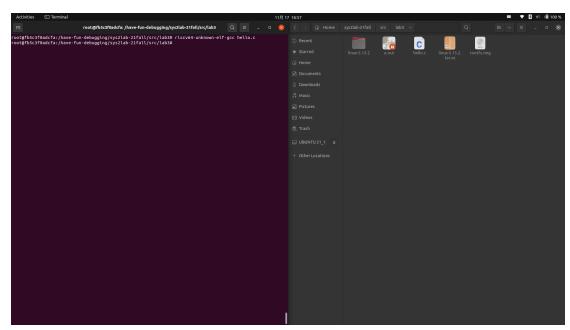
运行内核,结果如下:



并 对 qemu-system-riscv6 -device virtio-blk-device,drive=hd0 -append "root=/dev/vda ro console=ttyS0" 以及 qemu-system-riscv64 -bios default -drive file=rootfs.img,format=raw,id=hd0 等命令进行运行,进入 QEMU 后退出。

六、 讨论、心得

1/2 编译 hello.c 文件,并对编译产物进行反汇编:



```
| Temporal | Temporal
```

- 3. a.使用 layout asm 命令
 - b.使用 b*0x80000000
 - c. info break
 - d.使用 b * 0x80200000
 - e.clear 1
 - f. continue
 - g.n
 - h.quit
- 4. 利用 make clean 命令, 进行清除
- 5. Vmlinux 为一个包含着 Kernel 的 ELF 或 a.out 文件,是最原始的 kernel 文件,而 Image 则是一个在 vmlinux 基础上删除调试信息以及符号表的二进制文件,比如 QEMU 中即为 Image