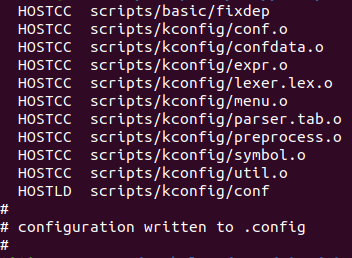
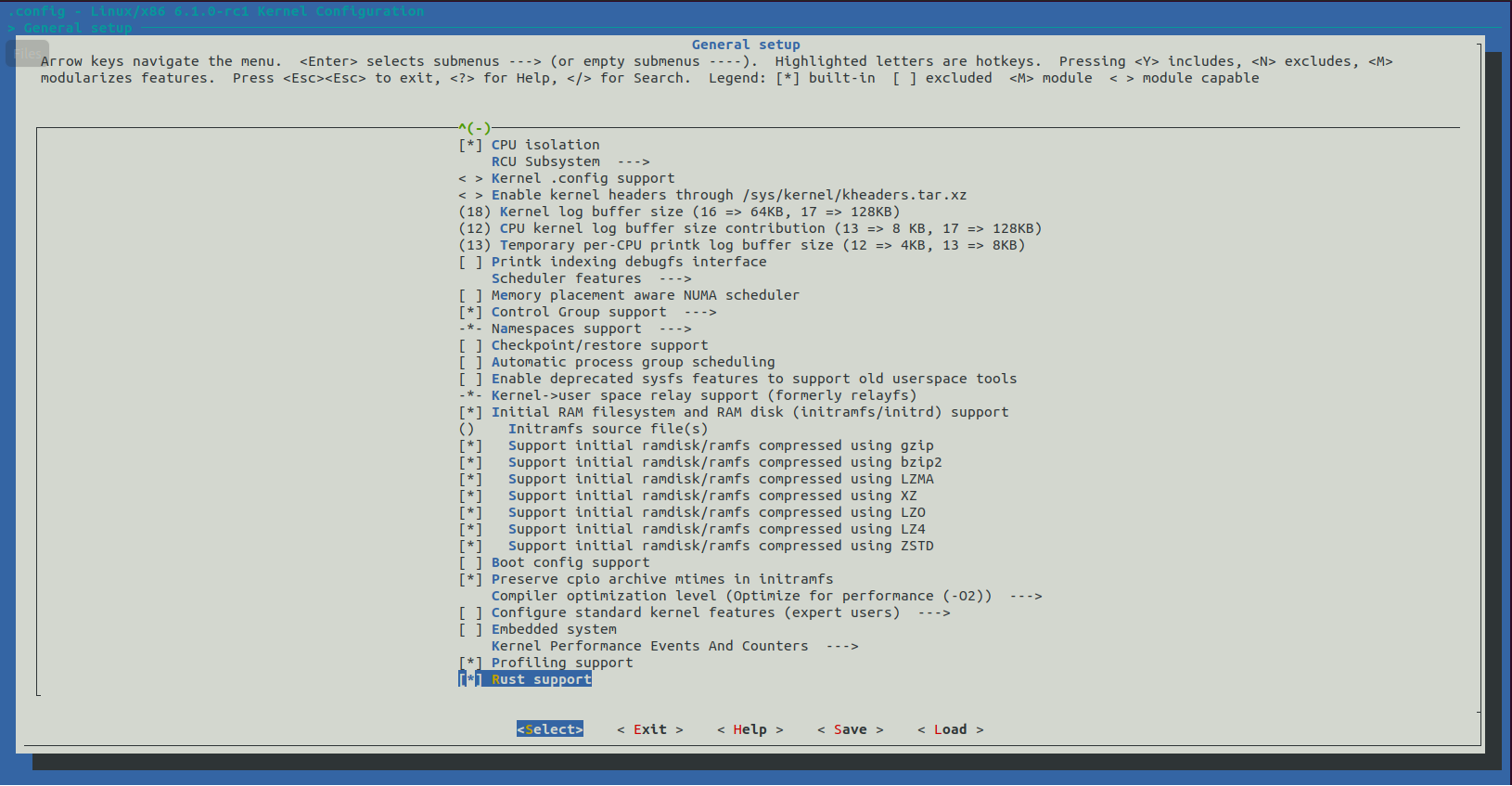
# 作业报告

## 作业1

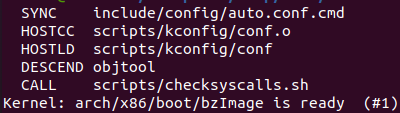
1. make x86\_64\_defconfig

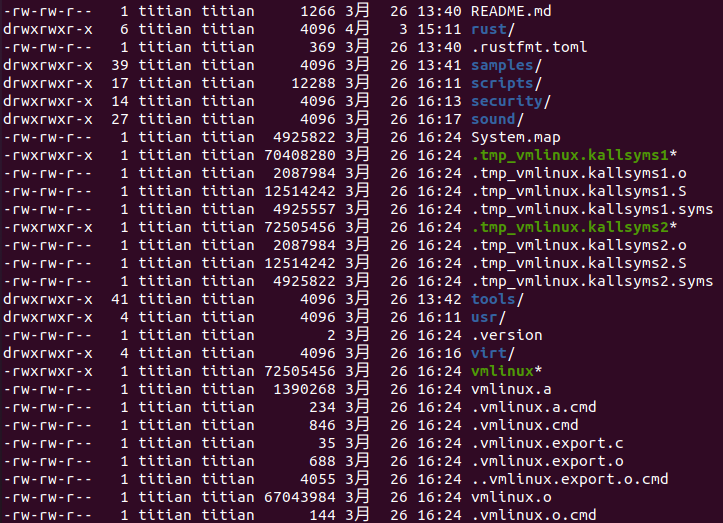


1. make LLVM=1 menuconfig



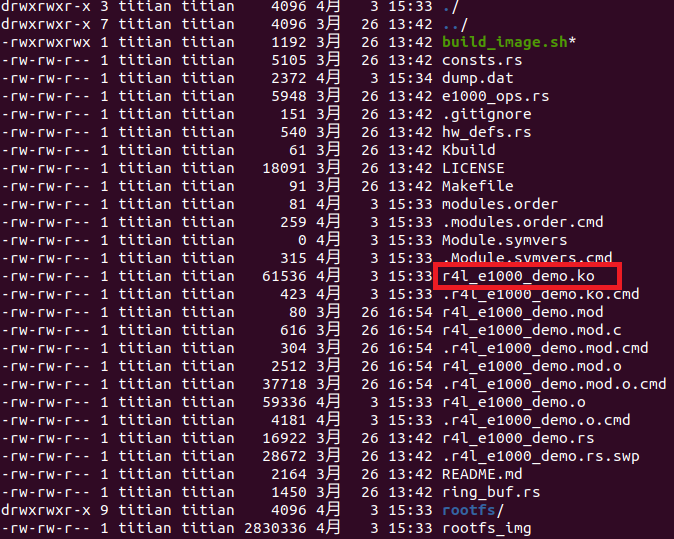
1. make LLVM=1 -j$(nproc)



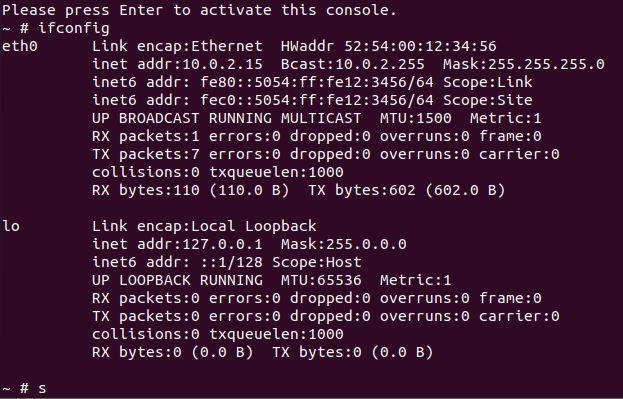


## 作业2

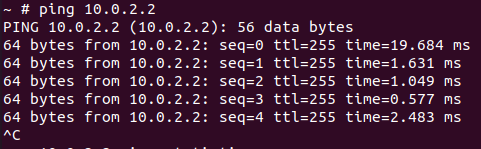
1. make LLVM=1



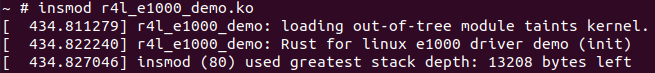
1. ./build\_image.sh && ifconfig



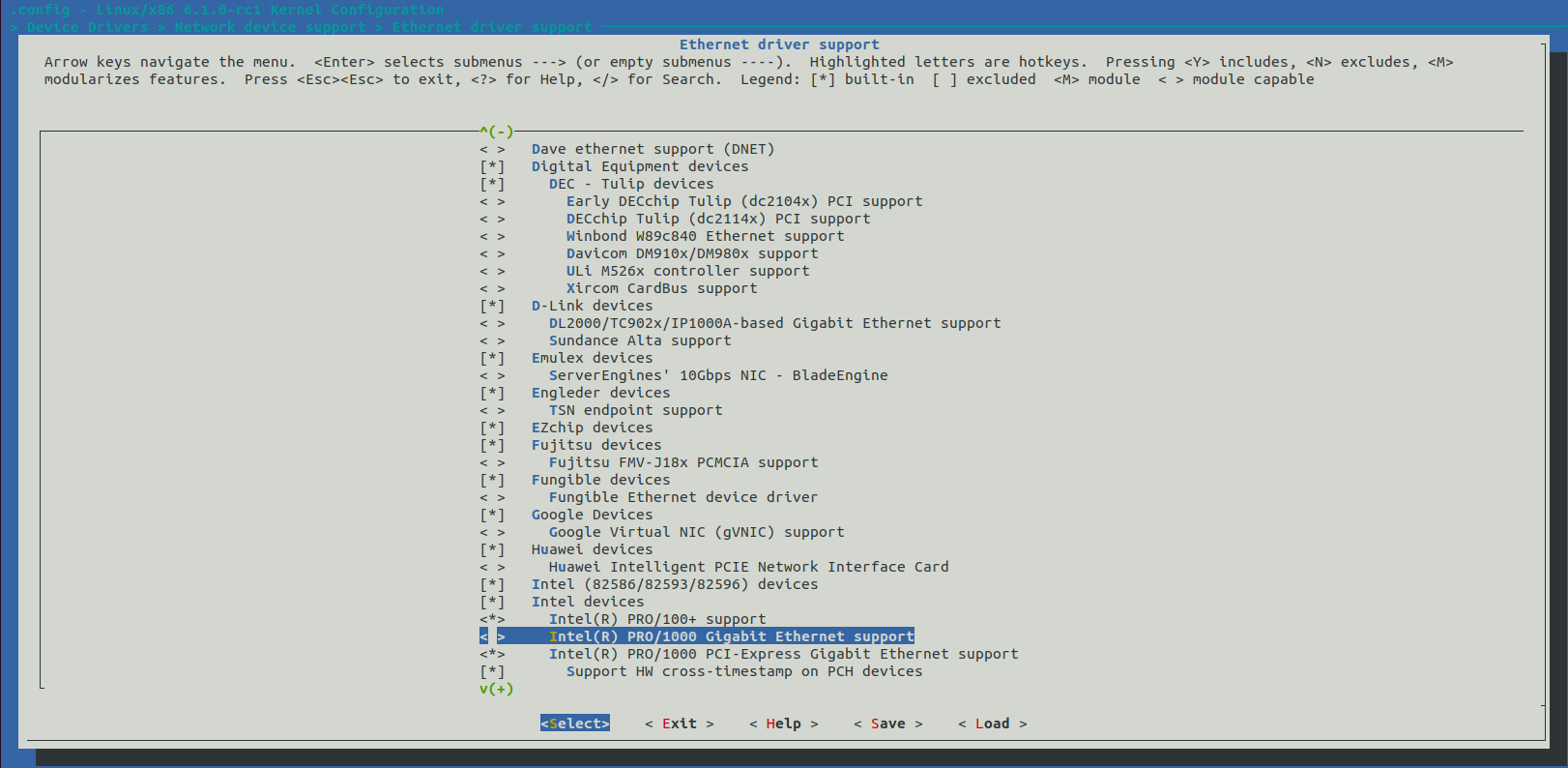
1. ping 10.0.2.2



1. insmod r4l\_e1000\_demo.ko



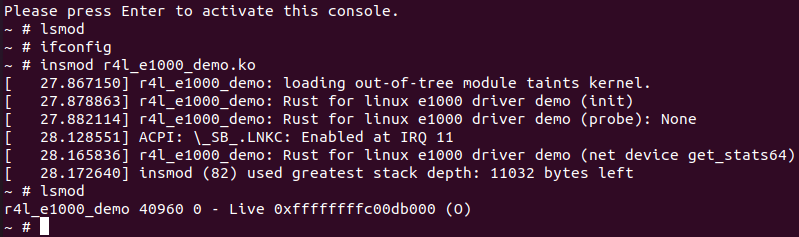
1. make LLVM=1 menuconfig



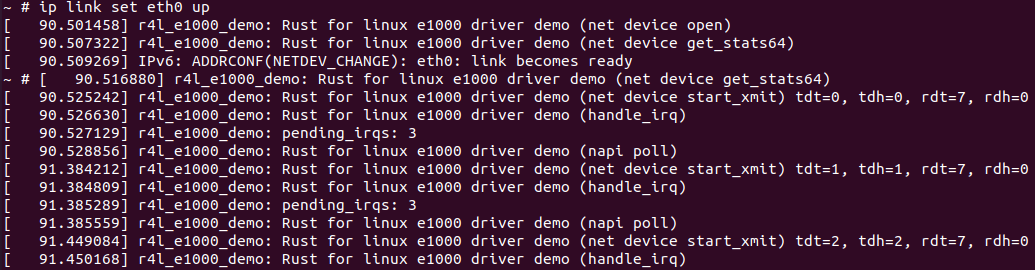
1. Ifconfig



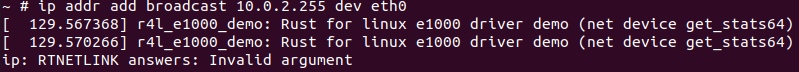
1. insmod r4l\_e1000\_demo.ko && Ifconfig && lsmod



1. ip link set eth0 up



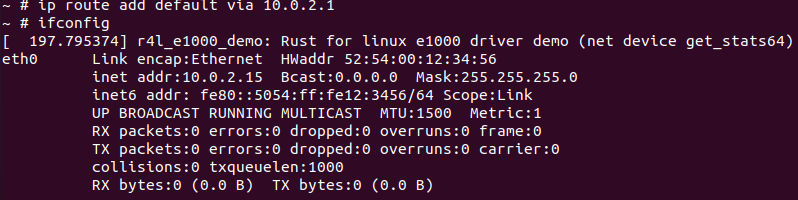
1. ip addr add broadcast 10.0.2.255 dev eth0



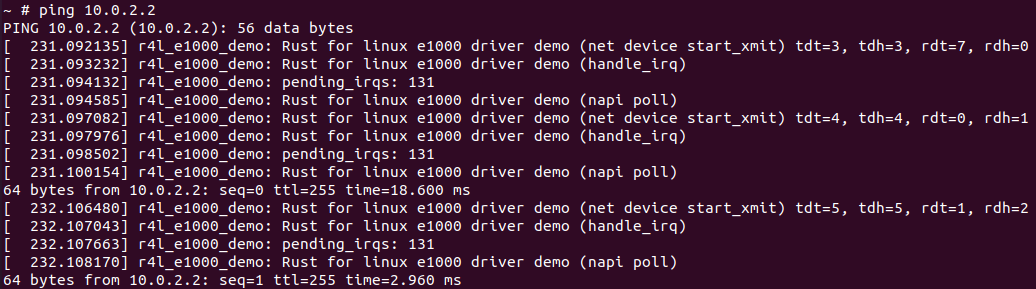
1. ip addr add 10.0.2.15/255.255.255.0 dev eth0



1. ip route add default via 10.0.2.1



1. ping 10.0.2.2



QA1编译成内核模块，是在哪个文件中以哪条语句定义的？

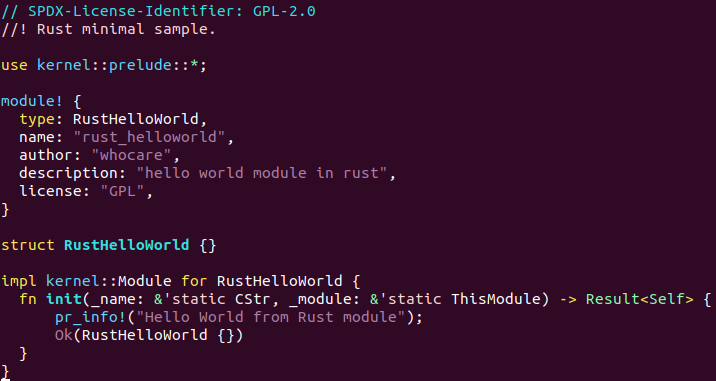
**Kbuild中如下定义“obj-m := r4l\_e1000\_demo.o”**

QA2该模块位于独立的文件夹内，却能编译成Linux内核模块，这叫做out-of-tree module，请分析它是如何与内核代码产生联系的？

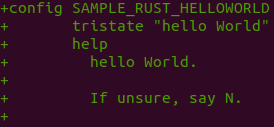
**build\_image.sh执行的时候将编译出来的内核模块r4l\_e1000\_demo.ko移动到rootfs目录中，使得我们在QEMU模拟器启动内核后可以借助insmod将它加载进内核。**

## 作业3

1. rust\_helloword.rs



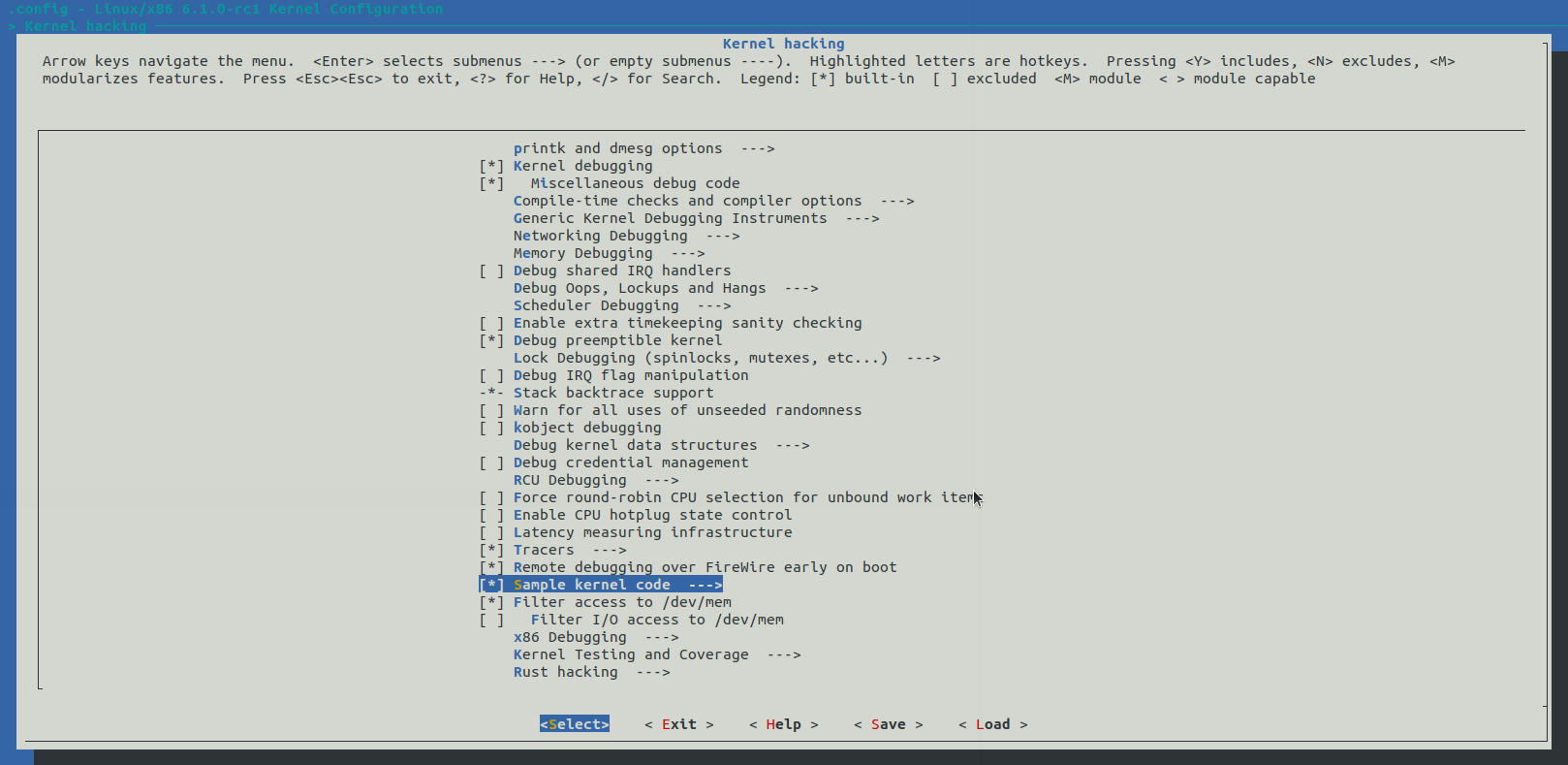
1. KConfig

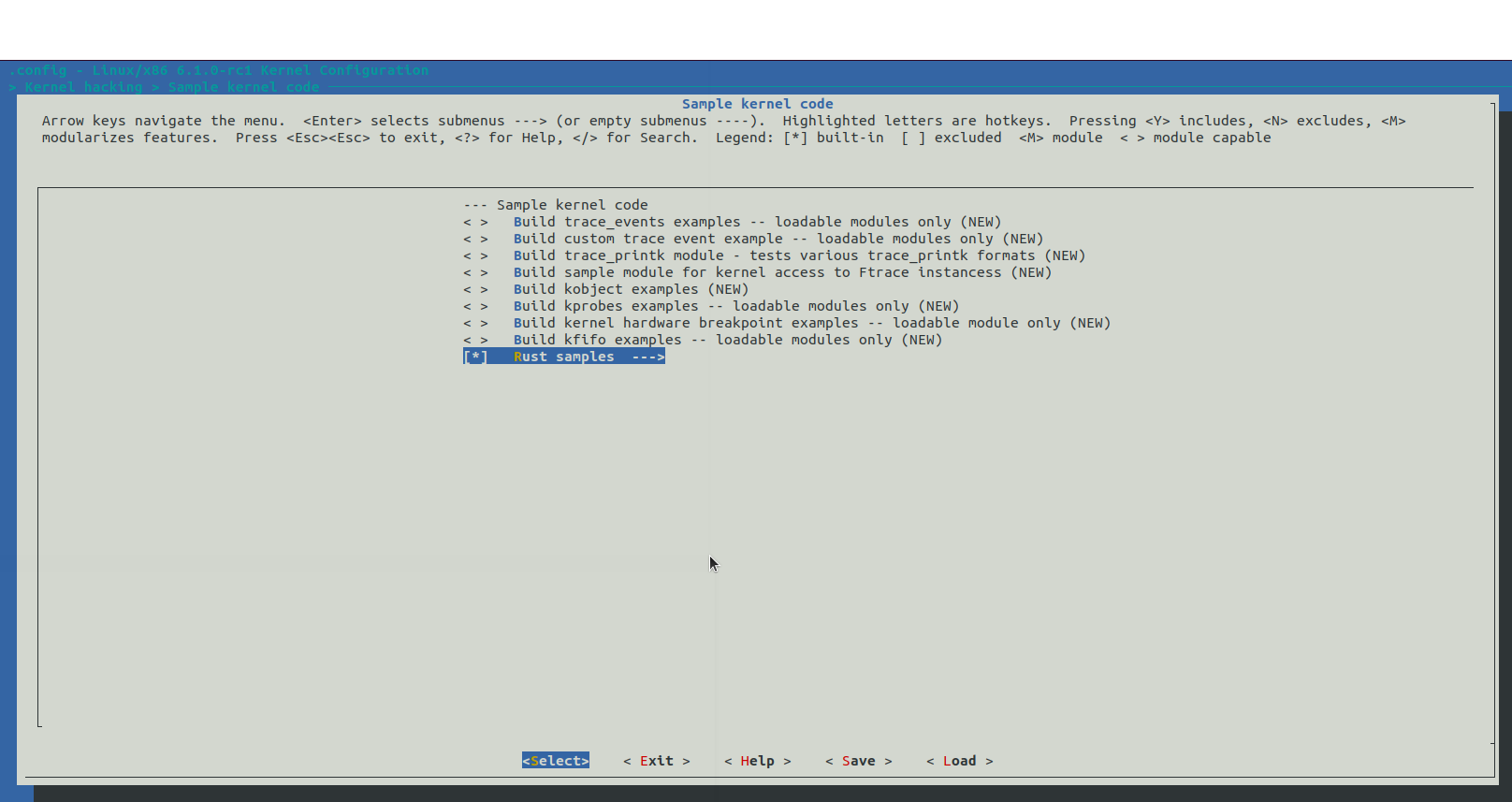


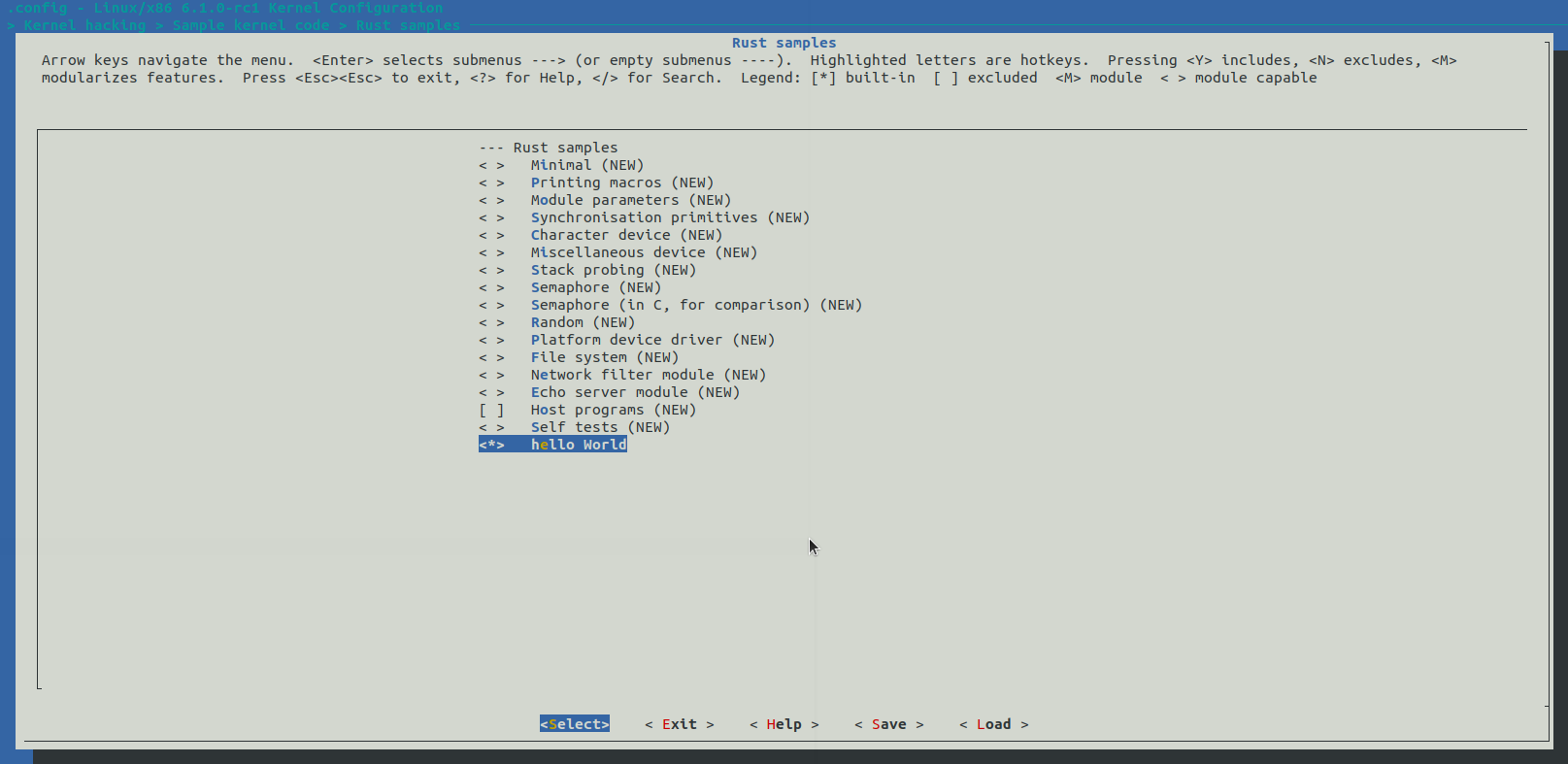
1. Makefile

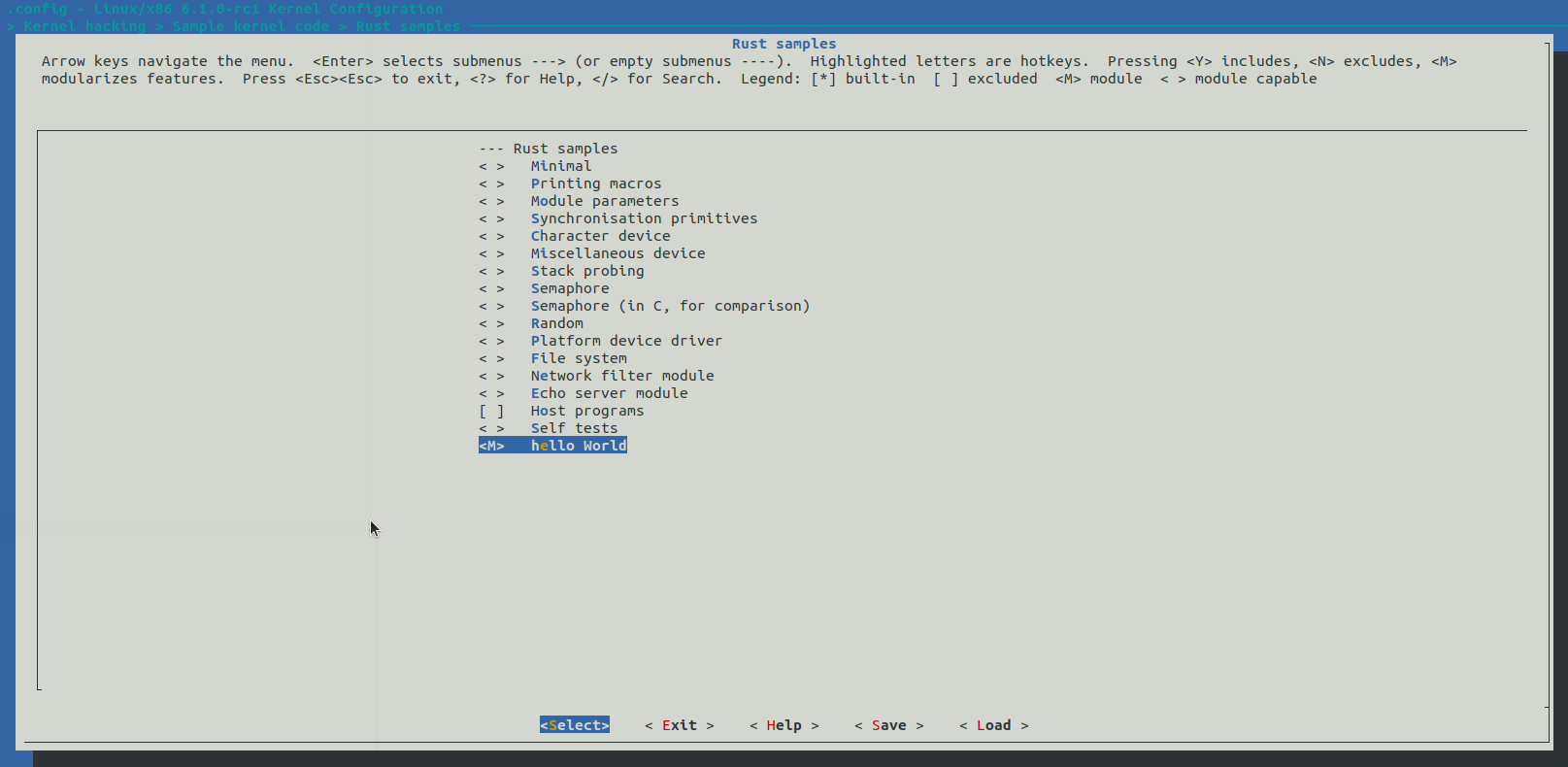


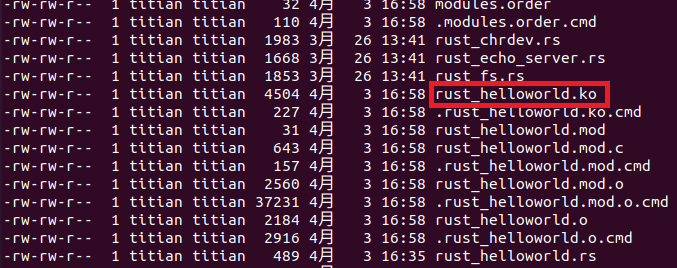
1. make LLVM=1 menuconfig



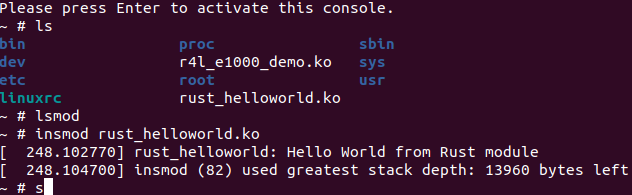






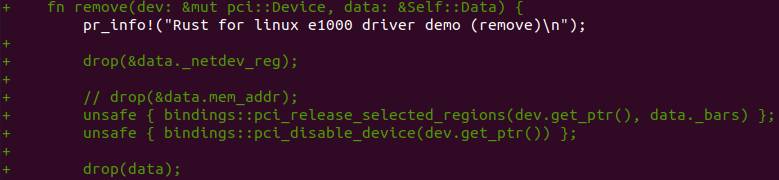
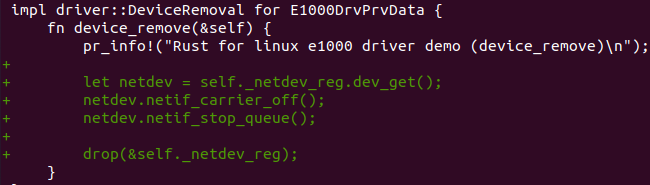
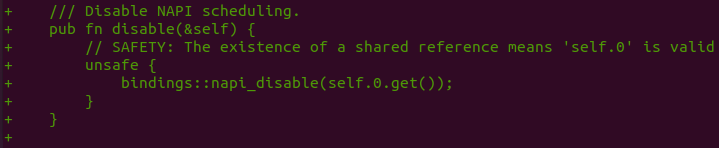


5.将编译出的rust\_helloworld.ko拷贝到rootfs目录下，执行./build\_image.sh

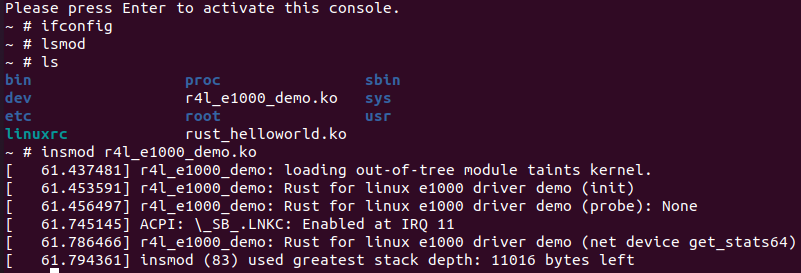
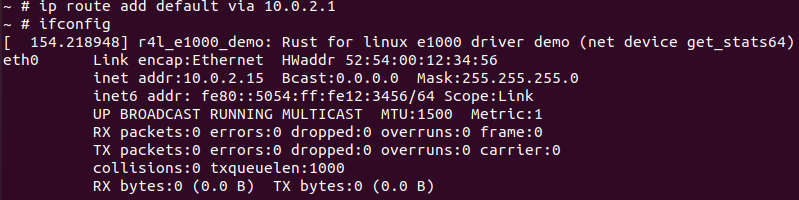


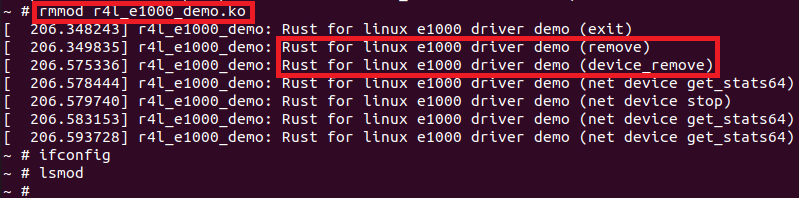
## 作业4

1. 修改代码主要内容如下



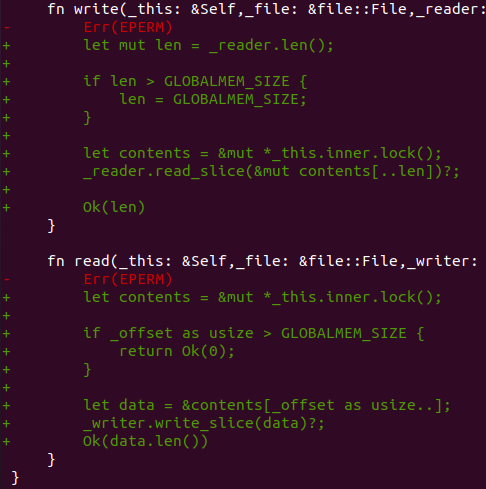
1. make LLVM=1 -j$(nproc) && ./build\_image.sh



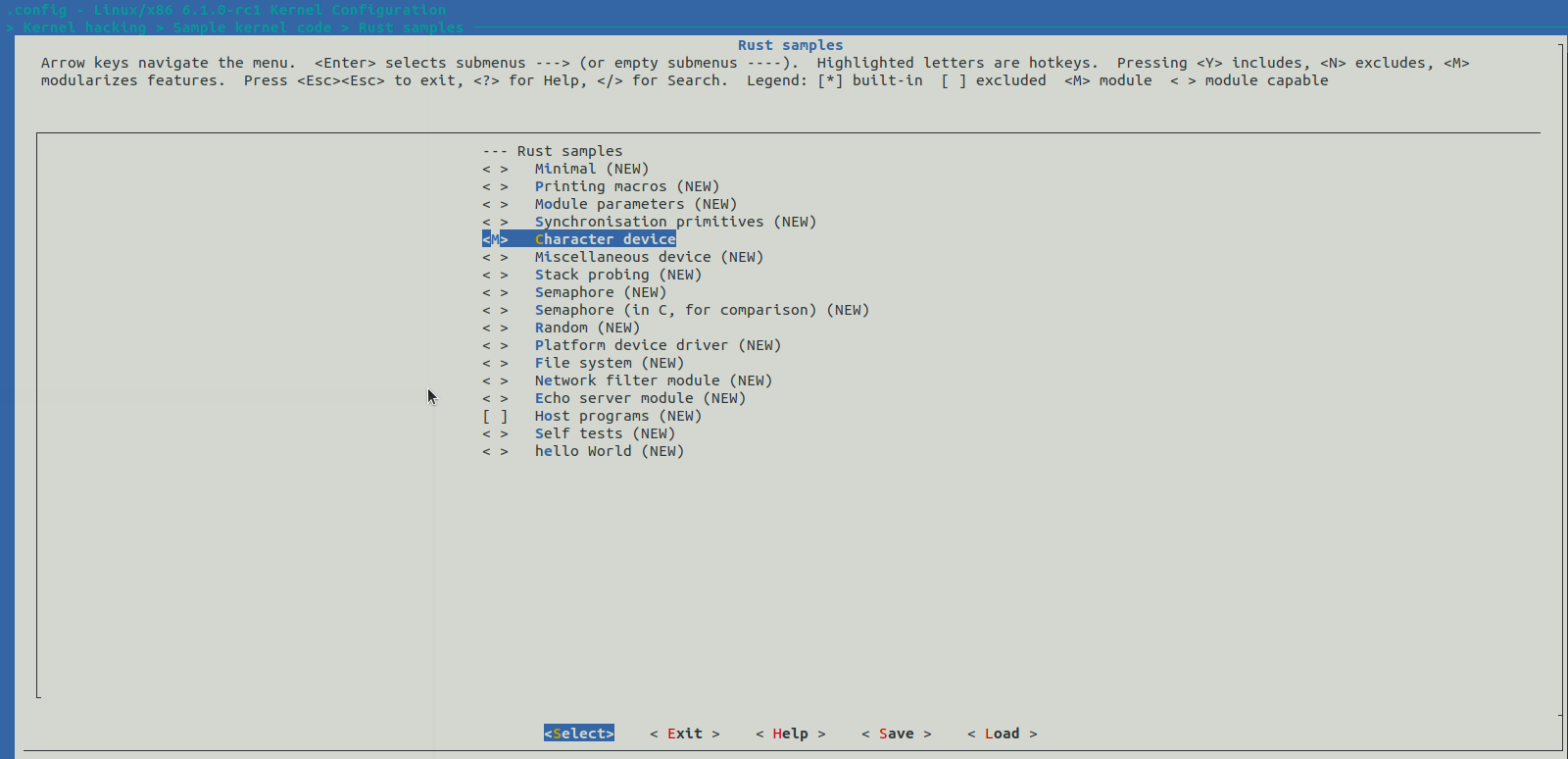


## 作业5

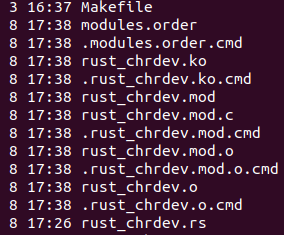
1. 修改代码



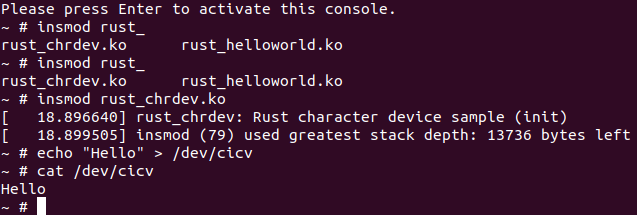
1. make LLVM=1 menuconfig



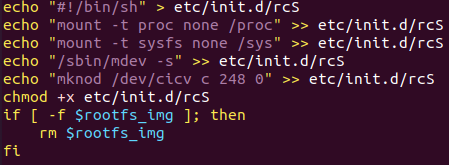
1. make LLVM=1 -j$(nproc)



1. 将编译出的rust\_chrdev.ko拷贝到rootfs目录下，执行./build\_image.sh



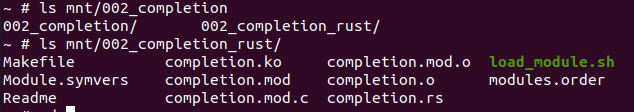
QA1作业5中的字符设备/dev/cicv是怎么创建的？它的设备号是多少？它是如何与我们写的字符设备驱动关联上的？



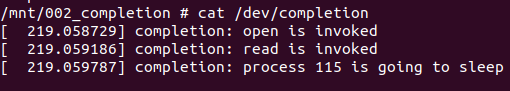
在build\_image.sh脚本中有以上内容，echo "mknod /dev/cicv c 248 0" >> etc/init.d/rcS这行命令将字符串"mknod /dev/cicv c 248 0" 追加到文件 etc/init.d/rcS 的末尾。mknod 是一个用于创建设备节点的命令，/dev/cicv 是要创建的设备节点的路径，c 表示这是一个字符设备节点，**248 是主设备号，0 是次设备号**。通过将这行命令追加到 rcS 脚本中，可以在系统启动时自动创建 /dev/cicv 设备节点。

## 实验

1. Cat /dev/completion







1. telnet localhost 7023

