**题目1：新增Linux系统调用**

1. **设计目的与要求**

本设计旨在通过在Linux内核中添加新的系统调用，深入理解操作系统内核的工作机制和系统调用的实现方法。此外，通过实际编程练习，增强对Linux内核编程和环境配置的掌握。

1. **设计内容**

采用编译内核法，在Linux中增加系统调用。系统调用实现的功能：

（1）计算一个数字的三次方，并打印出来。

（2）给定圆的半径计算圆的面积并打印出来

1. **设备与环境**

操作系统：Ubuntu 20.04

虚拟化软件：VMware Workstation 17 Pro

开发环境：Visual Studio Code

内核版本：5.6.8

1. **设计思想**

设计的主要思想是通过直接修改Linux内核源代码来添加自定义的系统调用，实现对核心功能的扩展。主要包括以下步骤：

1.获取当前Linux内核的源代码。

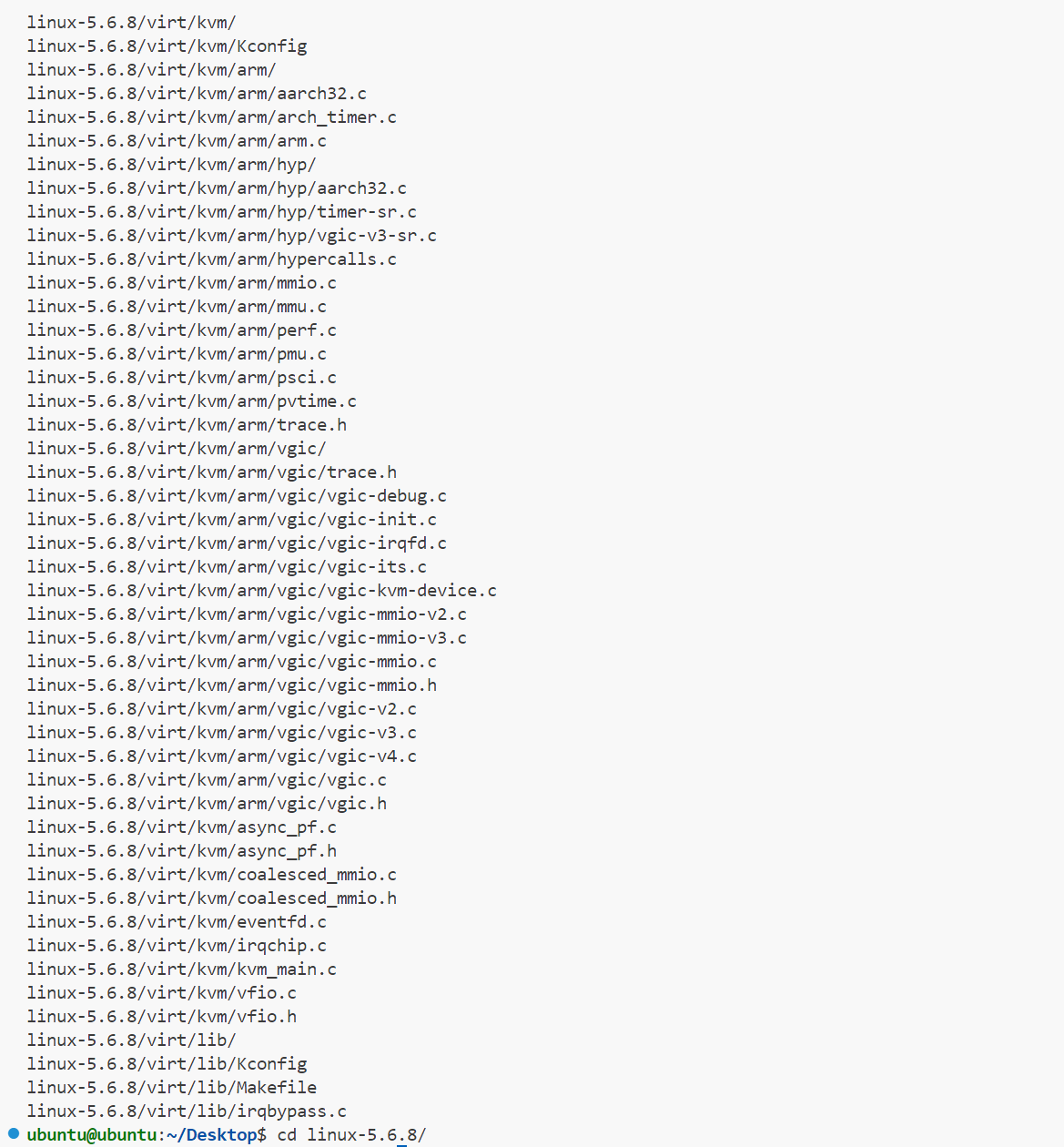
2.修改内核源代码以添加新的系统调用定义和实现。

3.修改系统调用表和相关头文件，注册新的系统调用。

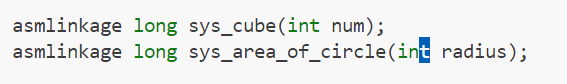
4.重新编译和加载内核，对新增的系统调用进行测试。

1. **主要数据结构和流程**

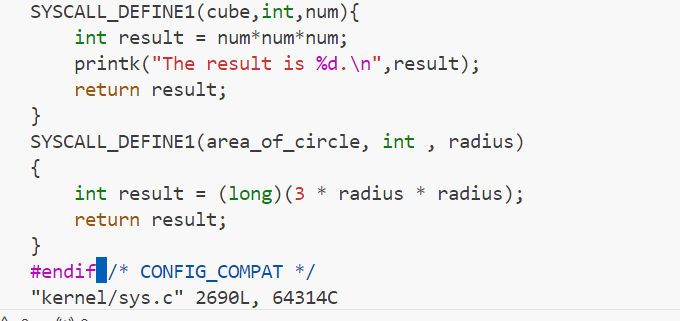
1.下载内核源码并解压



2.添加函数调用声明



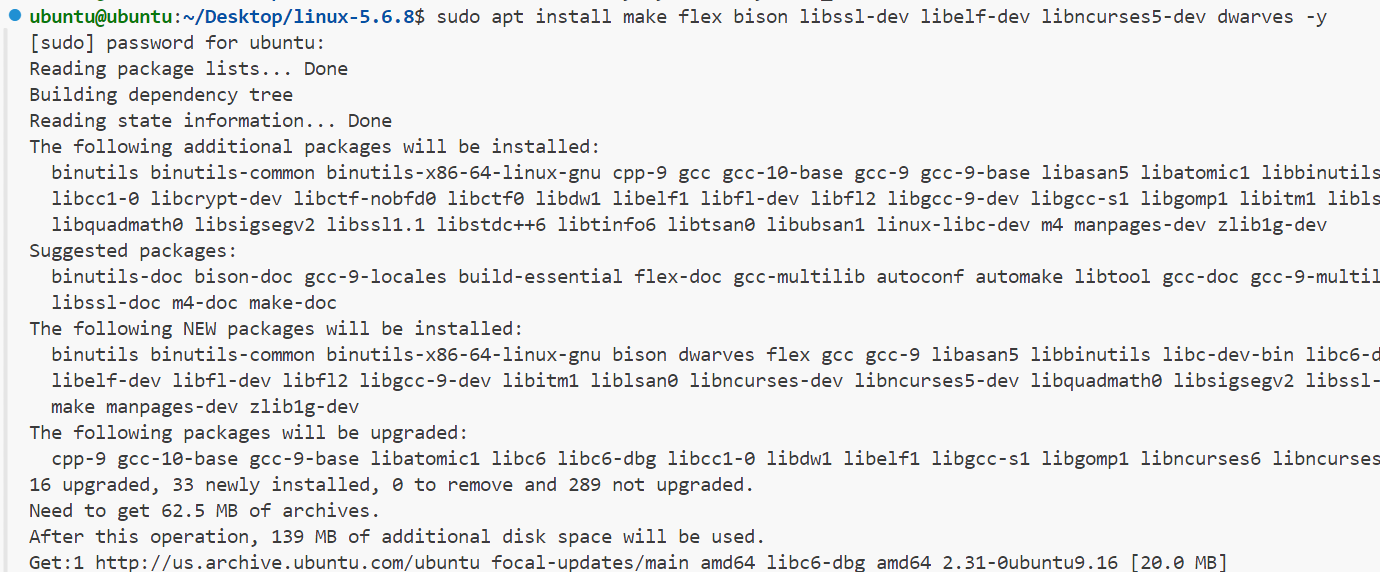
3.添加调用函数定义，在终端中输入 vim kernel/sys.c ,在文件中输入求三次方函数



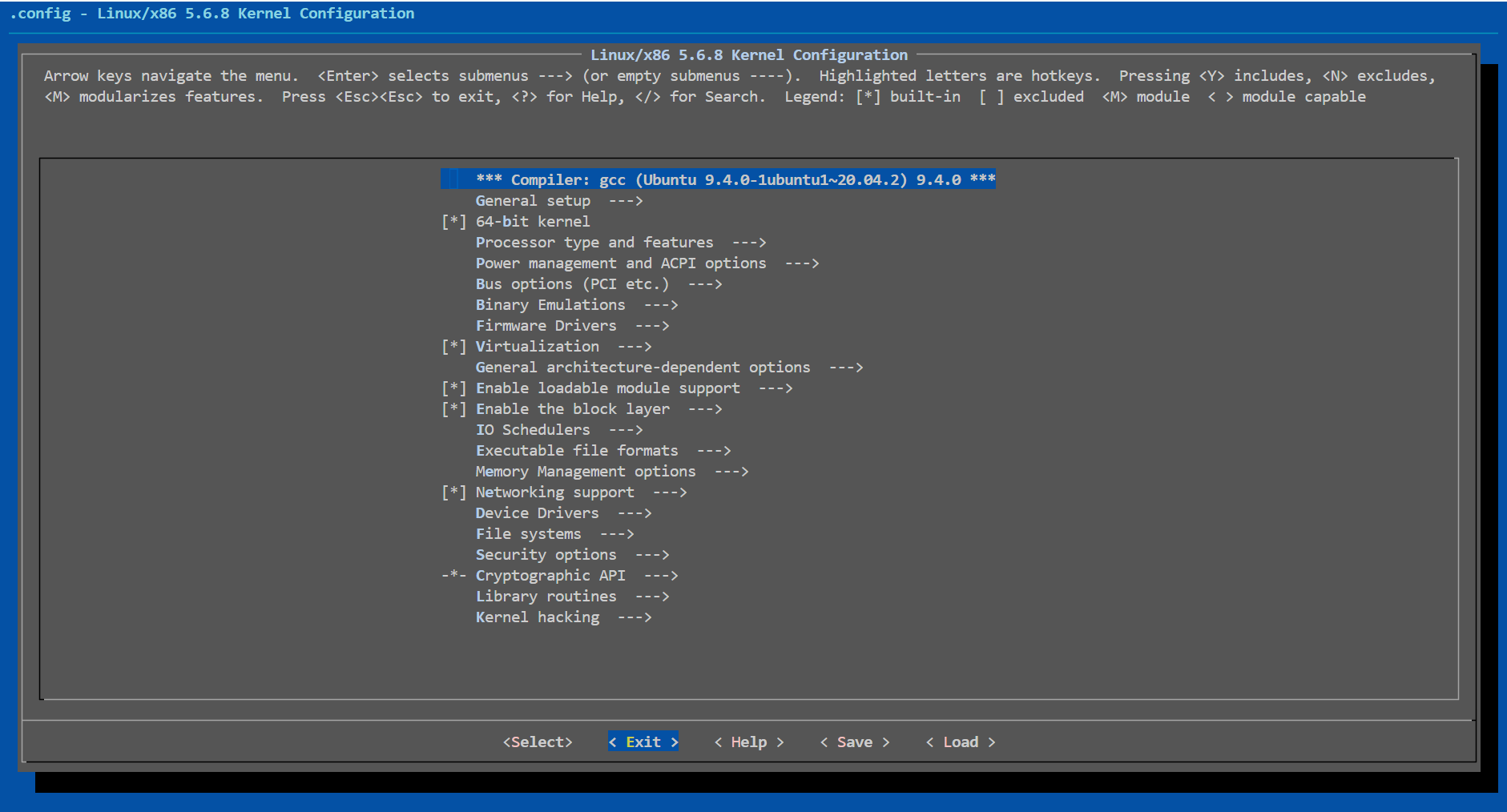
4. 注册系统调用号，输入 vim arch/x86/entry/syscalls/syscall\_64.tbl ，在syscall\_64.tbl文件中添加函数系统调用号分别为439和440



5.安装编译内核所需的依赖包



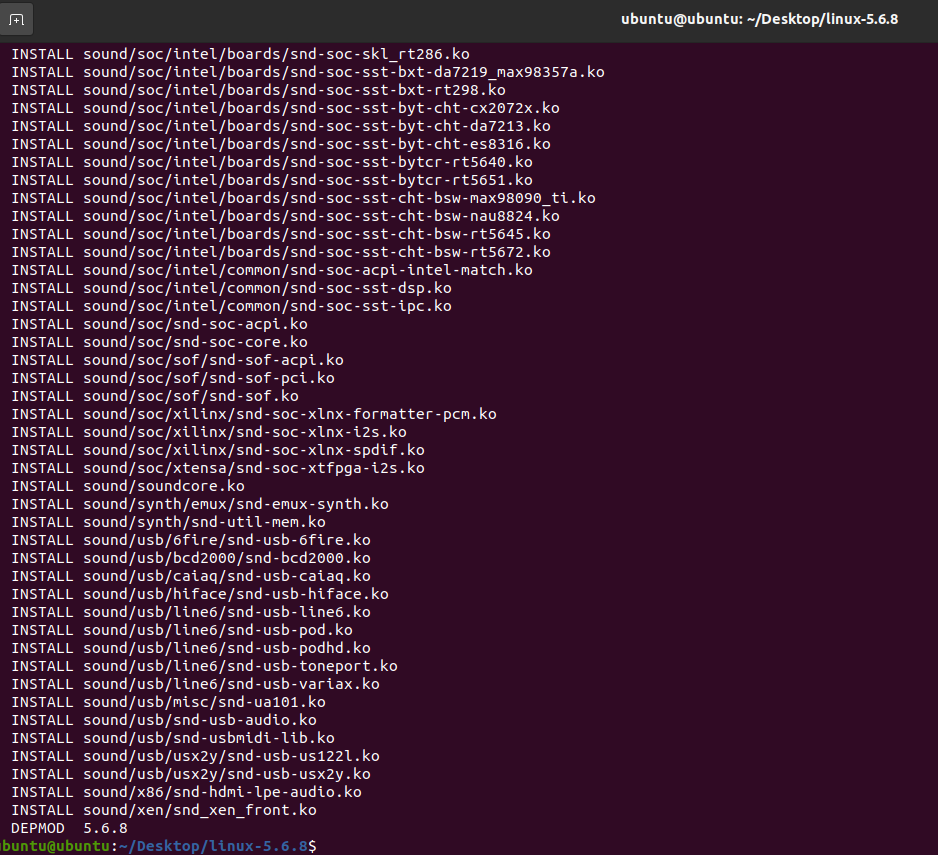
6. 配置内核，输入**make menuconfig**，在跳出的界面中进行配置



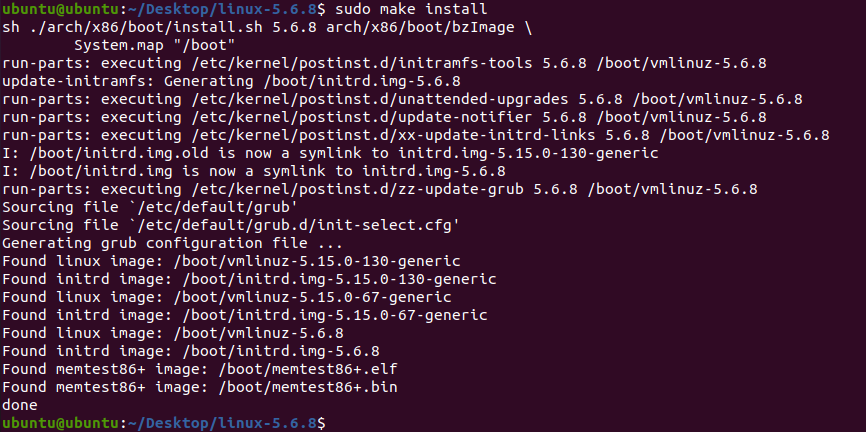
7.编译，查看自己机器处理器的核数，中键入 **make -jn**（n个线程开始编译）



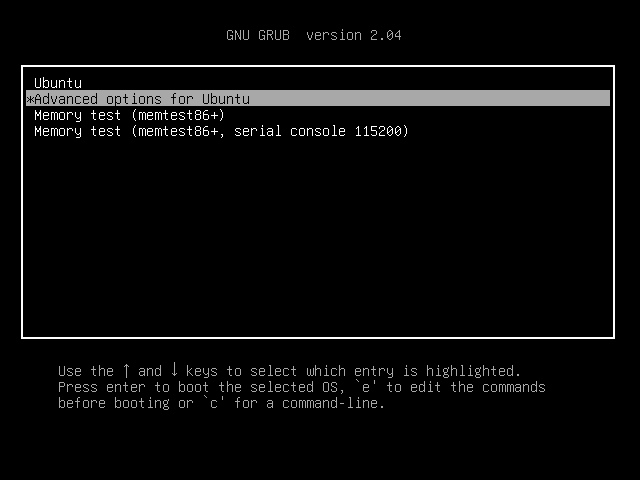
1. 安装编译好的模块，输入sudo make modules\_install 命令，完成模块的安装操作

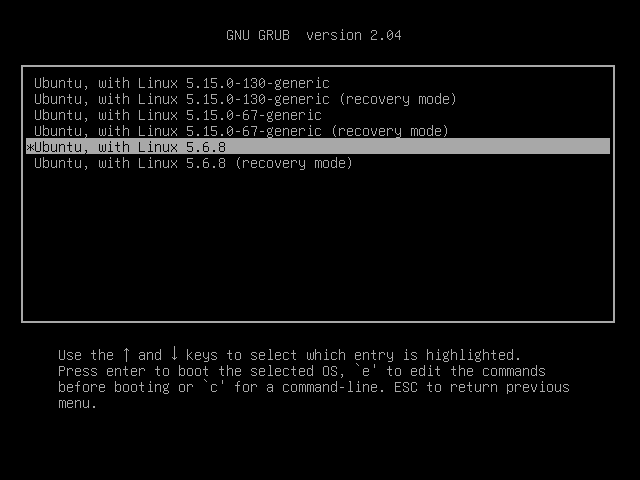


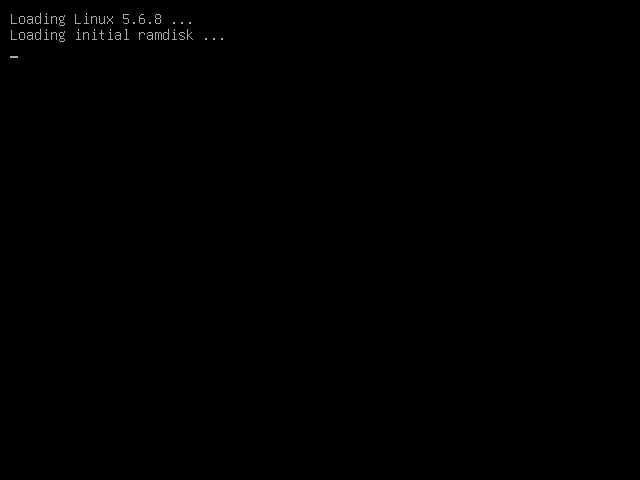
9. 输入 sudo make install安装内核

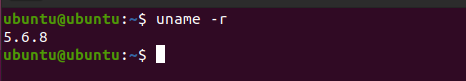


10.重启，内核切换成功





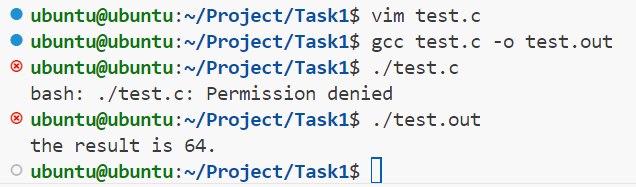




1. **实验测试结果及结果分析**

1.编写程序测试三次方函数





由the result is 64可以看出系统调用成功。



