

本科生实验报告

实验课程: 操作系统原理实验

实验名称: 添加新的系统调用 专业名称: 计算机科学与技术

学生姓名: 张玉瑶

学生学号: 23336302

实验地点:

实验成绩:

报告时间: 2025年3月27日

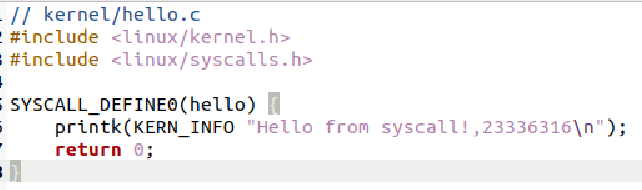
**Section 1 实验概述**

本次实验学习在linux内核中添加新的系统调用，以便我们更加深入地理解内核的工作原理。

**Section 2 实验步骤与实验结果**

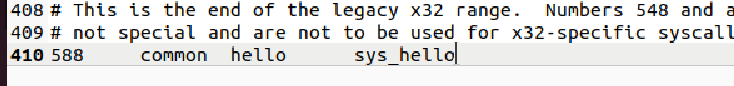
1. 实现系统调用函数

在linux5.10.235文件中打开kernel文件，创建hello.c文件后输入如下代码。



2.添加系统调用号

依照arch/x86/entry/syscalls/syscall\_64.tbl，进入文件在最后一行添加如下语句。



3.修改linux5.10.235/kernel中的Makefile文件，添加如下语句。（注意不是外面的kernel！）



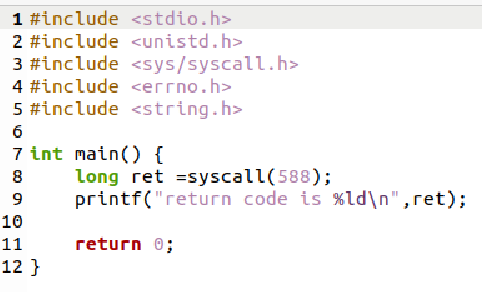
4.修改系统调用头文件

在include/linux/syscalls.h中文件末尾、#end if之前添加如下语句。



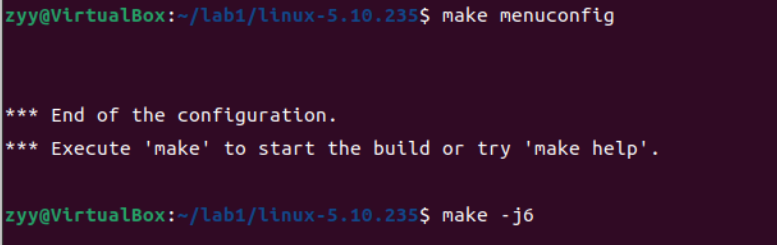
5.增加测试文件

在lab1创建文件test\_syscall.c，输入如下代码。

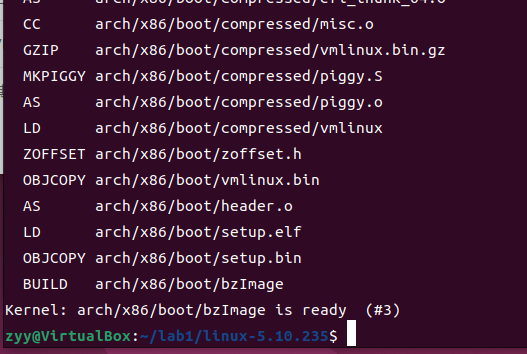


6.编译内核

在linux-5.10.235的终端中输入以下代码。



最后可以看到语句Kernel: arch/x86/boot/bzImage is ready ，说明编译成功！



7.安装内核



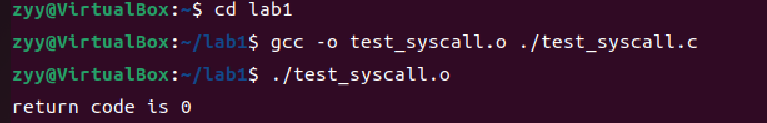
重启，重启时长按shift进入选择项选择advanced options for ubuntu选项。



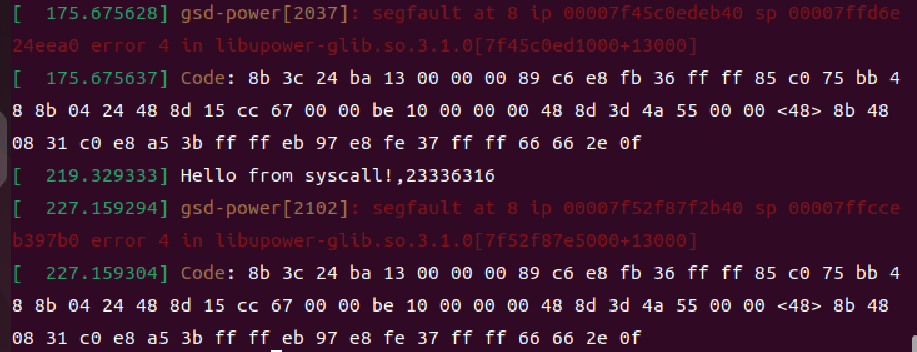
进入如下内核。



在lab1终端中编译text\_syscall.c文件，再进行测试。可以看到return了0。



最后输入dmesg，返回如下，成功执行了新的系统调用！



**Section 3 实验总结与心得体会**

这次增加系统调用的实验看似简单，步骤也少，但是还是吃了很多亏。第一次吃亏是没有分辨清楚不同文件下的Makefile，改错了导致我重新搞了一个内核，很崩溃。第二次是不知道为什么编译总要一两个小时还经常报错，忍无可忍，直到现在我都不知道错在哪。总之过程非常折磨，非常搞我心态，但看到成功调用出来“Hello from syscall！，23336316“的时候还是挺开心的。