Linux内核课程项目：系统调用的动态修改

1、实验目的：

1）系统调用在内核是如何被调用的。

2）操作系统中的页表是怎么构建的。

3）如何去遍历页表。

4）如何修改页表。

5）各级页表中页表项的各个字段是什么含义。

6）当编写的驱动发生crash和panic时，如何去debug。

2、实验要求：

1）编写一个内核模块。

实验环境：ARM64，x86-64均可，Linux 5.0以上内核。

要求替换系统调用表（sys_call_table) 中某一项系统调用，替换成自己编写的系统调用处理函数（例如my_syscall()），在新的系统调用函数中打印一句 ”hello，I have hacked this syscall"，然后再调用回原来的系统调用处理函数。

比如以ioctl系统调用为例，它在系统调用表中的编号是__NR_ioctl. 那么需要修改系统调用表sys_call_table[__NR_ioctl]的指向，让其指向my_syscall(函数，然后在my_syscall()函数中打印一句话，调用原来的sys_call_table[__NR_ioctl]指向的处理函数。

2）卸载模块时把系统调用表恢复原样。

3）用clone系统调用来验证你的驱动，clone系统调用号是__NR_clone。

3、实验附件：

1）test.o

2）benchmark.o

实验效果

dmesg: 系统日志输出一次“hacked”信息；

运行test.o：系统日志输出一次“hacked”信息；

运行benchmark.o: 系统日志输出六次“hacked”信息；

作业验收与提交

验收方式：电院3号楼118向助教展示实验效果。

提交渠道：Canvas

提交文件："学号_姓名_final_project.zip"

- 源码文件夹"学号_姓名_final_project_src"（*.c Makefile）
- 实验报告《学号_姓名_final_project_report.pdf》，包括但不限于实验过程、实验分析、实验效果截图、实验心得。