**Project 1: Introduction to Linux Kernel Modules**

I. Kernel Modules Overview

II. Loading and Removing Kernel Modules

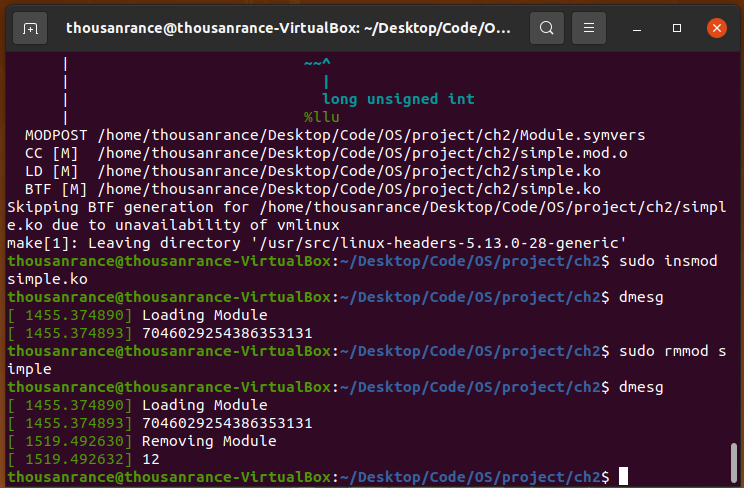
1. Print out the value of GOLDEN RATIO PRIME in the simple\_init() function.  
2. Print out the greatest common divisor of 3,300 and 24 in the simple\_exit() function.

实现细节：

<linux/hash.h>: printk(KERN INFO "%lu∖n", GOLDEN RATIO PRIME);

<linux/gcd.h>: unsigned long gcd(unsigned long a, unsigned b);

按照书上的提示，导入库，然后将上述代码分别加入到simple\_init()与simple\_exit()函数中。

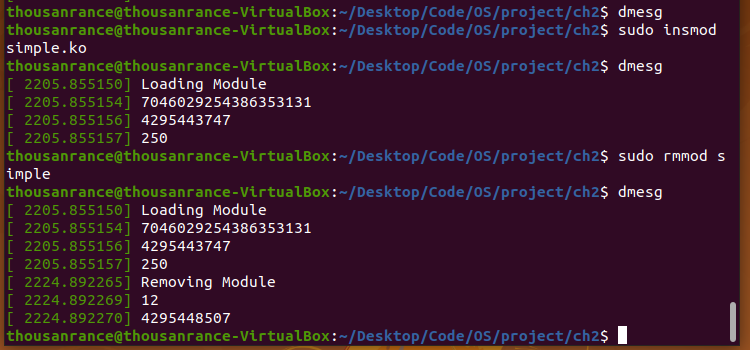


1. Print out the values of jiffies and HZ in the simple\_init() function.

2. Print out the value of jiffies in the simple\_exit() function.

实现细节：

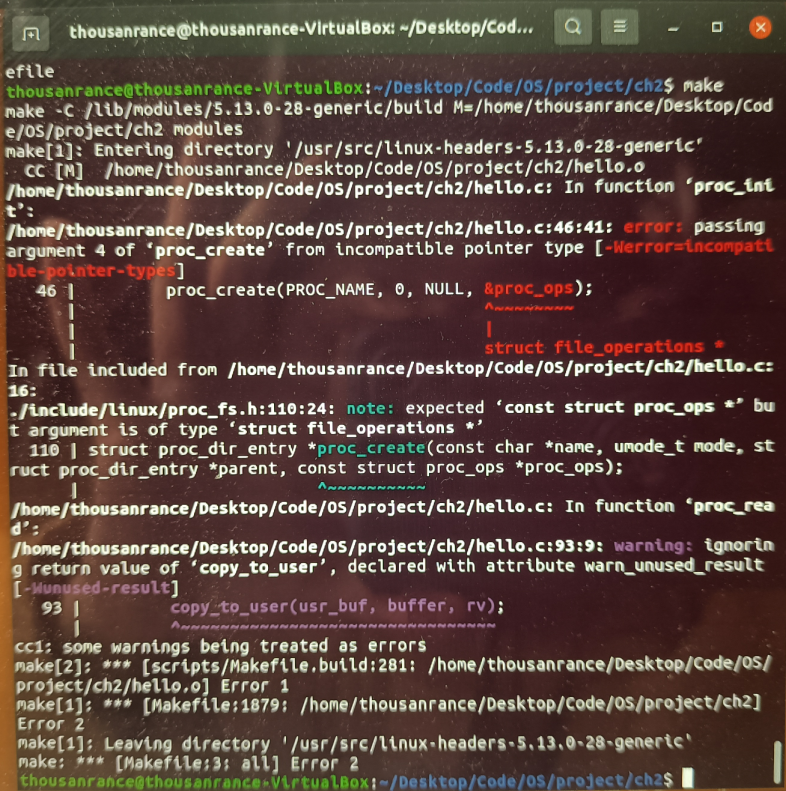
按照书上提示导入库<asm/param.h><linux/jiffies.h>，然后在simple\_init()与simple\_exit()函数中通过printk()语句打印要求的值。



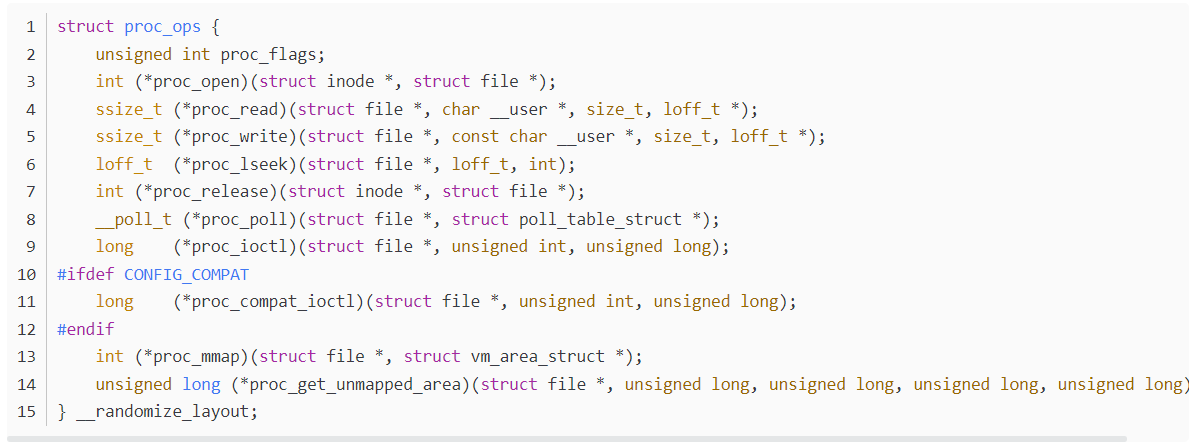
模块自加载到删除经过的秒数 =（删除时的jiffies值 - 加载时的jiffies值）/ HZ的值

Ⅲ The /proc File System

这部分原本只需要按照书上步骤操作完成，但是可能由于linux内核版本更新，proc\_create()函数的最后一个参数类型不再是像书上写的那样是file\_operation结构体指针，而是一个proc\_ops结构体指针，原本的示例代码hello.c在make时反而会报错。

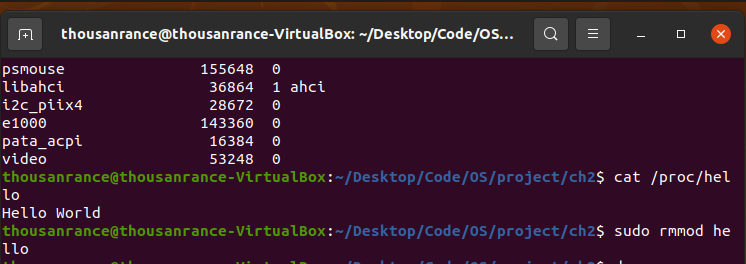


经过资料搜索，两结构体在头文件<linux/proc\_fs.h>中的声明如下：

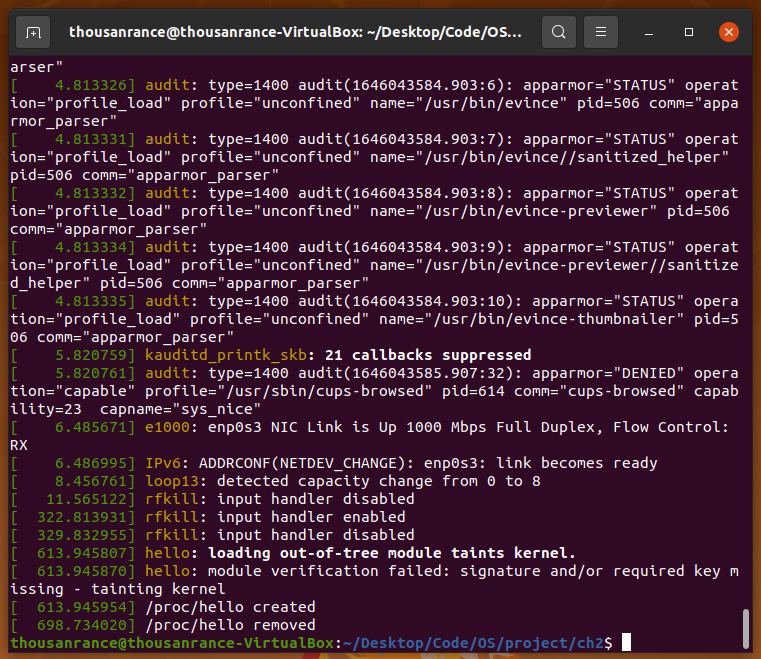




据此对示例代码hello.c进行了修改，注释掉原本的file\_operation结构体，补写了proc\_ops结构体，最终编译成功。



proc\_ops结构体的缺点是，不再有struct module \*owner这个参数，没有指向该结构体所属模块的指针，所以在逻辑上是一个不属于这个模块的结构体。



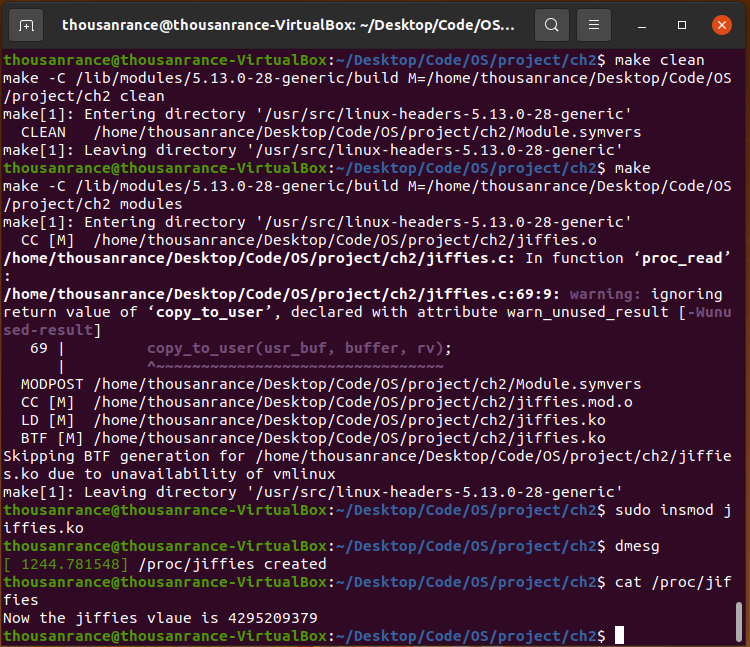
Ⅳ.Assignment

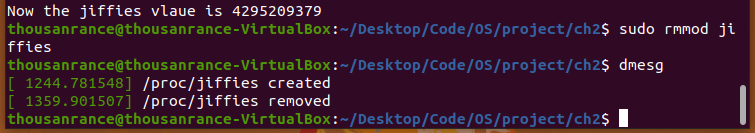
1、Design a kernel module that creates a /proc file named /proc/jiffies that reports the current value of jiffies when the /proc/jiffies file is read, such as with the command

cat /proc/jiffies

Be sure to remove /proc/jiffies when the module is removed.

实现细节：基本仿照hello.c，导入库<linux/jiffies.h>，然后只需要对proc\_read()函数中的printf()的参数进行修改，输出要求的jiffies的值即可。





2、Design a kernel module that creates a proc file named /proc/seconds that reports the number of elapsed seconds since the kernel module was loaded. This will involve using the value of jiffies as well as the HZ rate. When a user enters the command

cat /proc/seconds

your kernel module will report the number of seconds that have elapsed since the kernel module was first loaded. Be sure to remove /proc/seconds when the module is removed.

实现细节：基本仿照hello.c，导入库<asm/param.h><linux/jiffies.h>，然后只需要对proc\_read()函数中的printf()的参数进行修改，利用公式“模块自加载到删除经过的秒数 =（删除时的jiffies值 - 加载时的jiffies值）/ HZ的值”，输出要求的值即可。

