动态模块

动态模块实验的本质其实就是将自己写的模块放入到内核中，然后在内核进行一定的操作，观察输出的信息来判断模块状态。

操作注意事项：  
很重要的一点是这里面要用到超级用户的权限，那么如何将权限从普通用户升级到超级用户呢？

Su命令：

说起来很简单，首先在普通用户（calmovely）下，在terminal中输入信息

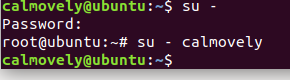
**Su**

这样就从普通转换成root用户了。

当然，一直在root状态会使系统处于不安全的状态，所以在实验结束后要返回到普通用户状态，即

**Su – calmovely**

如下：



OK, 前提条件结束，正文开始。

实验内容：

首先执行命令：

Vim mymodules.c

在文件中写入一下代码：  
#include<linux/init.h>

#include<linux/kernel.h>

#include<linux/module.h>

static int mymodule\_init(void)

{

printk("hello, my module was loaded!\n");

return 0;

}

static void mymodule\_exit(void)

{

printk("goodbye, unloading my module.\n");

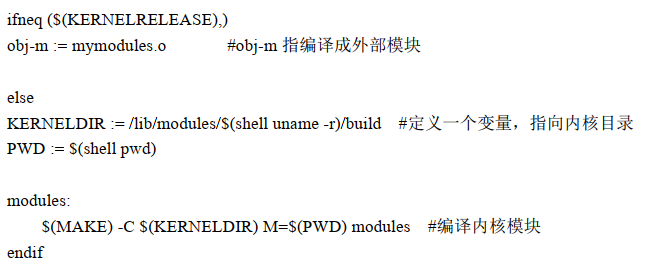
}

module\_init(mymodule\_init);

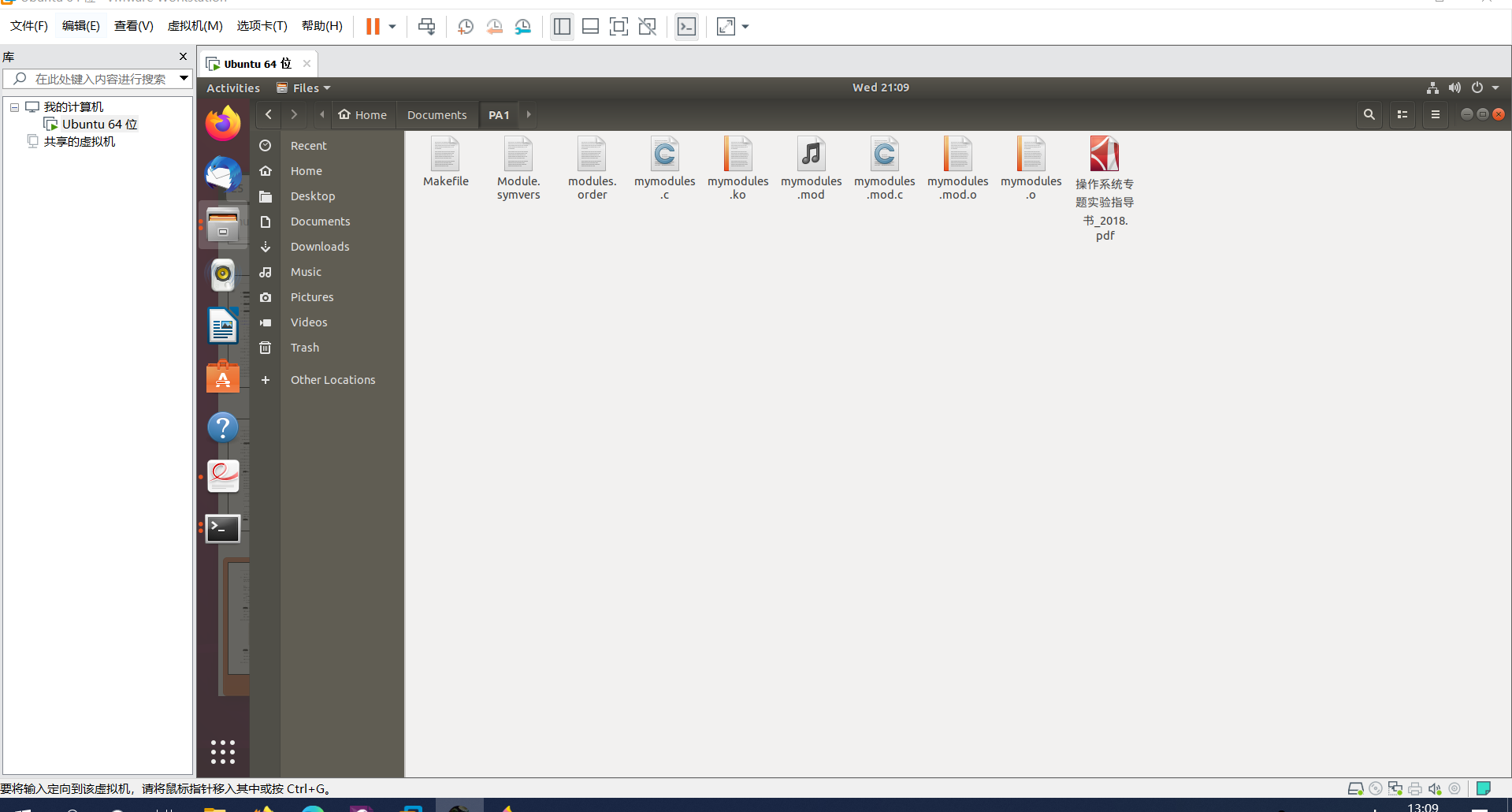
module\_exit(mymodule\_exit);

MODULE\_LICENSE("GPL");

在同一个文件夹下创建makefile文件，写入内容：

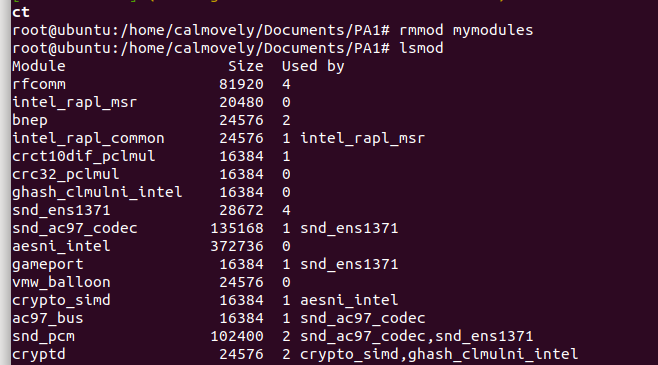


接下来就简单了，打开terminal，升级权限，在terminal中输入命令:make，完成模块的编译，编译结果如图：

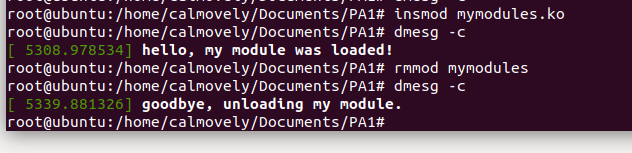


再使用命令insmod mymodules.ko

如图：

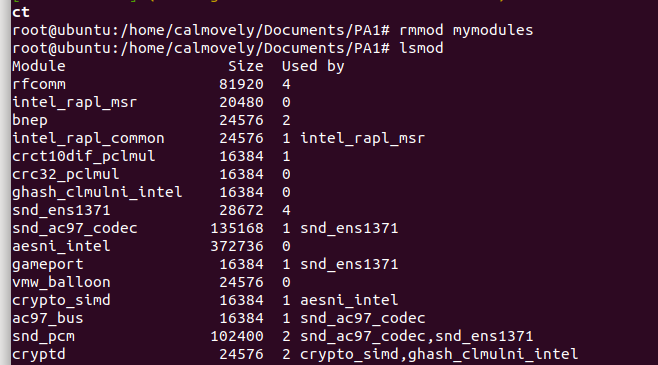


Dmesg -c命令是打印模块内容并清除缓存区



接下来直接remove模块释放内存

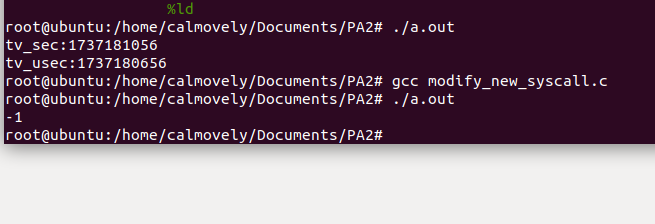
观察lsmod



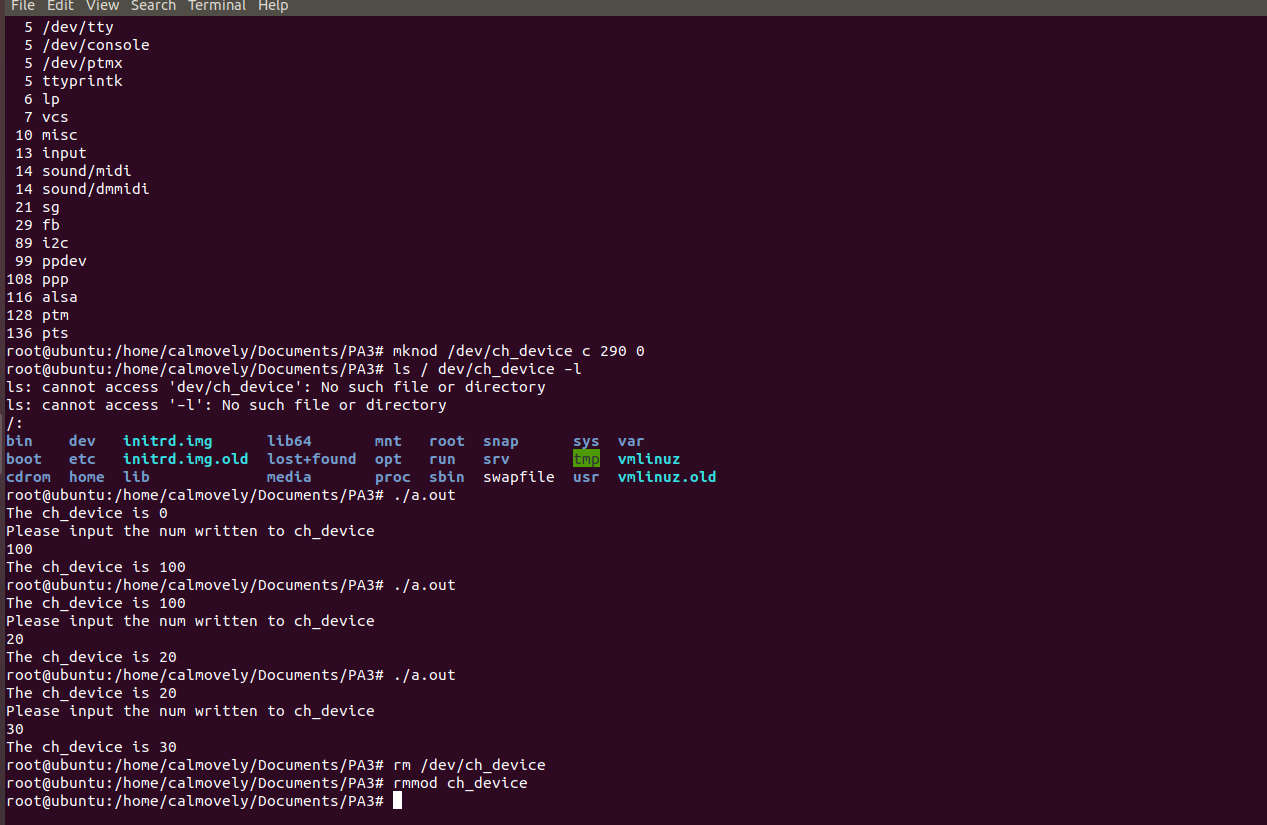
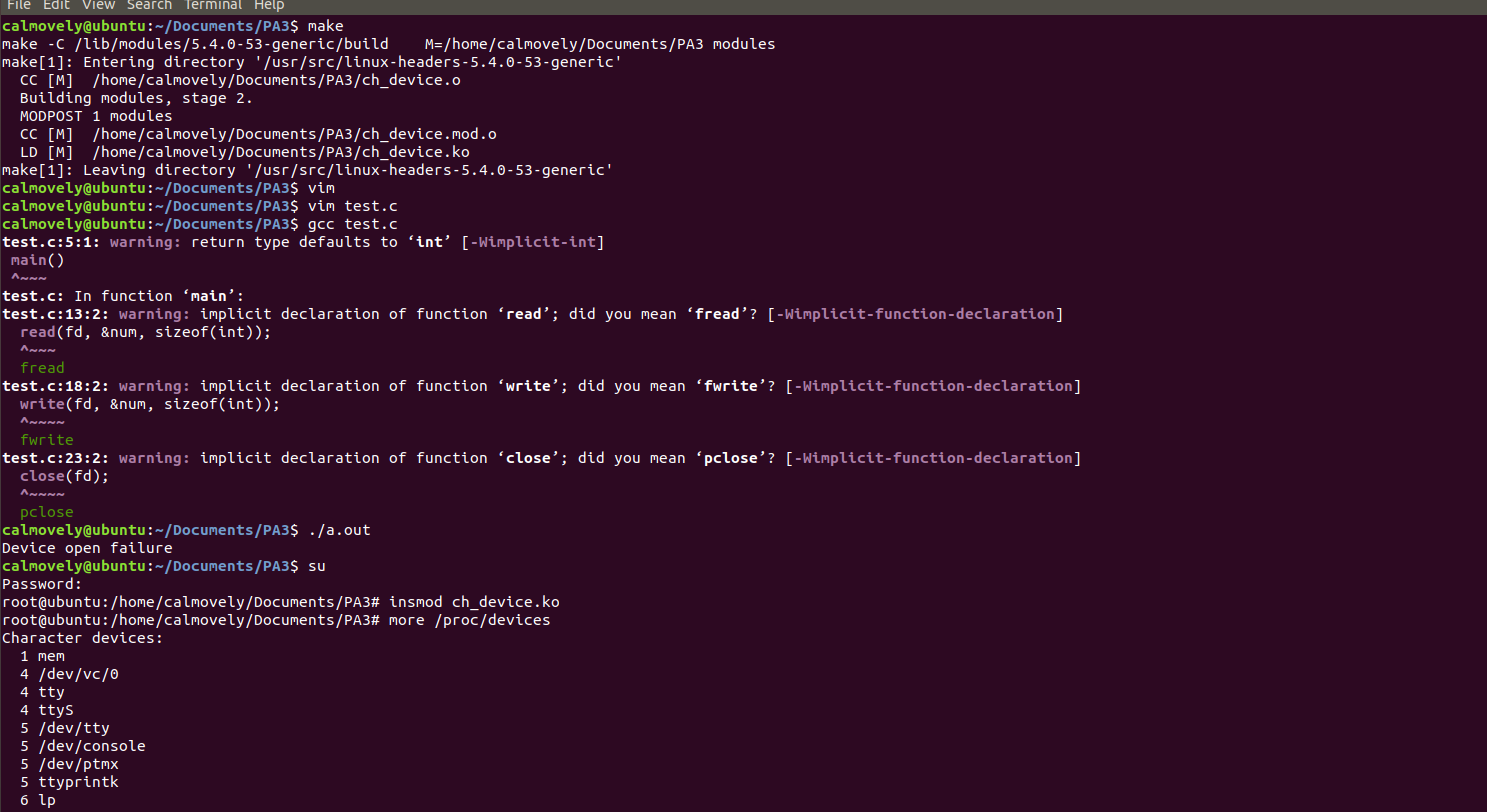
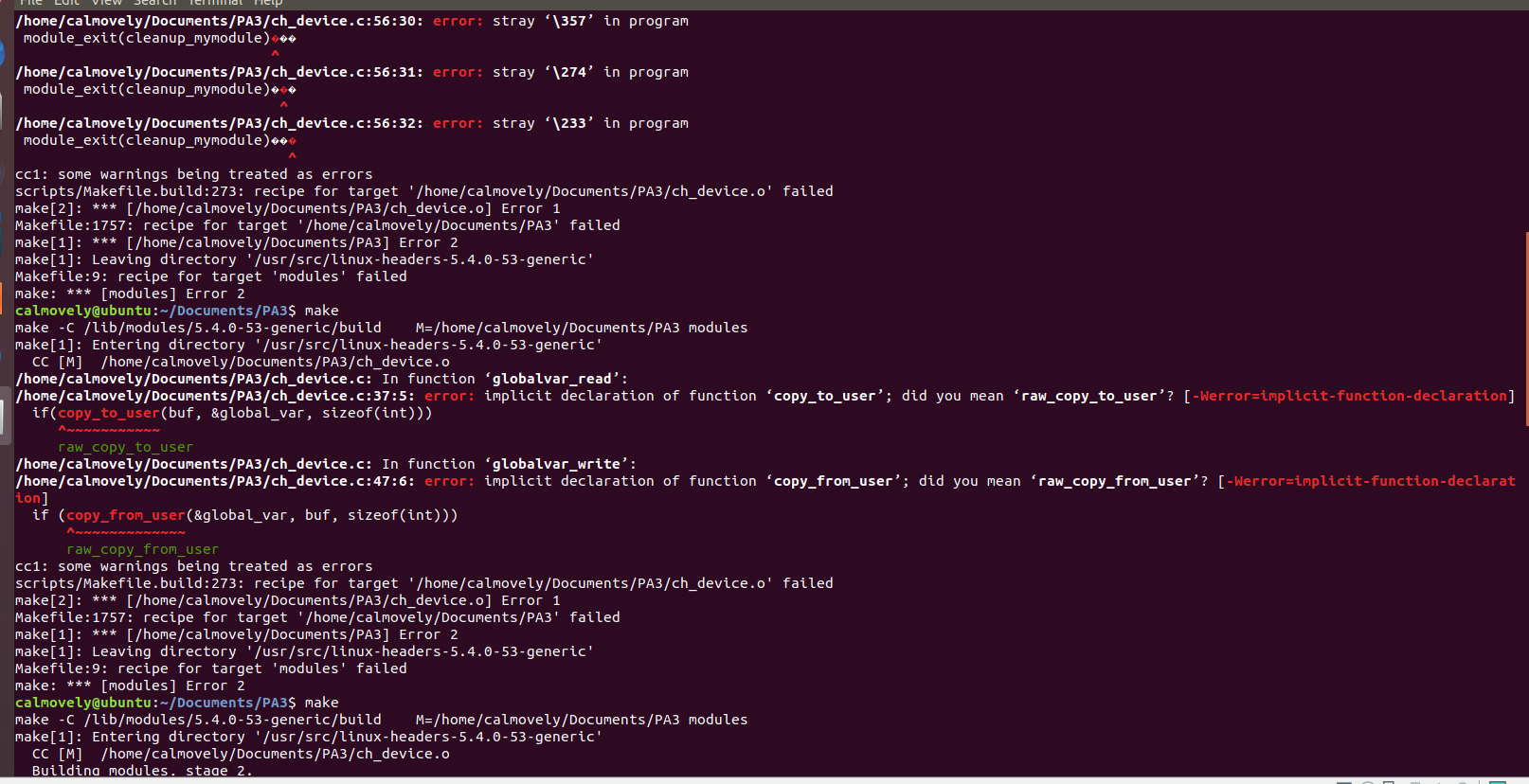
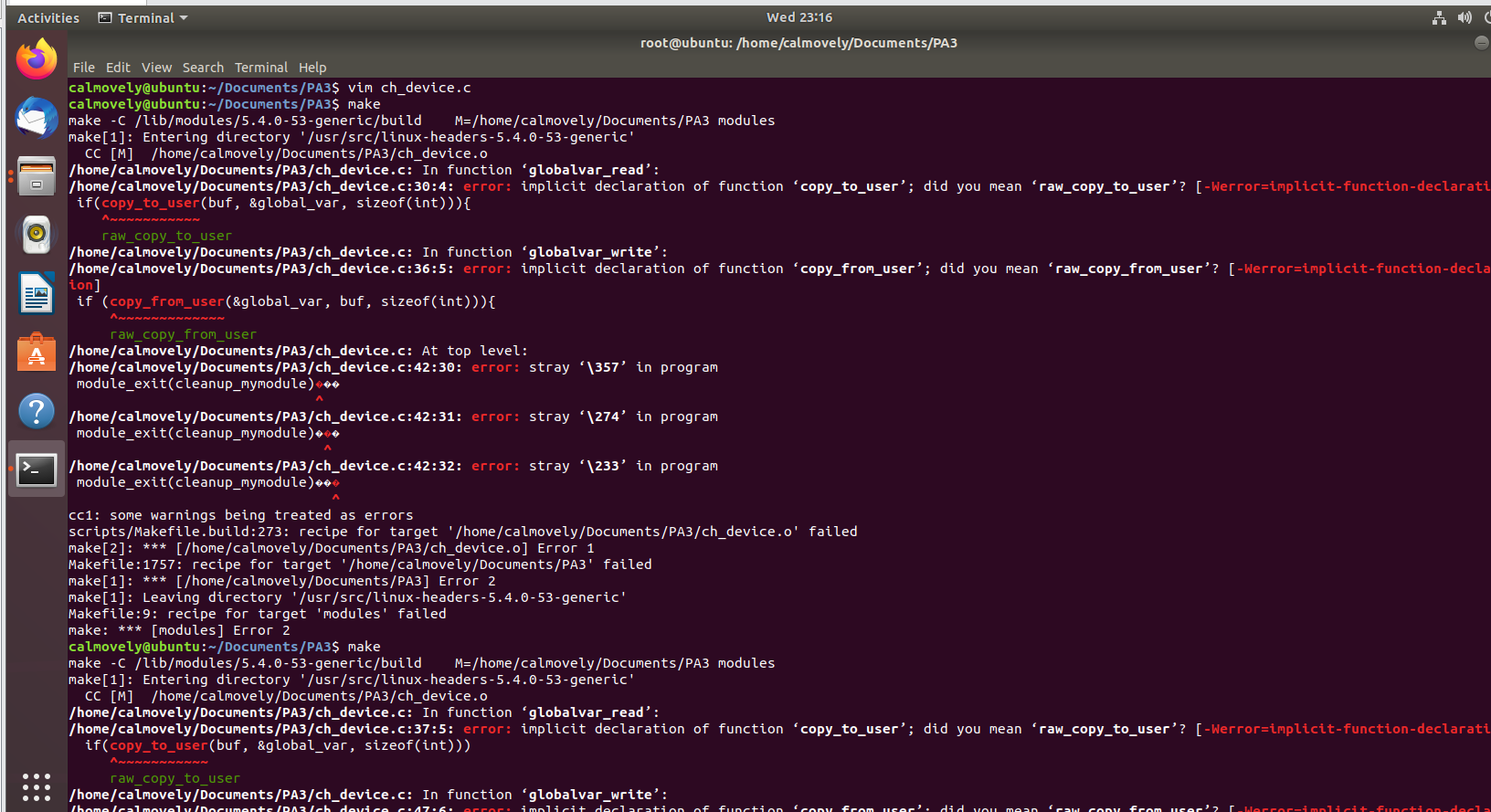
证明模块已经被移除。

接下来以同样的方式判断测试用例：

得到结果



**设备驱动**



开始时以为有了哪些错误，后来发现是老师的代码版本太老，有的需要更新（而且居然出现了中文字符），修改后就可以正常运行了。