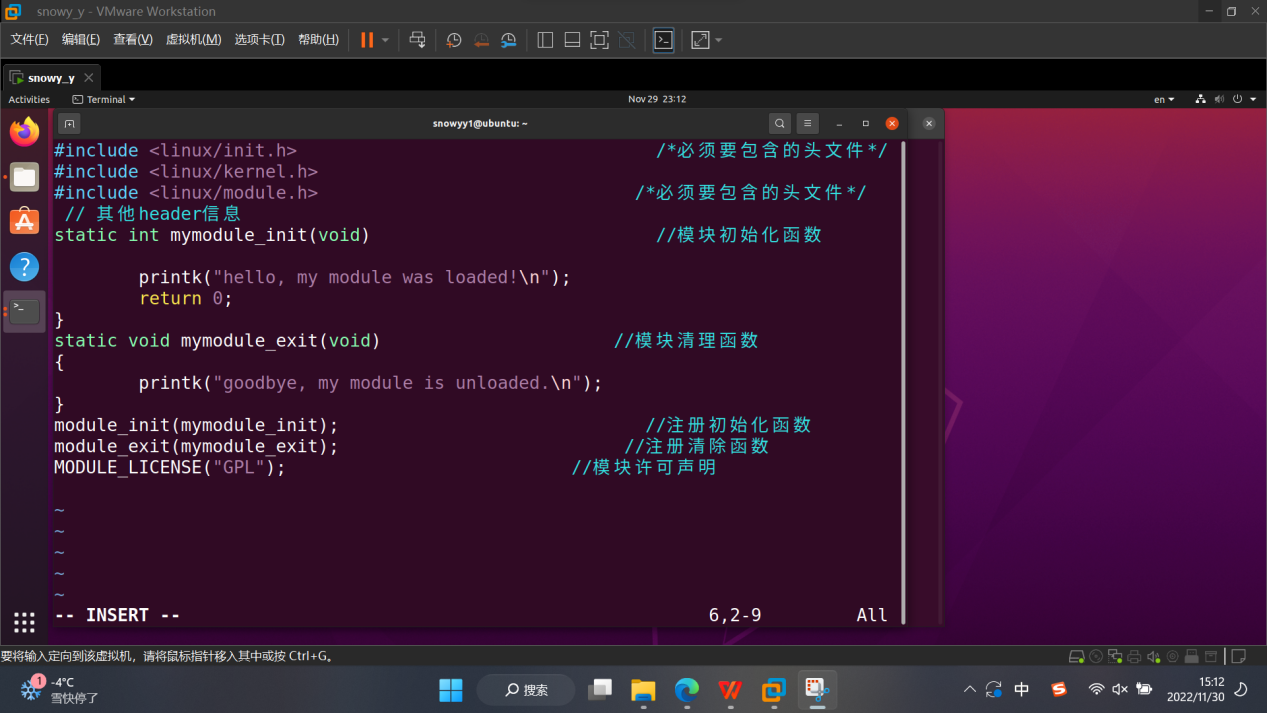
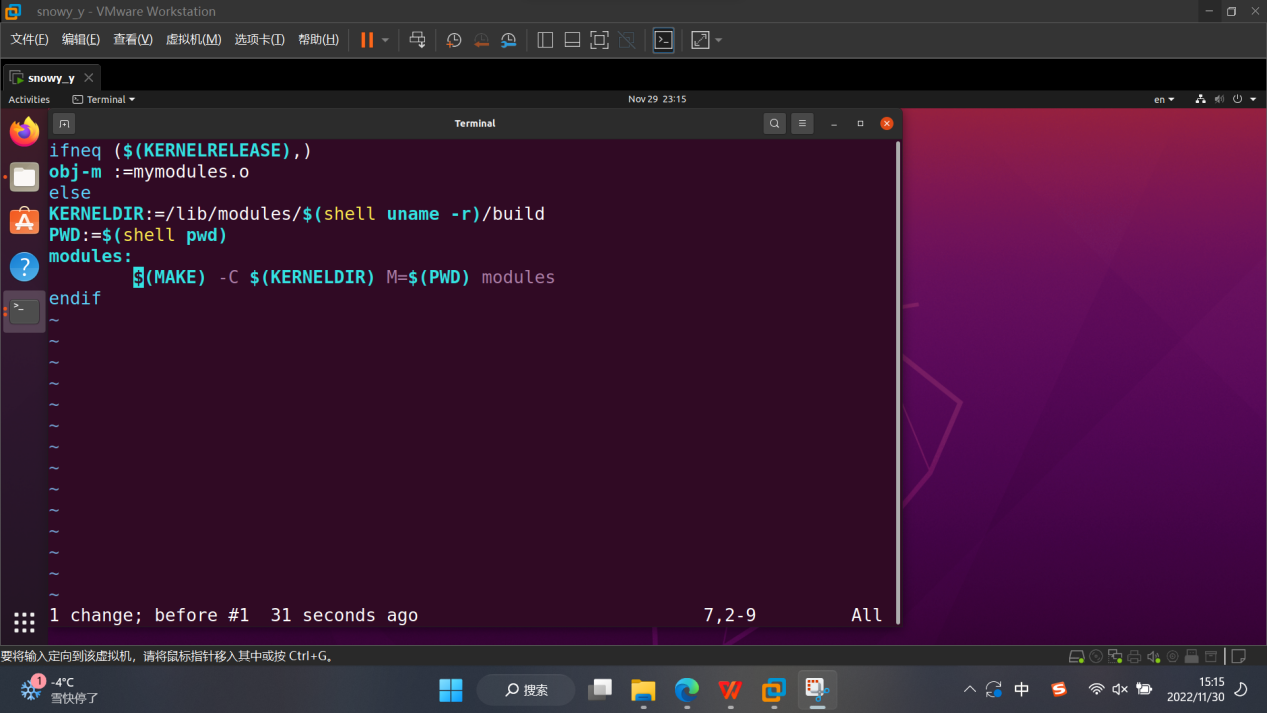
# 实验四：动态模块与设备驱动

## 1 动态模块

#### 步骤一：设计内核模块程序mymodules.c



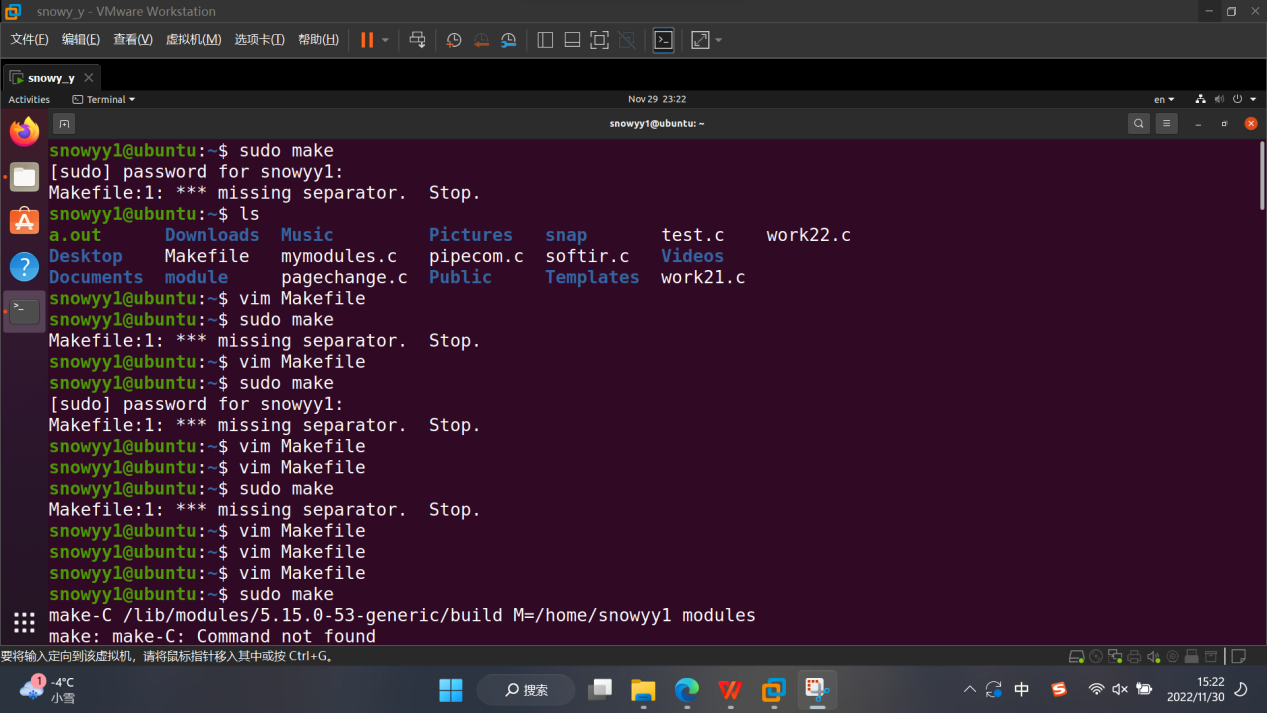
#### 步骤二：在相同文件夹下创建Makefile文件用来编译



Ps:问题：1.没有安装make，尝试在root权限下用apt-get install make失败；换源也失败；查找博客有人说是虚拟机没有联网的问题，尝试手动配置虚拟机网络，发现该电脑禁止共享网络；尝试修复注册表问题，下载了ccleaner，问题未解决。

解决：发现只给apt命令赋予权限，即（sudo apt-get install make）即可避开注册表等问题

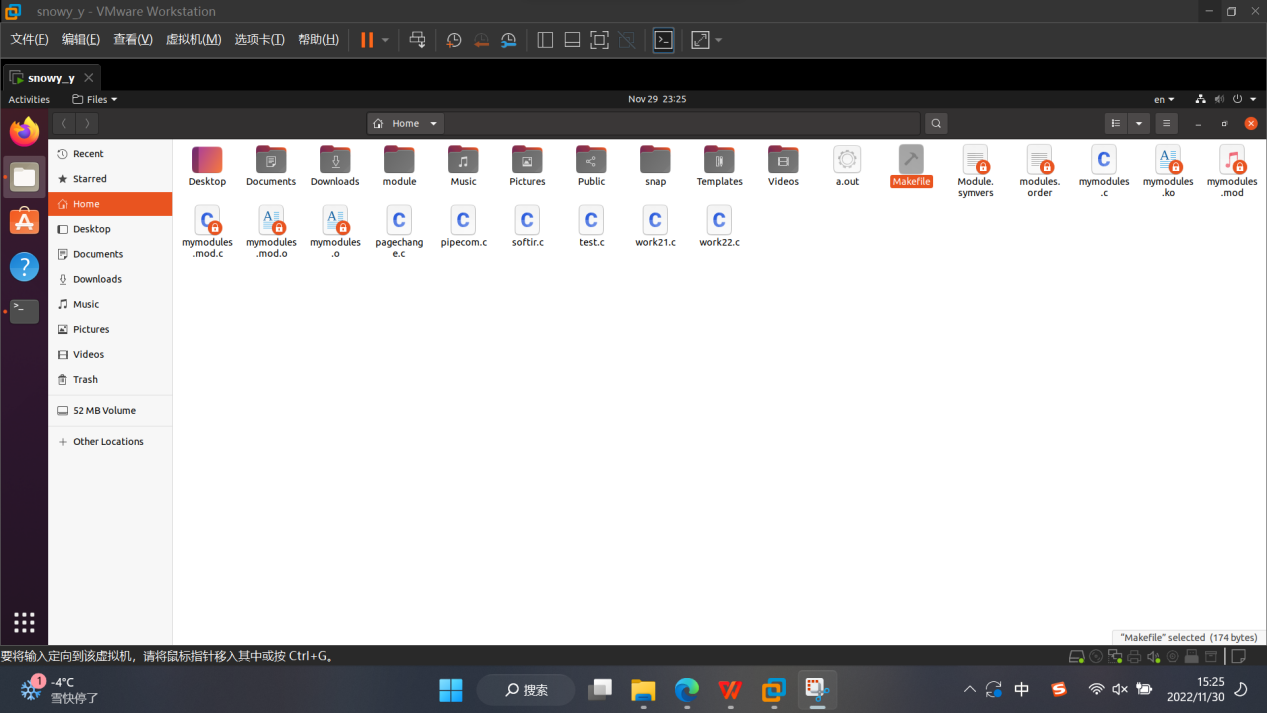
1. make命令执行失败，Makefile出现如下问题：



解决：通过博客了解到Makefile文件对格式敏感，只有命令行之前可以使用tab，且-m之前要空格（像终端中一样）

#### 步骤三：编译内核模块程序

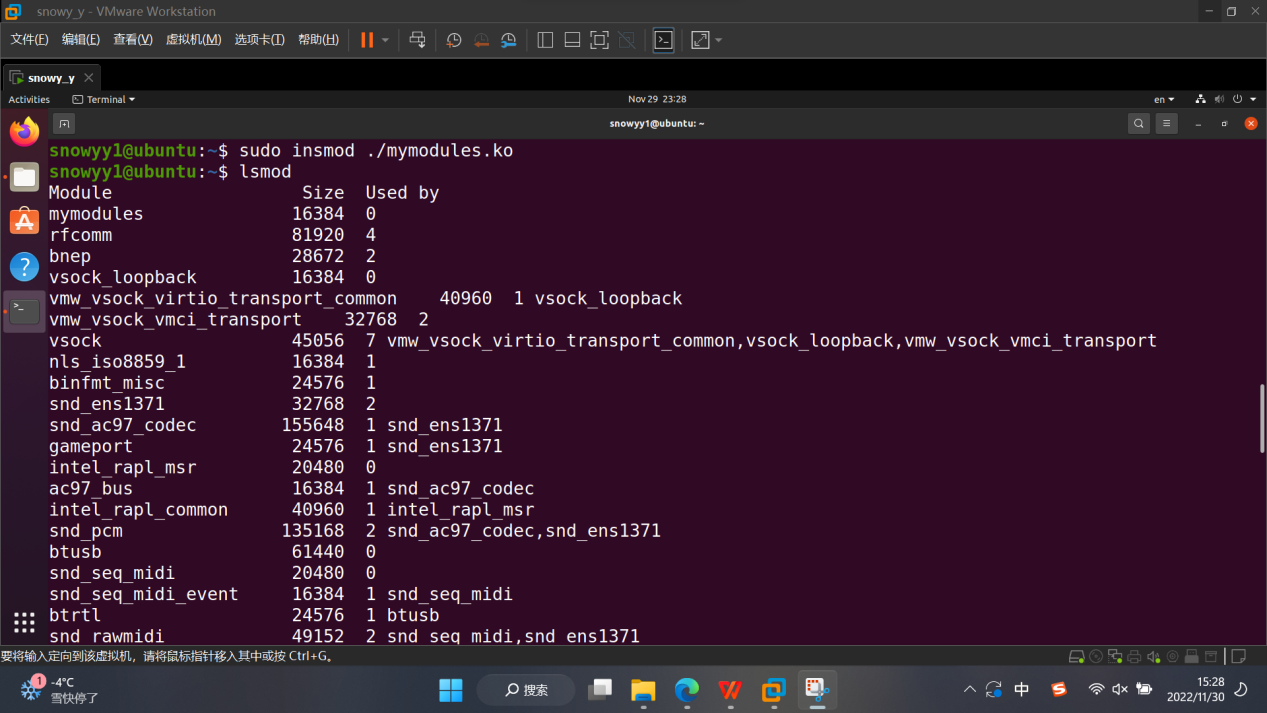
编译后结果如下：



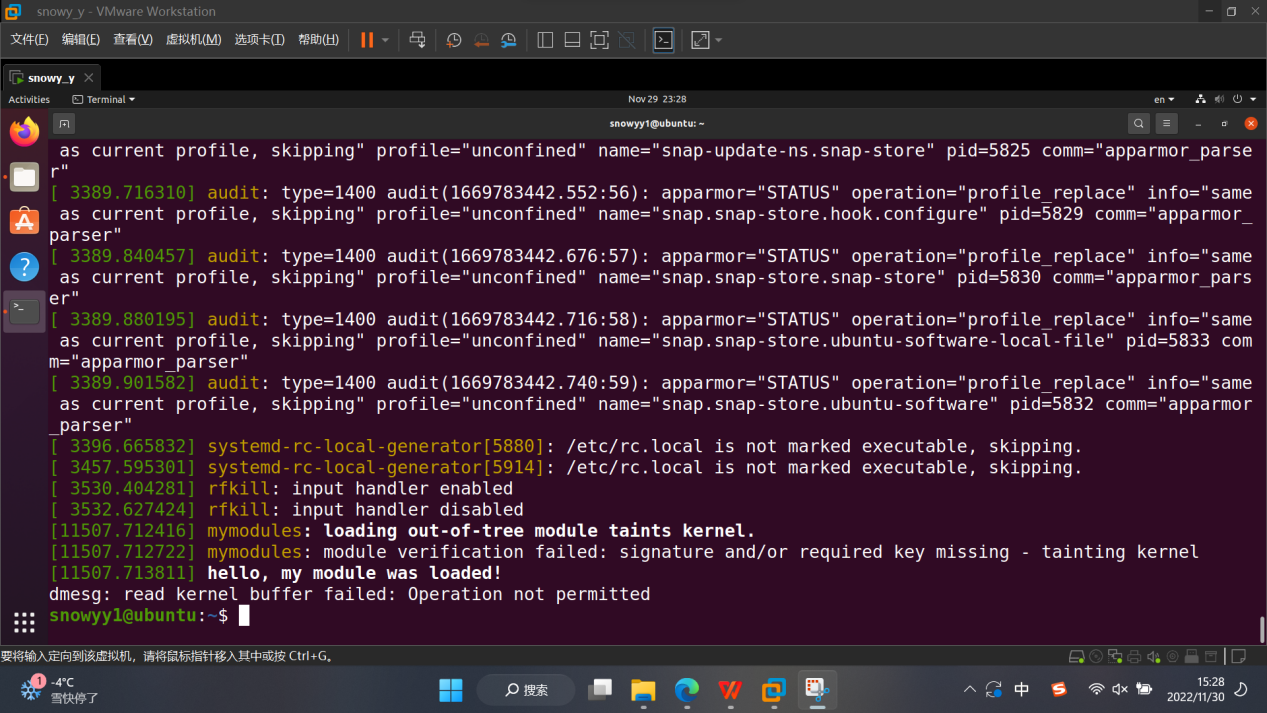
Ps:问题：出现编译错误

解决：函数返回值应是整形的没给返回值，另一处未加分号

用lsmod查看模块：



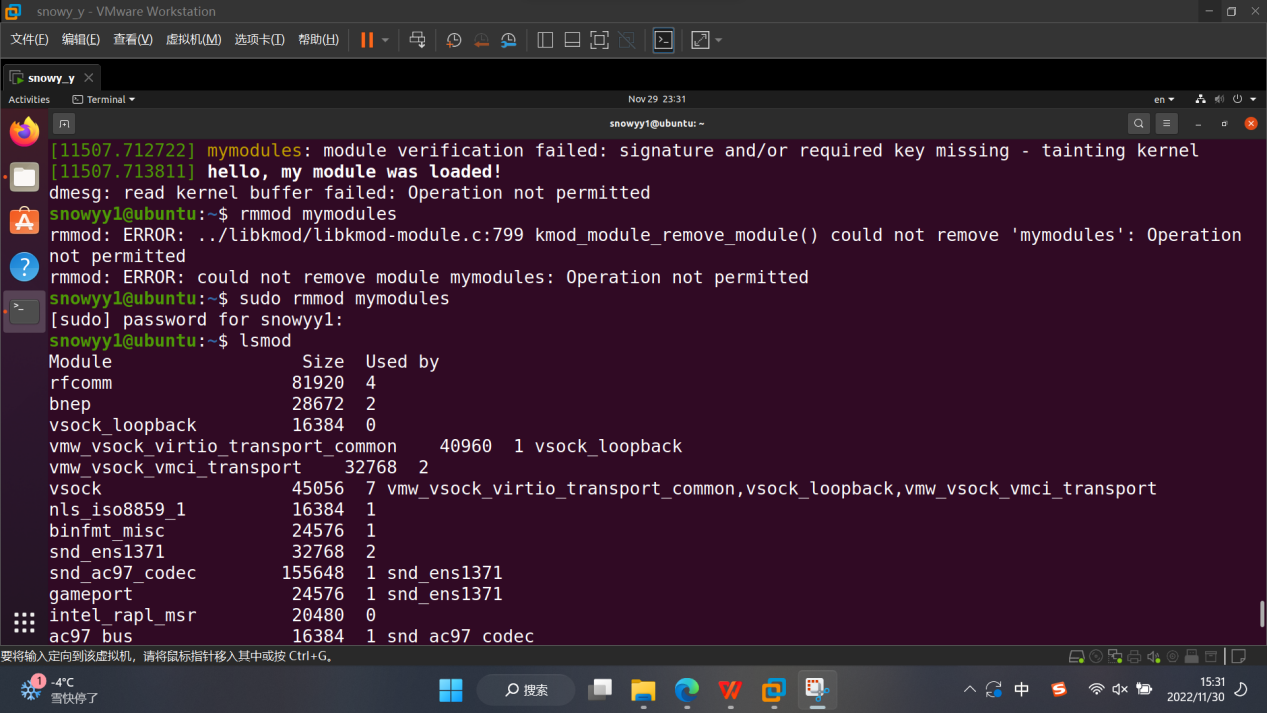
用dmesg -c查看日志：



#### 步骤五：卸载内核动态模块

使用命令：rmmod mymodules

可以看到mymodules已经被卸载



Ps:问题：出现operation not permitted

解决：使用sudo给管理员权限

## 2 字符驱动

#### 文件准备

文件结构：chardev文件夹下有chardev.c和chat.c和do.sh和del.sh和Makefile文件，编写细节见下：

步骤一：编写字符驱动文件chardev.c

以模板为基础

包含一个\_device\_data结构体用来设定设备数据内容

包含一个device\_read函数用来将键盘输入的字符存入buffer

包含一个device\_write函数用来将buffer中的内容输出到屏幕

包含一个device\_open函数用来打开设备

包含一个device\_release函数用来释放资源

写一个file\_operation来将函数与文件对应

包含一个设备初始化函数和设备结束函数

步骤二：编写编译文件Makefile

编写过程要注意语法和tab的使用，具体内容见附件



步骤三：安装和卸载字符设备程序

安装模块，并且创建是个设备文件，最后开启这些文件的读写权限

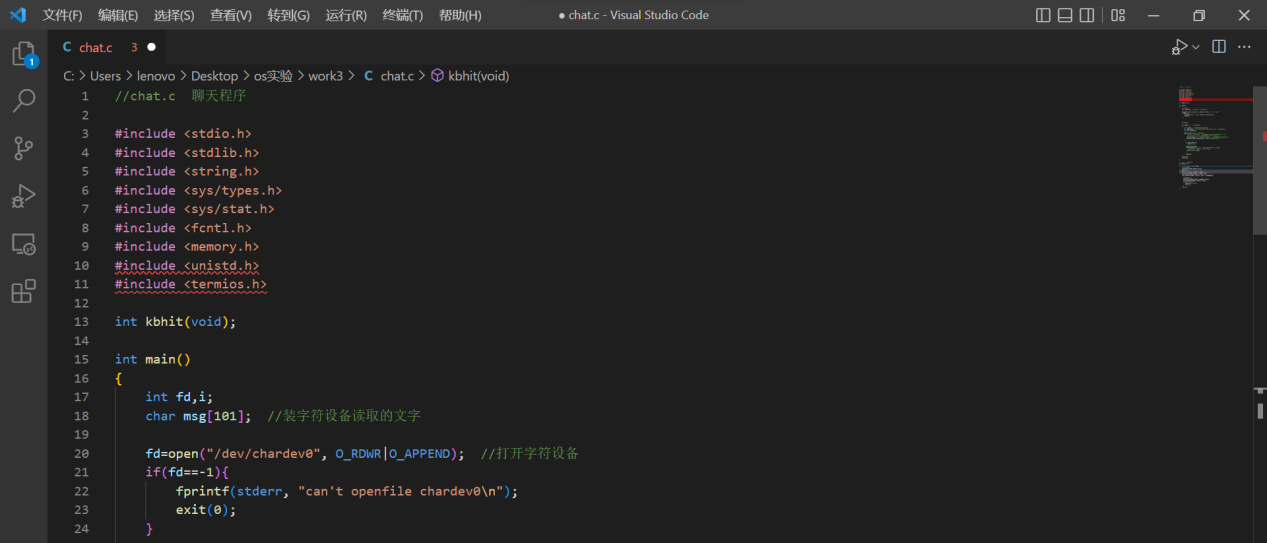


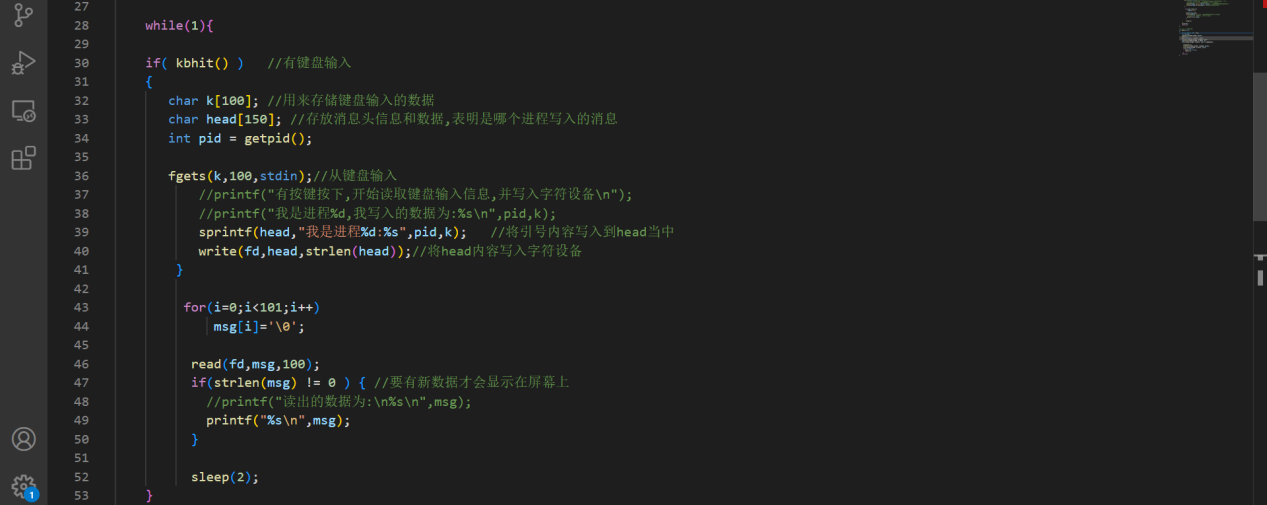
步骤四：编写聊天程序chat.c

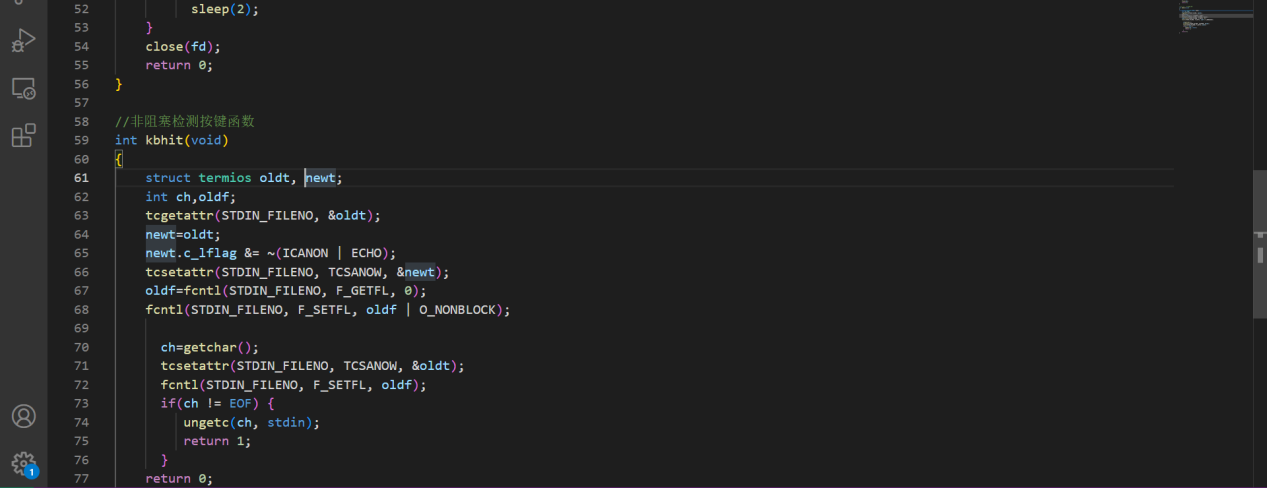
使用非阻塞控制方法，如果某用户的键盘有输入，其他用户就可以看到该用户的输入，并且在屏幕上输出是哪个进程输出了内容方便沟通。Ps:文件中从键盘读取内容使用了fgets()函数，对head内容输如使用了sprintf()函数

错误：刚开始会出现其他用户输出总是附带部分以前用户的输出的问题

解决：每次一个用户开始输出之前，总是将中间数组msg[101]清零







#### 2.编译安装过程

将makefile和要编译的文件放在同一个目录下，在终端输入“make”命令，编译字符驱动，生成编译文件。

给do.sh和del.sh特权。命令是：chmod +x do.sh 和 chmod +x del.sh

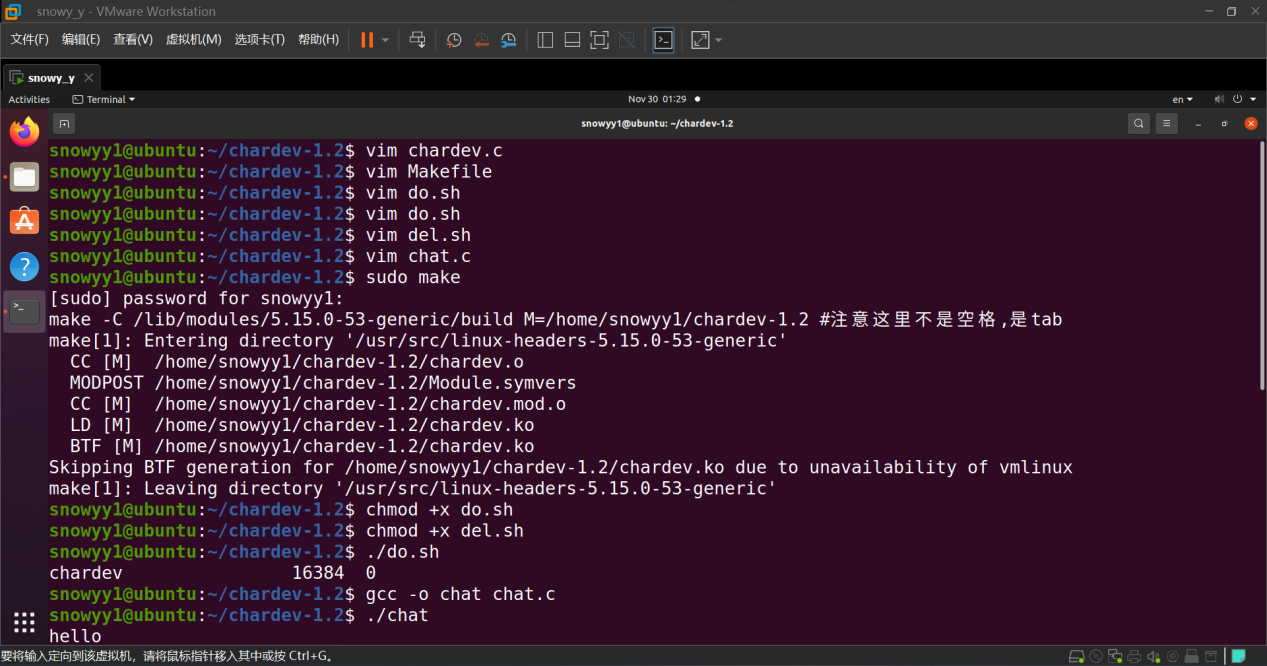
运行do.sh安装字符驱动 方法：./do.sh

编译chat.c文件，方法：gcc -o chat chat.c

运行三次chat程序，创建三个用户窗口来做test。方法：./chat

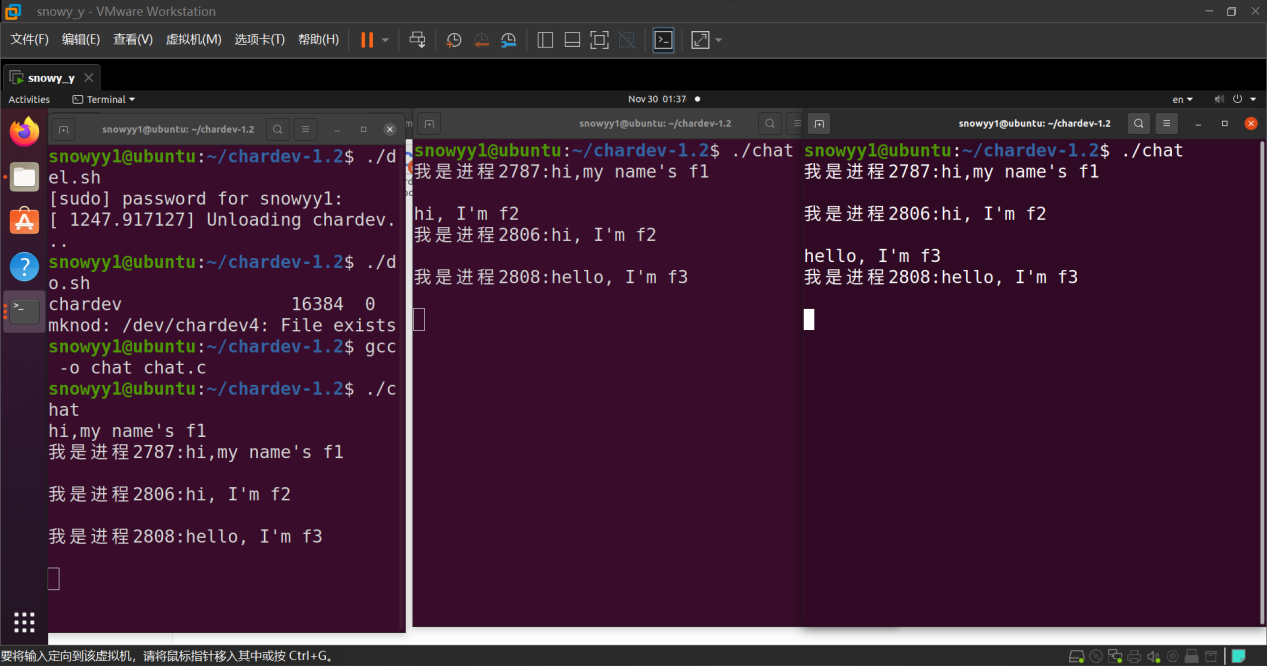
最后./del.sh卸载字符驱动

在代码中用符号位实现了读写进程的互斥



#### 聊天功能展示

（私聊功能在后面展示）创建三个user广播通信如下：



进一步实现私聊功能：实现思路是在判断键盘输入函数将键盘输入读取后，判断第一个字符是否是@，如果是就将后面直到’ ’前的数字用atoi()函数转变为整形，再与要读取的进程号进行对比，如果不是，就不输出在屏幕上，如果是就输出在屏幕上，从而实现私聊。

