**《操作系统原理》实验报告一**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 姓名 | 王粟鹏 | 学号 | U201817044 | 专业班级 | 软工1803 | 时间 | 2020.3.19 |

1. **实验目的**

1）理解操作系统生成的概念和过程；

2）理解操作系统两类用户界面（操作界面，系统调用）概念；

1. **实验内容**
2. 在Ubuntu或其他Linux平台环境下裁剪和编译Linux内核，并启用新内核。
3. 在Ubuntu或其他Linux平台为Linux内核增加1-3个新的系统调用，并启用新的内核，编写一个应用程序测试新增加的系统调用是否能正确工作。
4. 在Windows环境写，编写一个批处理程序（算命大师.bat），程序运行后，输入：出生年月日（例如 2000-07-31）.系统输出相应的属相和星座，例如：你属兔，狮子座。要求：输入进行合法性检查，能循环接受用户的输入，直到输入q或Q就退出。
5. **实验过程**

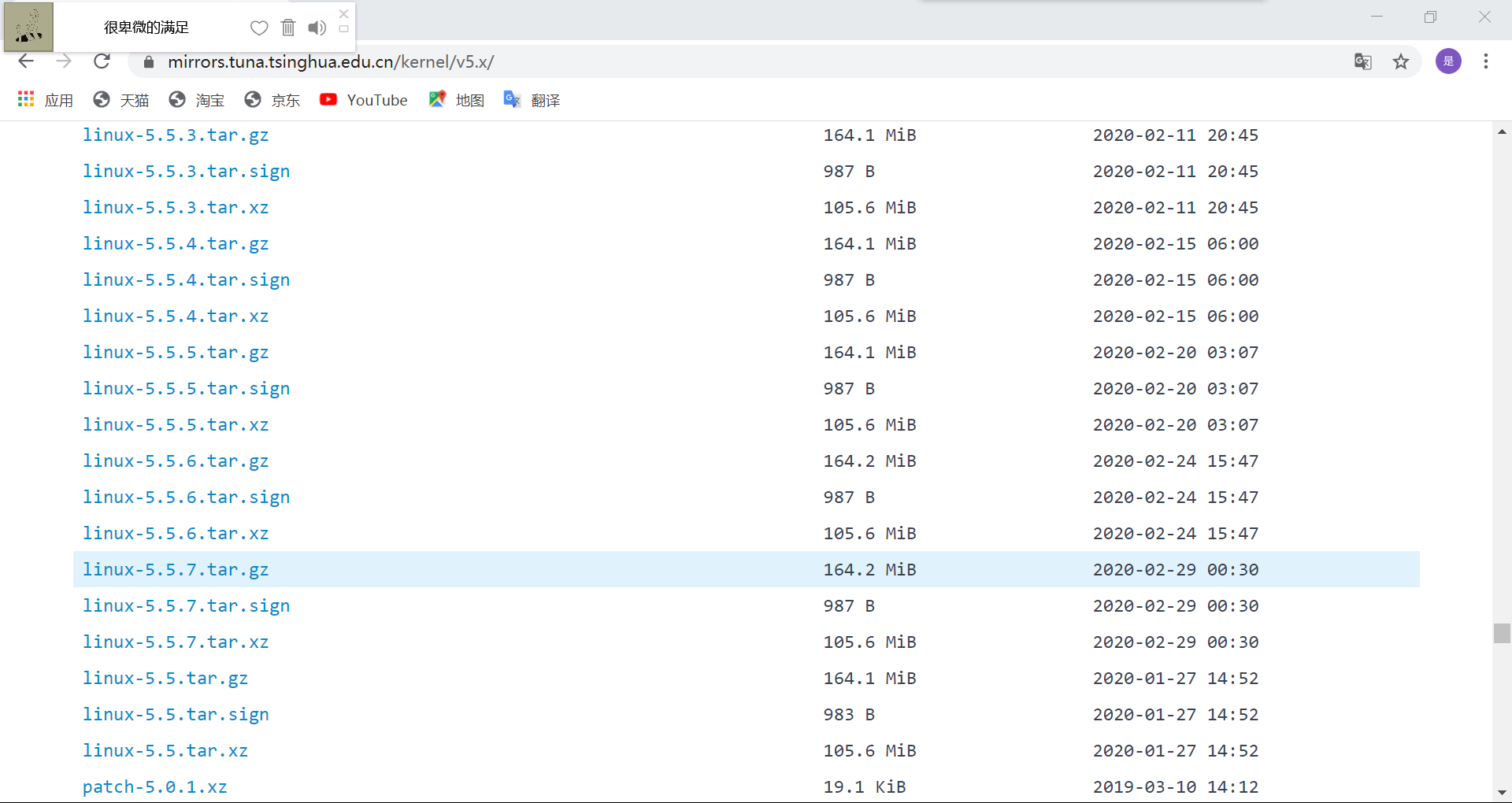
**（一）实验步骤**

1. **编译Linux新内核：**

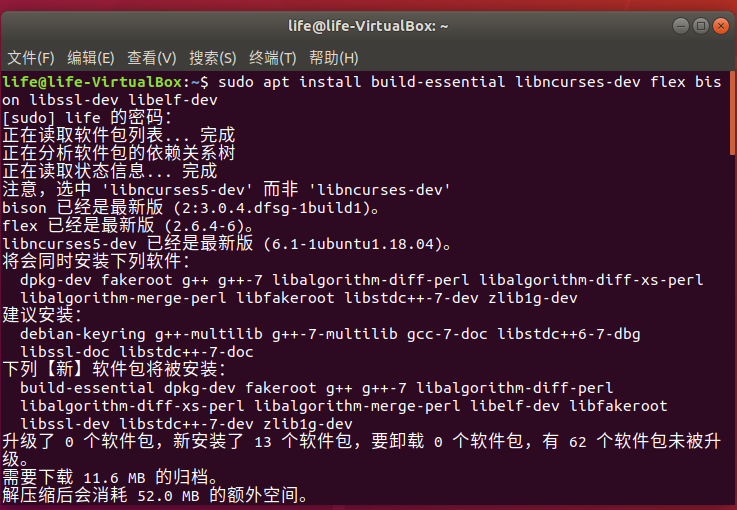
操作环境：Ubuntu系统，内核为Linux-3.8.0，编译的内核版本为Linux-5.5.7

开发过程：

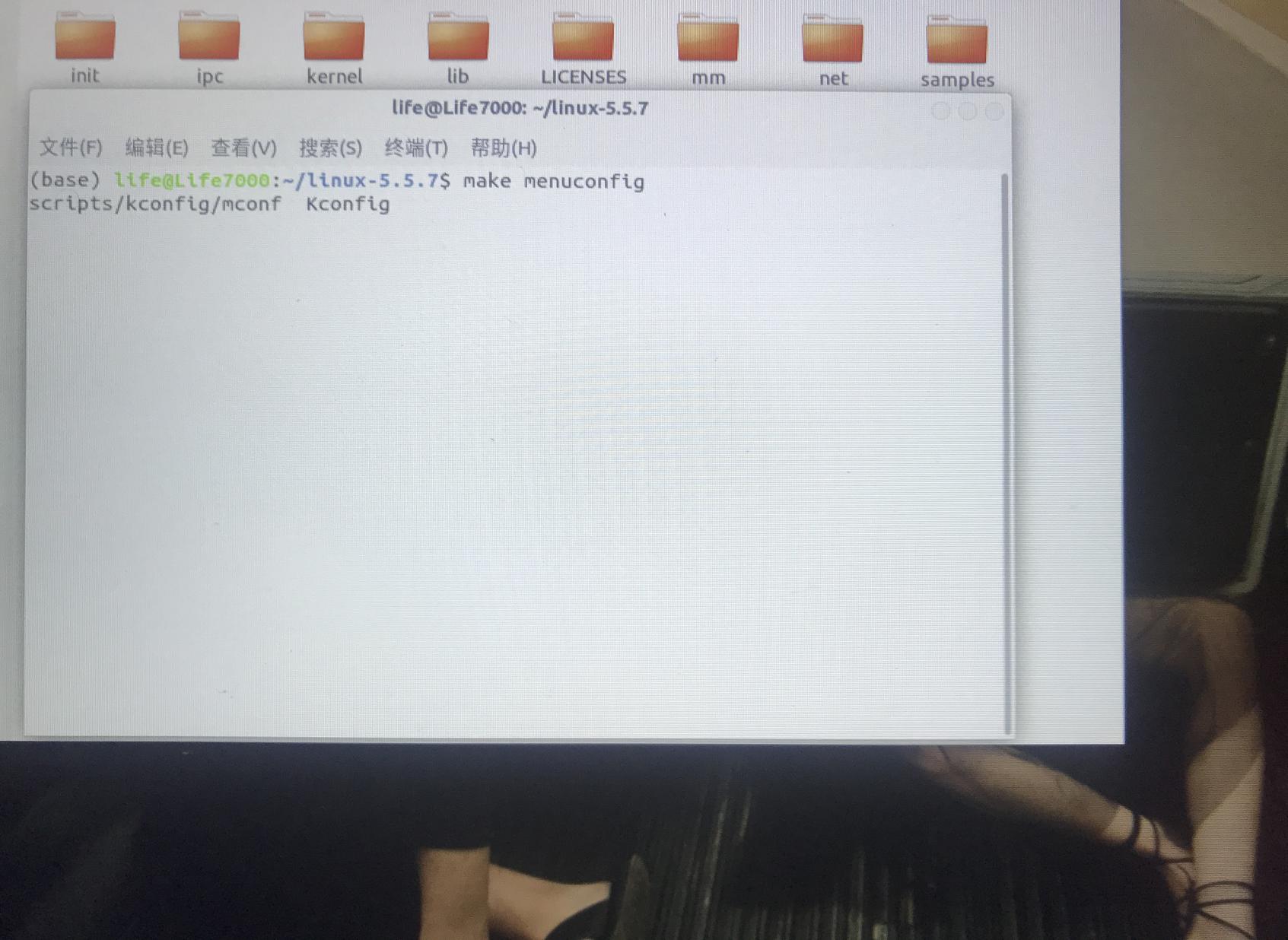
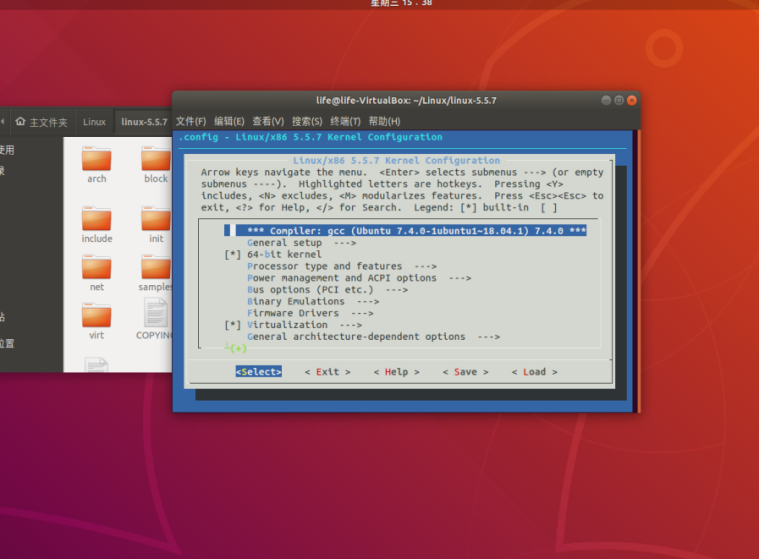
1. 获取Linux-5.5.7内核的源代码压缩包，解压到本地。



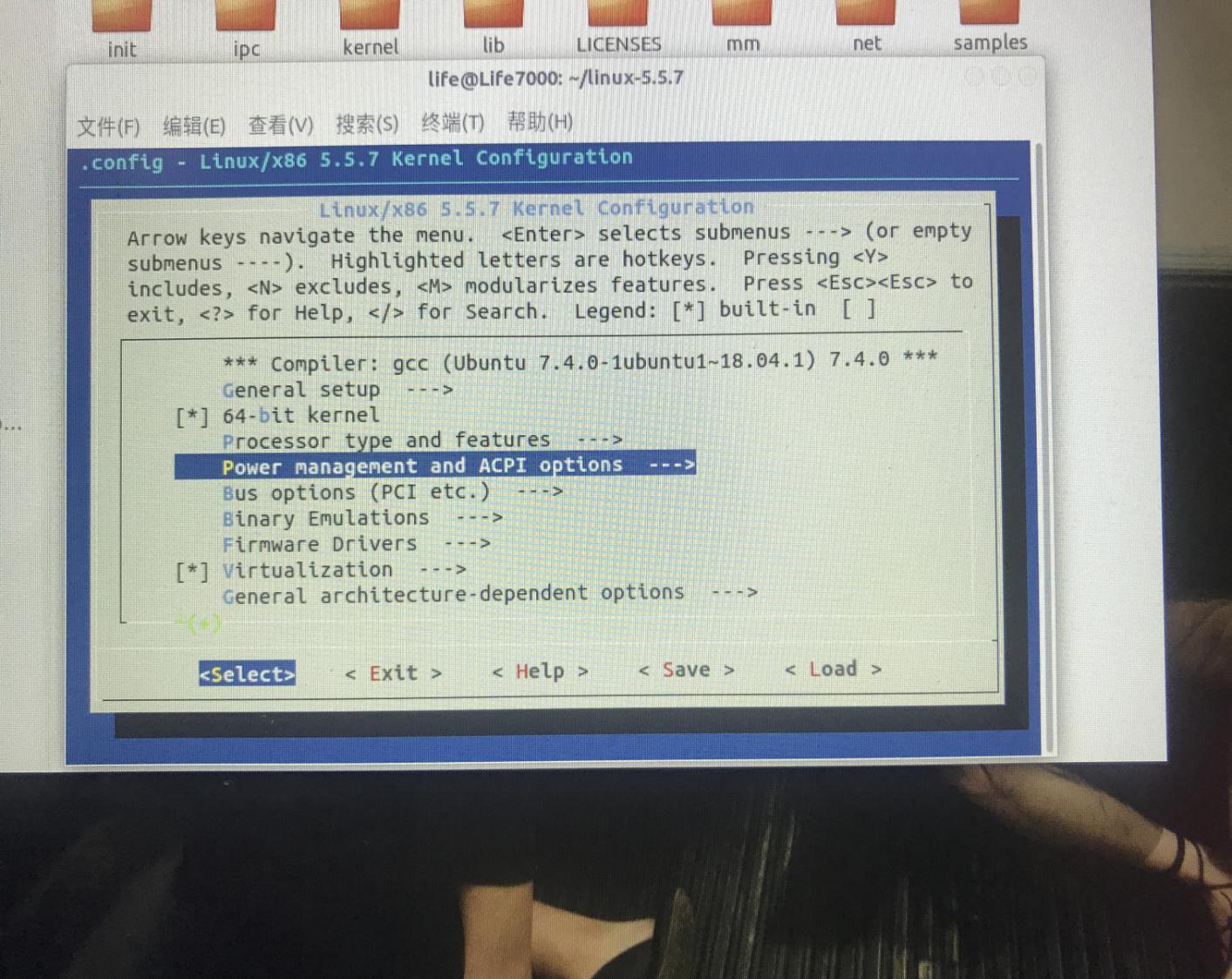
1. 安装编译内核所需要的依赖



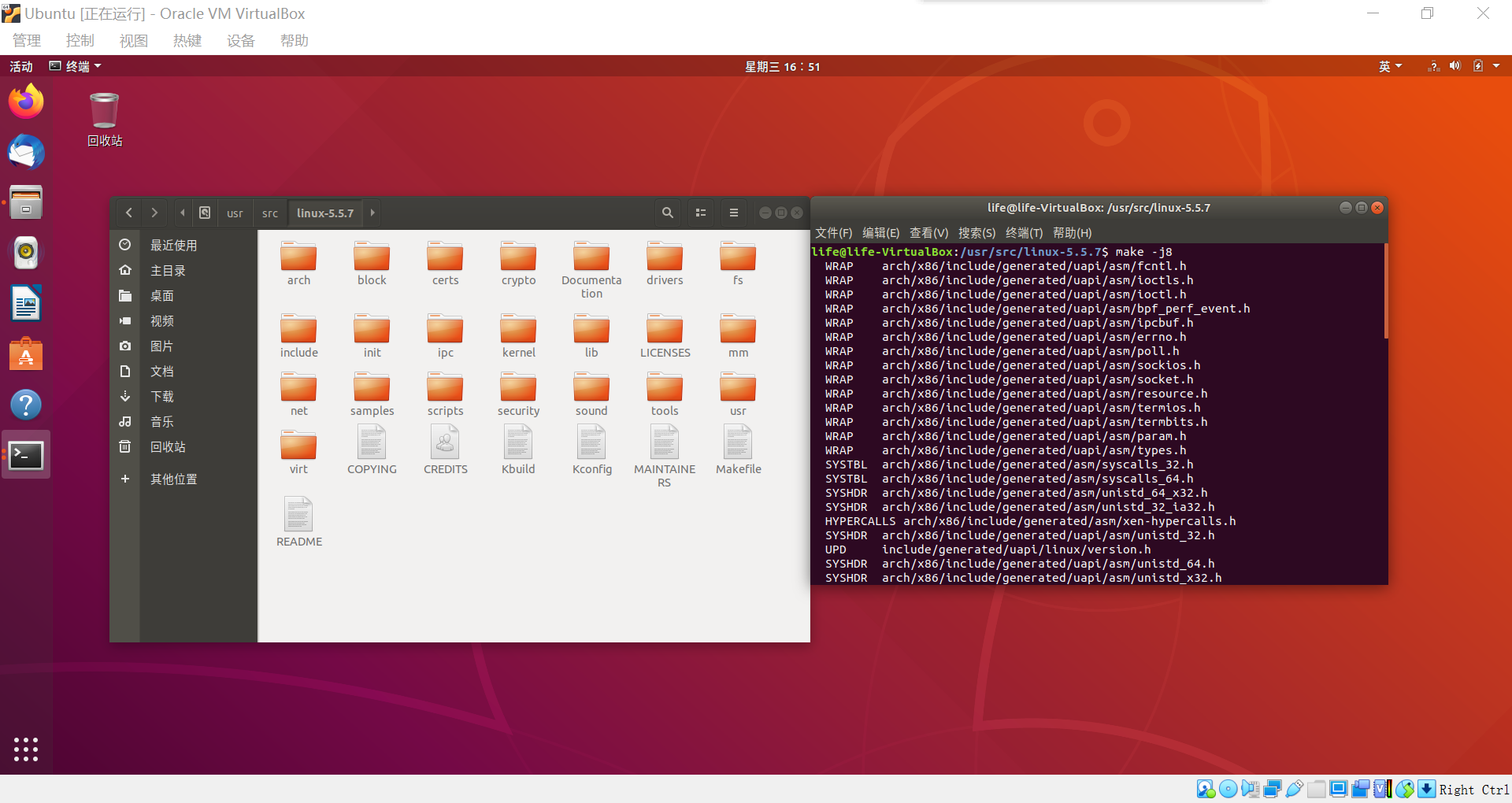
1. 启动内核配置程序，选择使用指令 make menuconfig。

1. 根据需要对内核进行配置。



1. 编译新的内核



1. 编译和安装模块：

依次使用以下指令，对模块进行编译和安装

$make modules $sudo make modules\_install

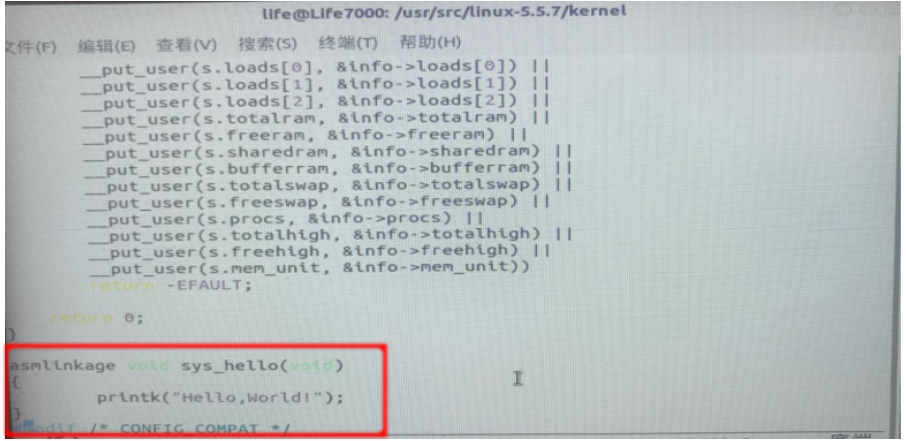
$sudo make install $sudo make header\_install

1. 重新启动启动新的内核。
2. **增加系统调用:**

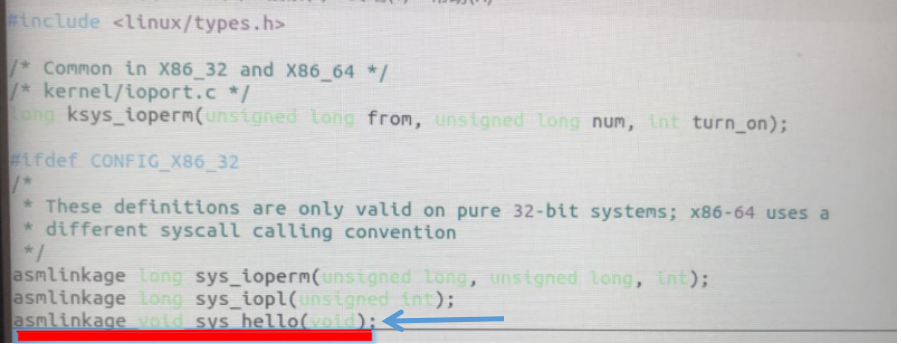
操作环境：Ubuntu系统，内核为Linux-3.8.0，编译的内核版本为Linux-5.5.7

操作过程：

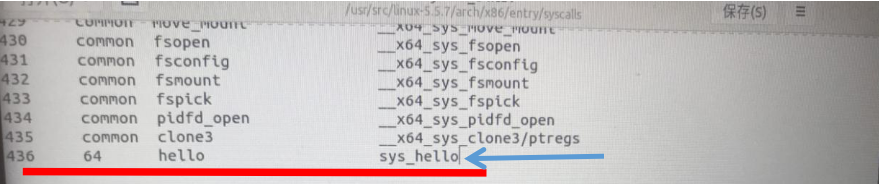
1. 编写系统调用函数。找到内核文件下的Kernel文件夹，打开里面的sys.c文件，在里面添加自己编写的系统调用函数，在这里，我编写的是调用该系统调用会向内核缓冲区输出“Hello,World!”。



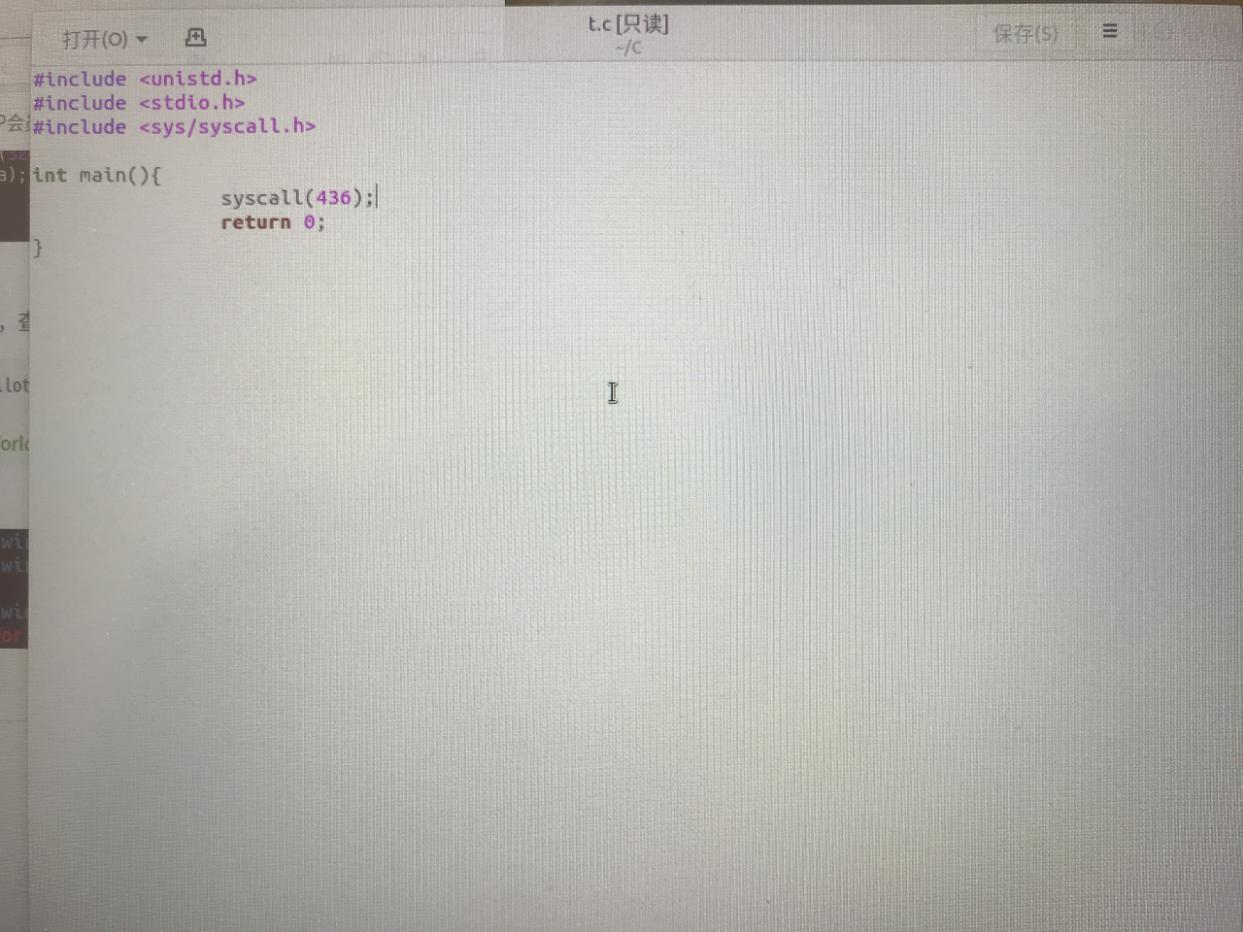
1. 在系统调用目录当中，添加该系统调用的声明。进入目录./arch/x86/include/ asm/syscalls.h，在相应位置添加该系统调用的声明。



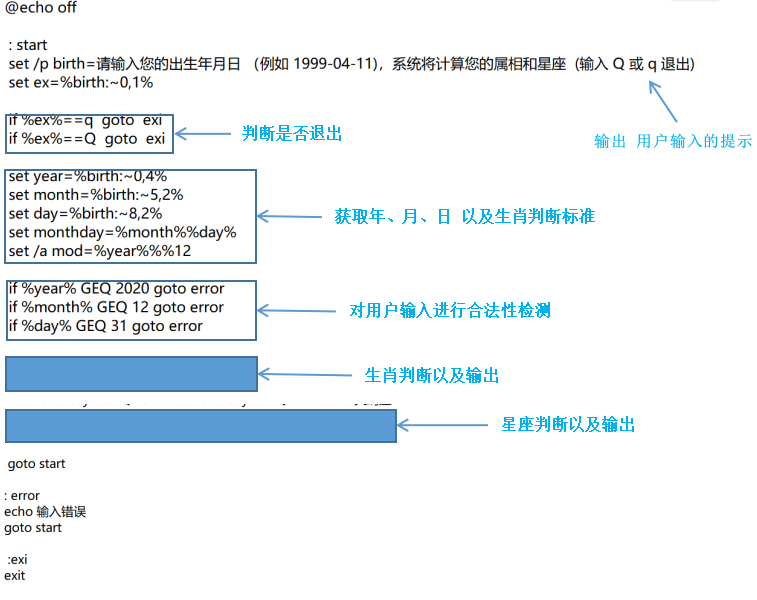
1. 添加系统调用的编号。打开文件./arch/x86/entry/syscalls/syscall\_64.tbl，在相应位置为系统调用添加调用编号。



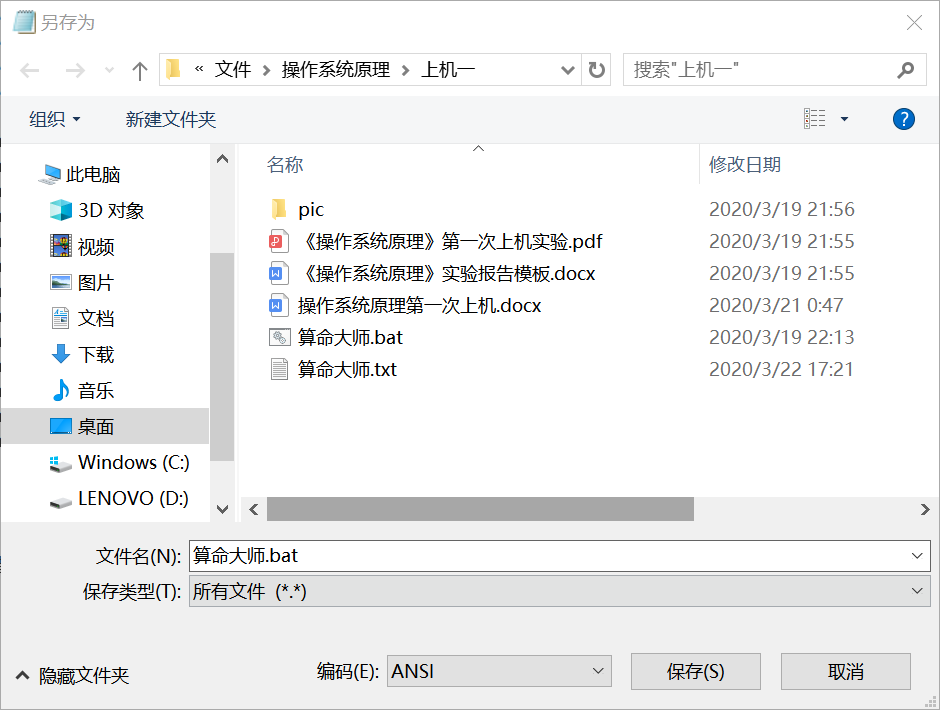
1. 编译内核。具体步骤如上。
2. 编写函数，对系统调用进行检测，并捕获输出结果。



1. **算命大师.bat:**
2. 新建一个txt文件，在该txt文件中写该程序的代码。代码如下：



1. 写完之后将程序保存为算命大师.bat，保存类型为所有文件，编码为ANSI。



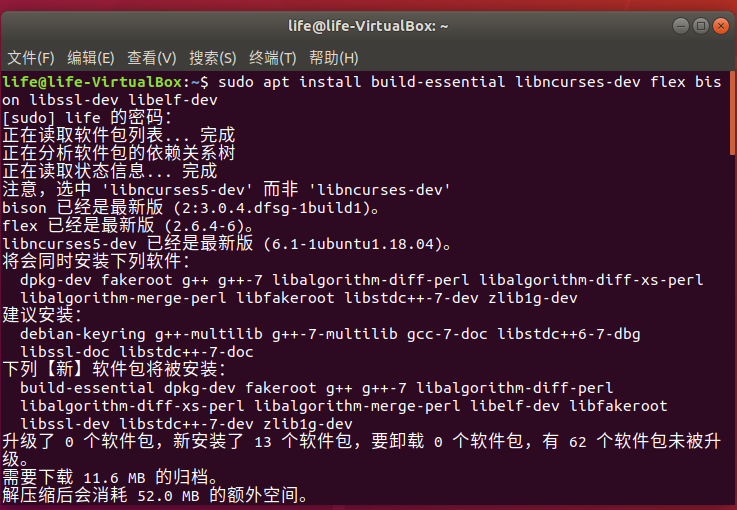
1. 测试程序。
2. **解决错误和优化**
3. **编译Linux内核**
4. 类型：环境配置错误。

原因：缺乏必要的环境依赖

现象：导致无法进行内核编译。

解决办法：安装相关的依赖。

建议：在编译内核之前，可以提前安装好这些依赖，可以提高效率。

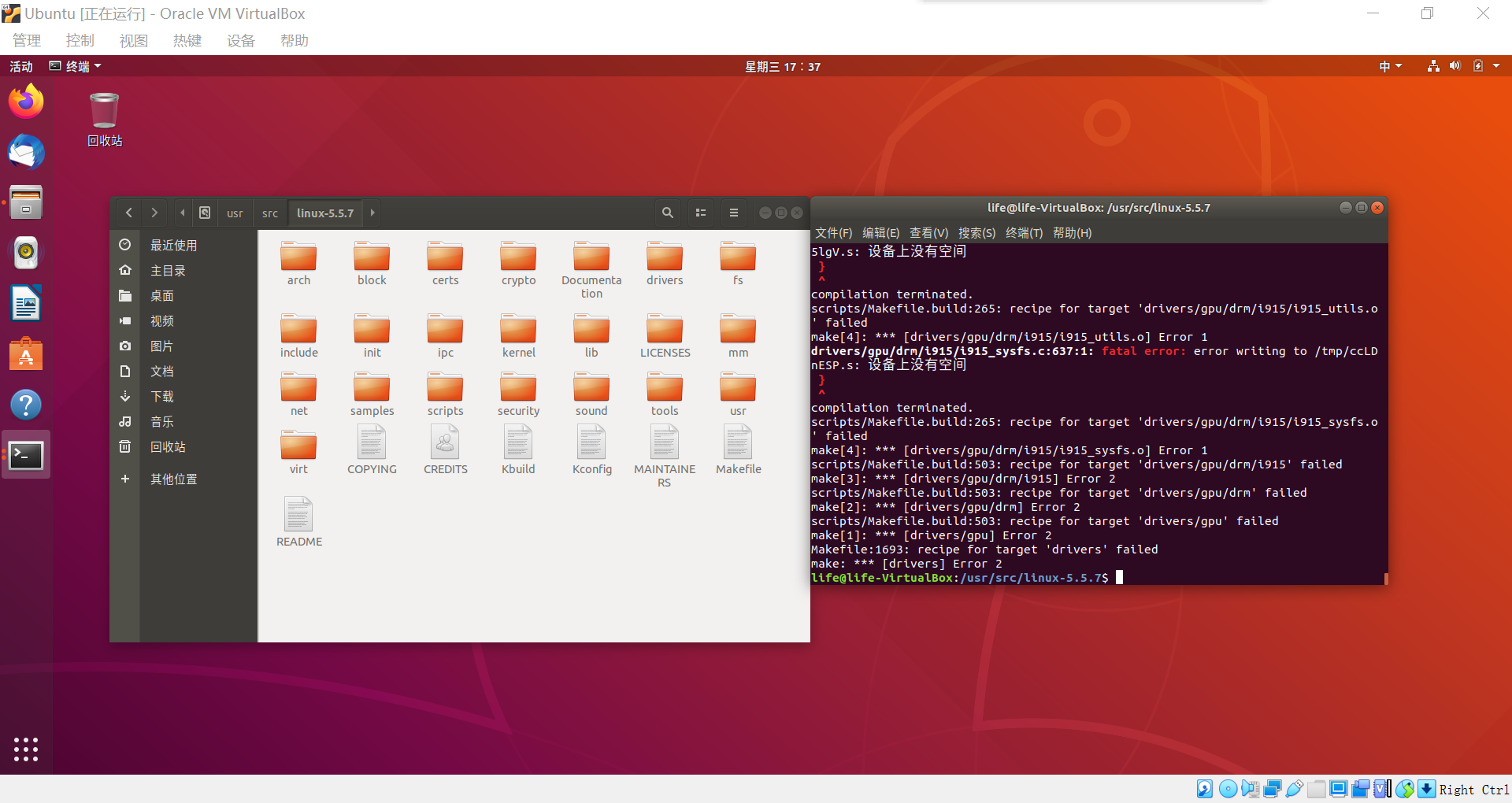


1. 类型：运行错误

原因：内存不足

现象：无法完成内核编译

解决办法： 给虚拟机分配更多的磁盘空间。在这里我使用的解决办法是直接到实体机上编译新内核，因为实体机上我之前分配的内存是足够的。在实体机上编译之后，最终成功了。



1. **增加系统调用**
2. 类型：特殊的语法错误

原因：在加入到系统调用目录时，应该是sys\_hello，我输入了sys.hello

现象：无法成功的将自己的系统调用添加到内核当中

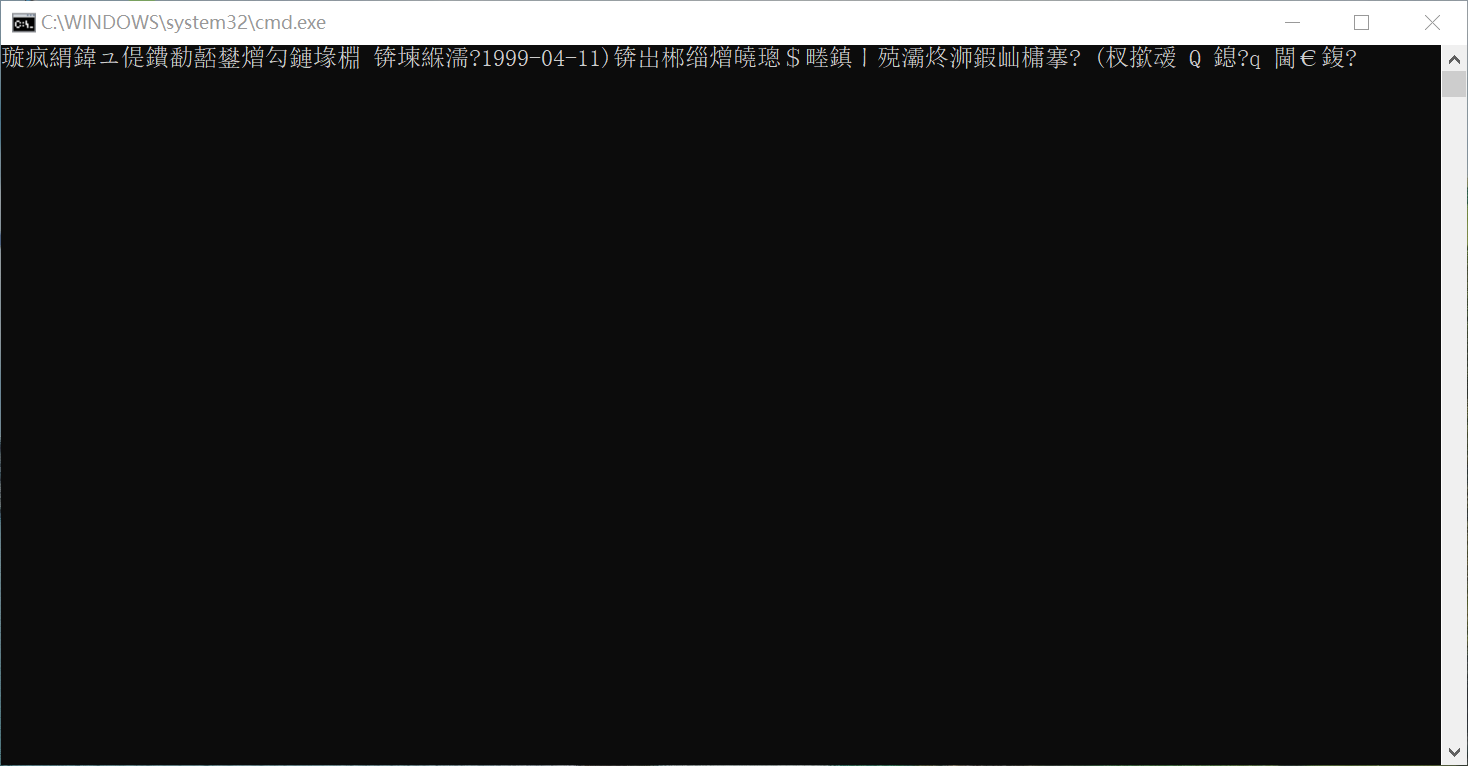
解决方法： 改正之后，在重新编译内核。

建议：在编译内核之前，一定要仔细检查每一步是否正确。

1. **算命大师.bat**
2. 类型：操作过程错误

原因：文件保存时，编码选择的UTF-8

现象：汉语输出部分输出为乱码。



解决方法：文件保存时，编码方式选择ANSI

1. 类型：运行错误

原因：在编写代码时，使用了未声明的变量

现象：程序报错

解决方法：找到未声明的变量，对其进行声明和定义

1. 类型：特殊的语法错误

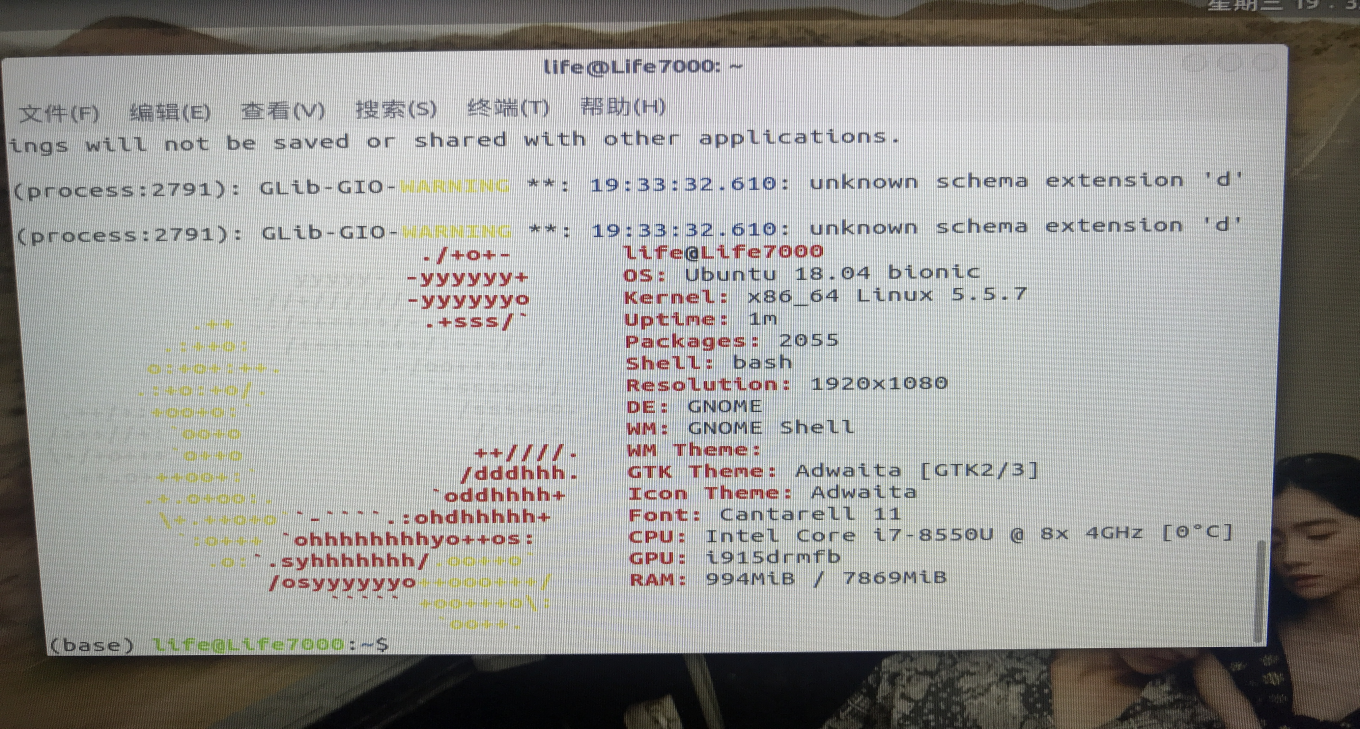
原因：在输入q或Q的判断时，不小心多添加了一个空格

现象：只有输入” Q”或者” q”时，程序才会退出

解决方法：根据测试时出现错误的类型，找到是因为多了一个空格的缘故，然后删除掉这个空格。

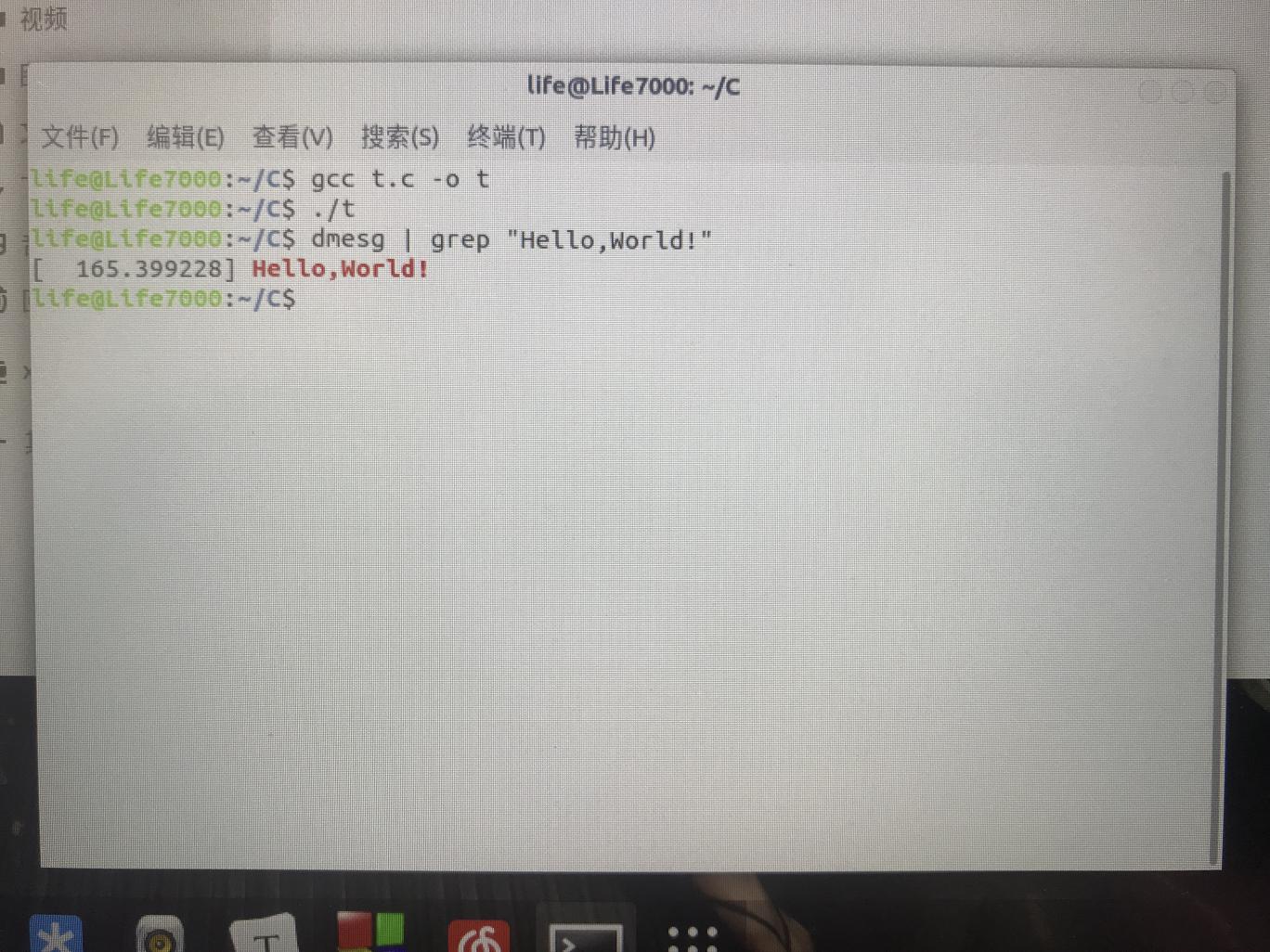
1. **实验结果**
2. **编译Linux内核**

重新启动之后，默认会自动启动新编译的内核。也可以进入grub，在高级选项中选择运行的内核版本，进入系统之后，使用指令查看当前内核版本，发现确实是自己刚刚编译过的版本。



1. **增加系统调用**

调用编写的检测程序，使用 dmesg捕获内核缓冲区的内容，可以看到，在调用了系统调用之后，确实向缓冲区输出了“Hello,World!”，因此编写的系统调用成功添加到了内核当中。



1. **算命大师.bat**

可以对用户输入进行正确的判断，输出正确的结果；可以对对用户输入的合法性进行检测；程序实现循环进行；输入q或Q可以退出。



1. **体会**
2. 学会了如何根据自己的需要和硬件环境配置Linux内核，能够处理一些在编译过程中出现的错误。除此之外，在进行一个任务之前，首先要对这个任务有大致的了解，要有什么注意事项，就比如内存空间不足 这件事情，如果事先准备充分的话，可能就不会出现这样的问题。
3. 学会了如何向Linux内核中添加自己的系统调用。在此过程中，还意识到了检查的重要性，一个看似不起眼的错误，可能会酿成很大的错误。在进行下一个步骤的时候，首先要检查一下之前的步骤是否正确，检查一下并不会浪费很多时间，但却给后来步骤的正确性提供了保证。就好像这次sys\_hello不小心写成了sys.hello，这一个小小的符号错误，却浪费了很多的时间。
4. 学会了如何在Windows系统下编写一个批处理程序。同时还要注意一些细节方面的问题，比如文件的存储方式，文件的编码方式，以及判断时要有一定的准确度，不要随便加上空格之类的。