一、题目：

新增Linux系统调用

1. 需要重新编译Linux内核
2. 增加一个Linux的系统调用
3. 写一个用户程序进行调用测试。

系统调用实现的功能：计算一个数字的三次方，并打印出来。

二、版本：

编译版本：Win10上虚拟机（Virtual Box） Ubuntu（64bit）15.04系统

原内核版本：3.19.0

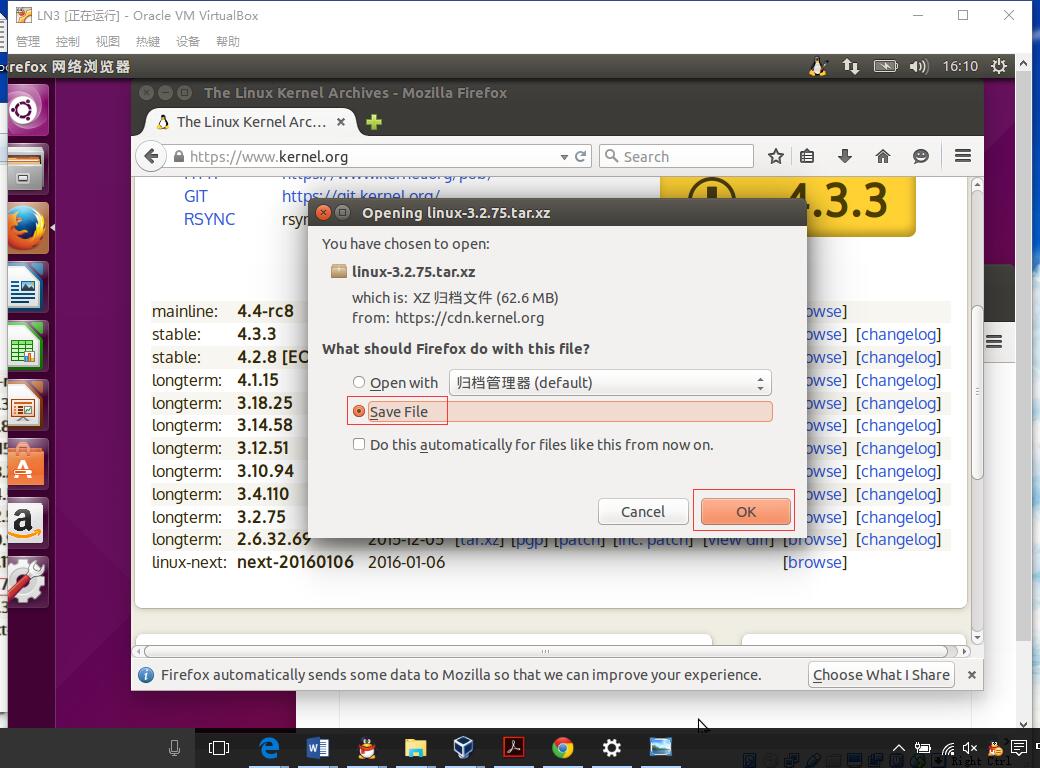
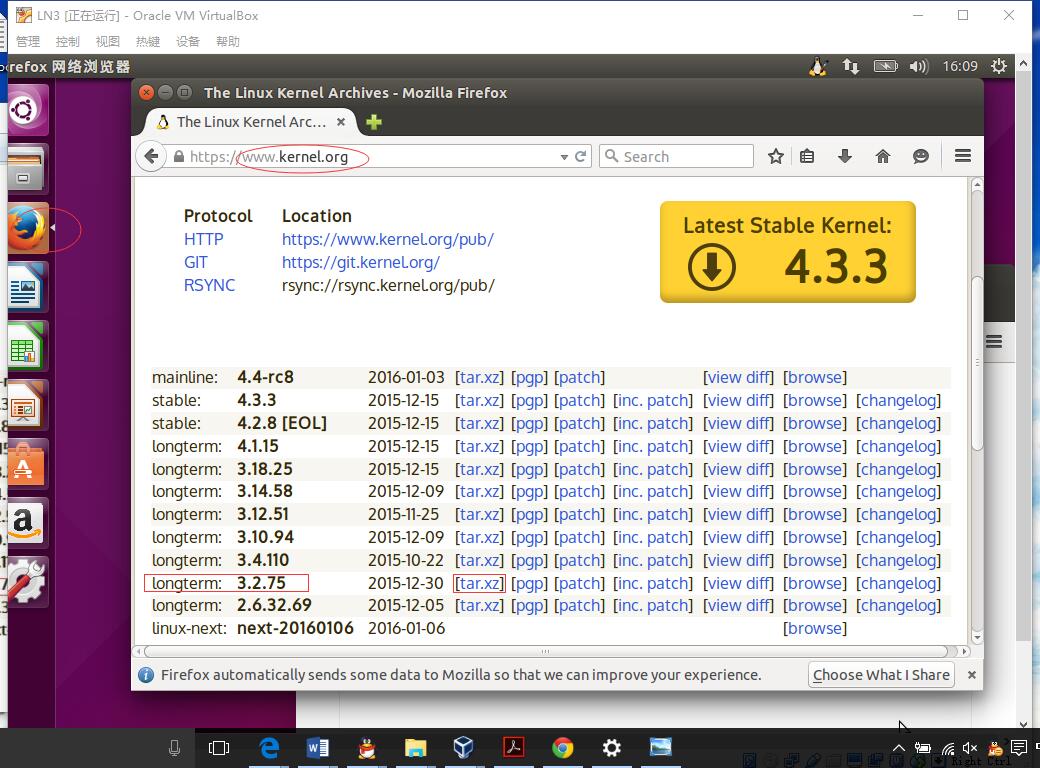
编译的新内核源码版本：3.2.75

不同内核版本，32位和64位文件位置以及文件名可能有所不同。添加系统调用时要根据自己的版本的文件位置和文件名修改相应的三个文件。

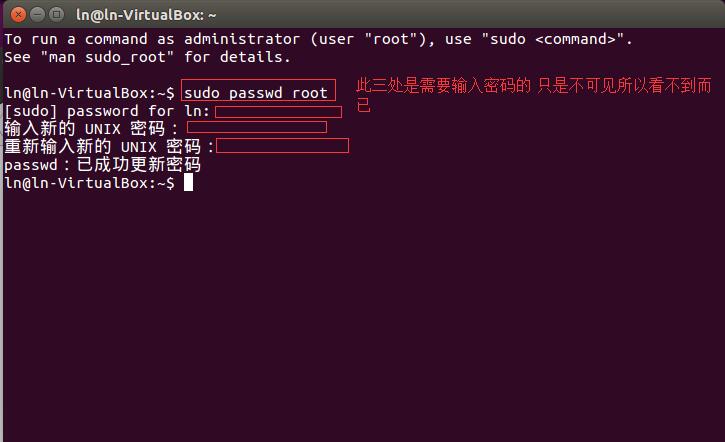
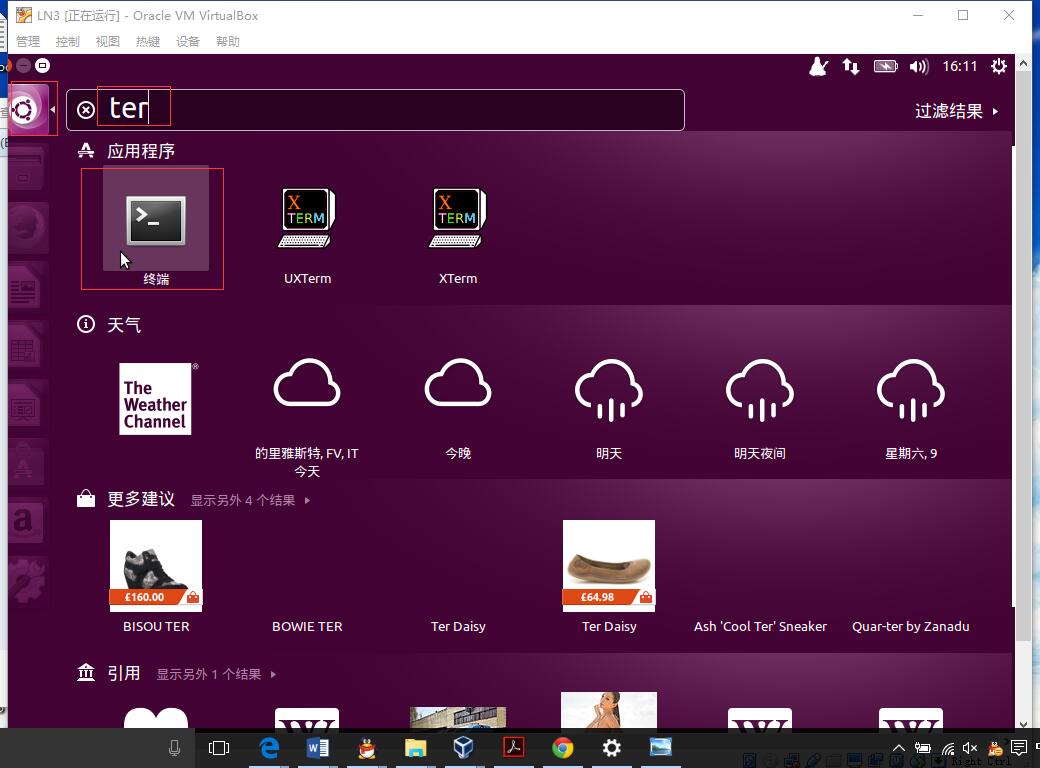
三、步骤：

1.前期准备：下载内核及解压

1.1下载内核：



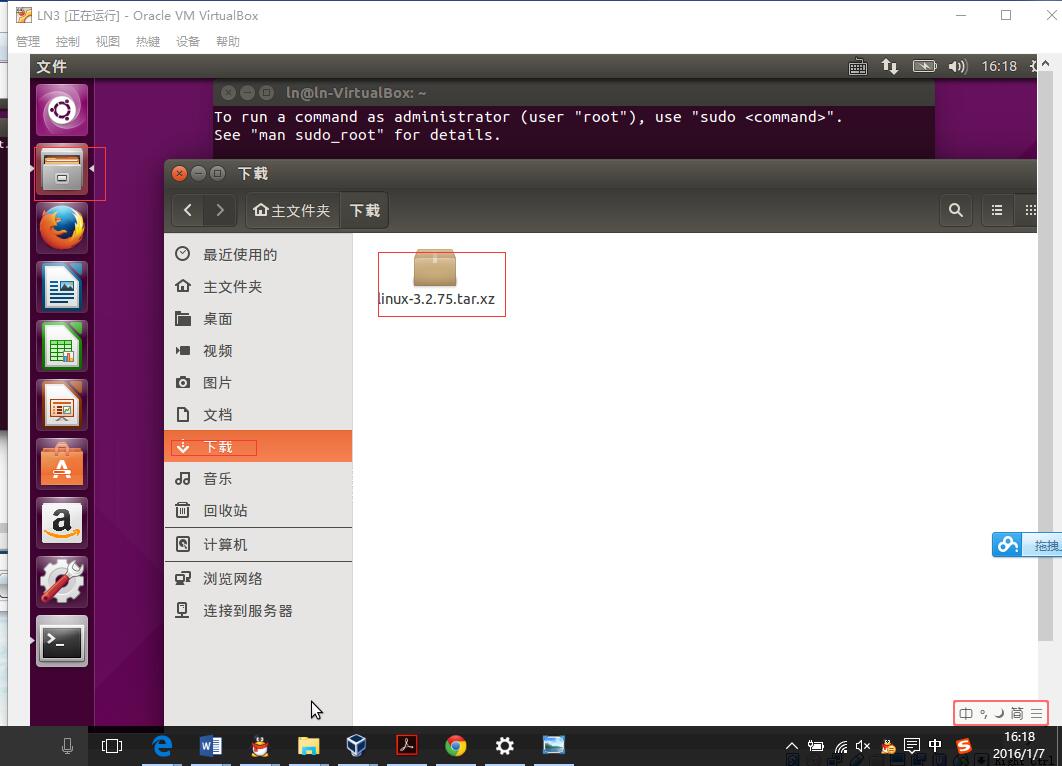
1.2打开终端 获得权限



然后再输入 su 输入刚刚设置的密码。

C:\Users\Dell\AppData\Local\Microsoft\Windows\INetCache\Content.Word\QQ截图20160109092120.jpg

1.3 移动文件并解压

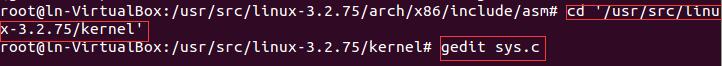


下载目录



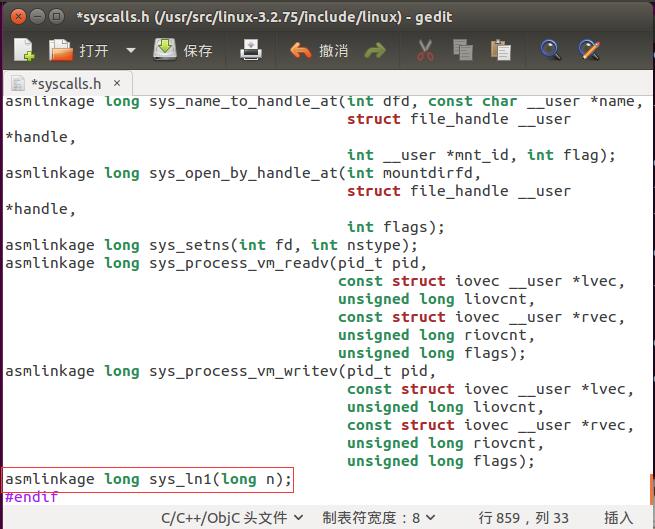
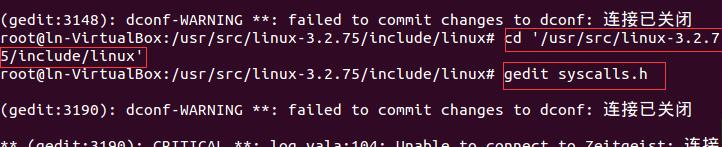
2.添加系统调用

2.1添加系统调用函数



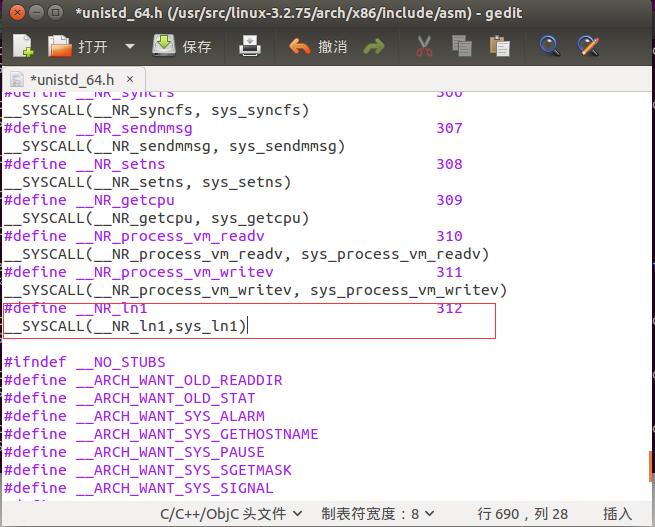
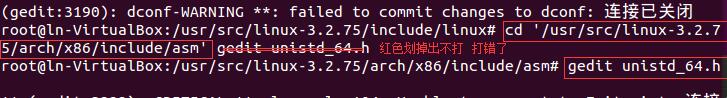
在文末添加调用函数。然后保存，关闭。

2.2 添加调用函数声明

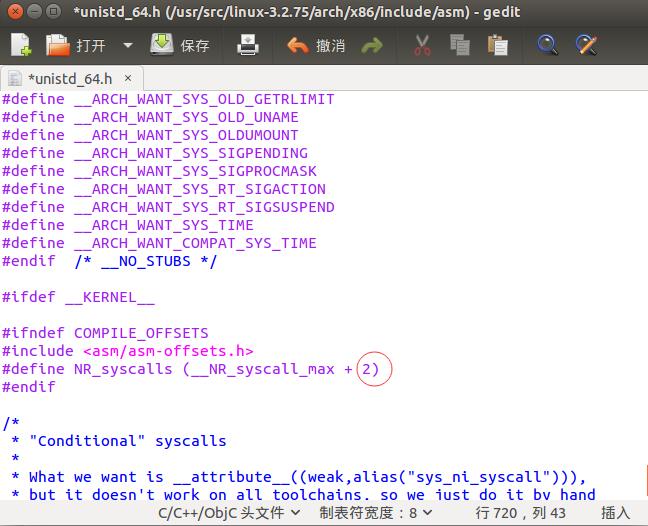


在文末 #endif前添加函数声明。然后保存关闭。

2.3添加系统调用号



因为前一个系统调用号是311 所以这里我们写312



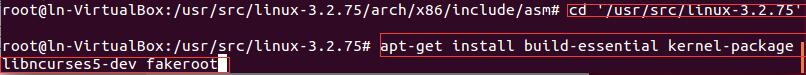
将原本的 #define NR\_syscalls (\_\_NR\_syscall\_max + 1)

修改为： #define NR\_syscalls (\_\_NR\_syscall\_max + 2)

然后保存，关闭。

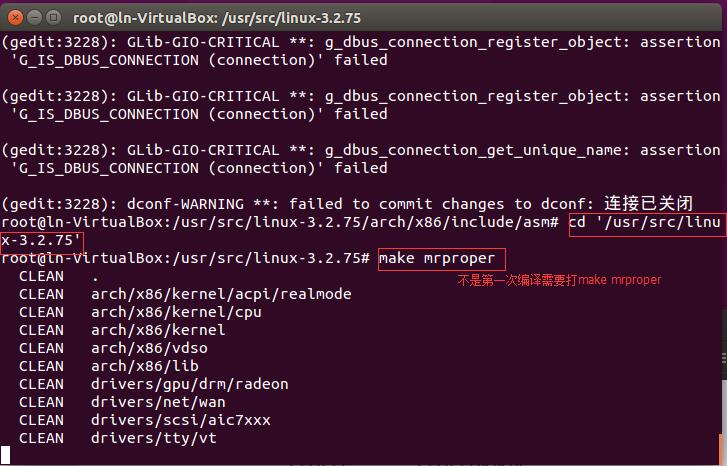
3.编译内核

3.1安装基本编译器套件

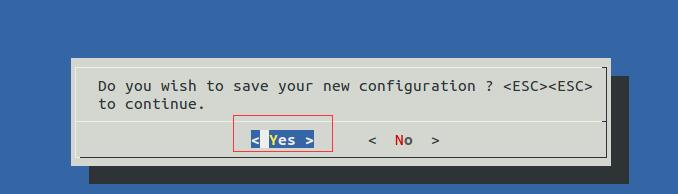
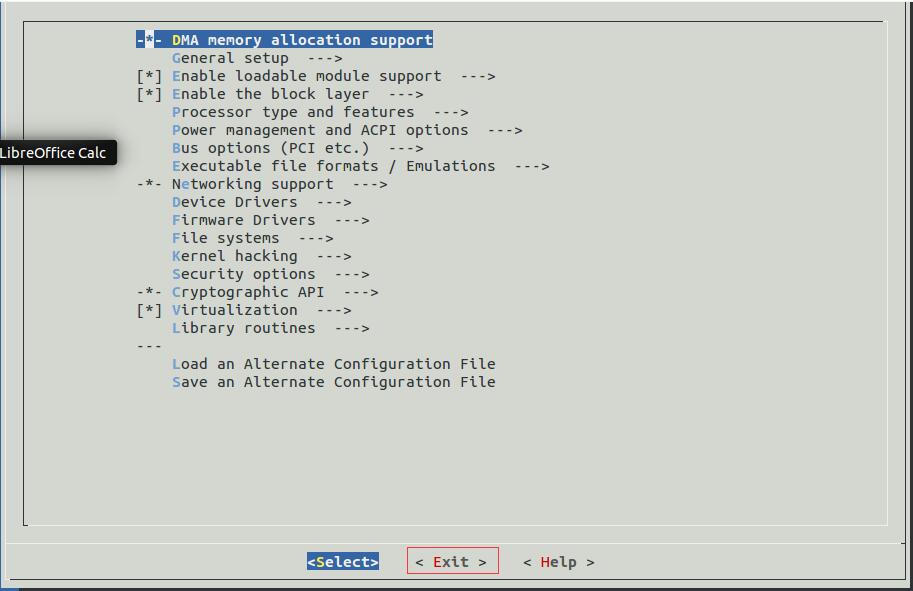
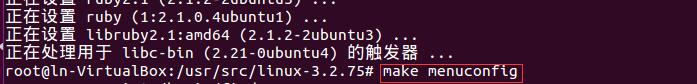


3.2编译

3.1 make mrproper清除以前配置（如果是第一次编译，不用执行此操作）



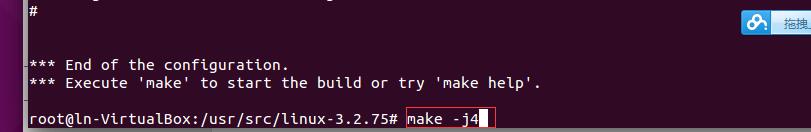
3.2 make menuconfig配置内核（此处选择了默认配置）



3.3 make编译内核



如果电脑是双核 则也可以用make –j4代替make（双核编译会更快一些）



接下来就是漫长的等待过程，不过要随时注意是否编译过程中因有错误而停止。我的电脑用了两个小时。

（也有教程里用的是make bzlmage和make modules，make bzlmage+make modules=make）

4.安装内核

4.1 make modules\_install

4.2make install

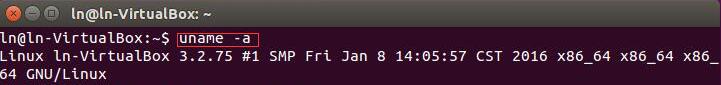
4.2 reboot 重启（或不使用命令直接对电脑进行重启）

重启之后在开机时候开机时候，如果是虚拟机需要同时按esc 和↑键出现开机启动项（如果是真机开机一般会自动出现开机启动项），选择新建的内核版本进入。





uname –a 查看内核版本，即为新建的内核版本。

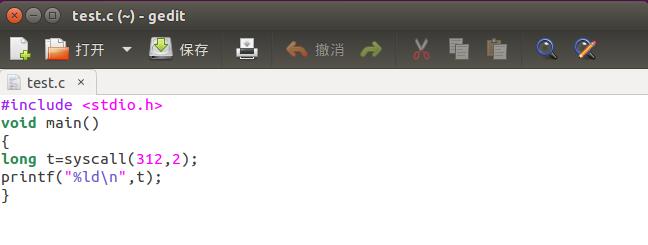


5.调用测试

5.1新建测试程序

gedit test.c 新建测试程序

编写测试程序：



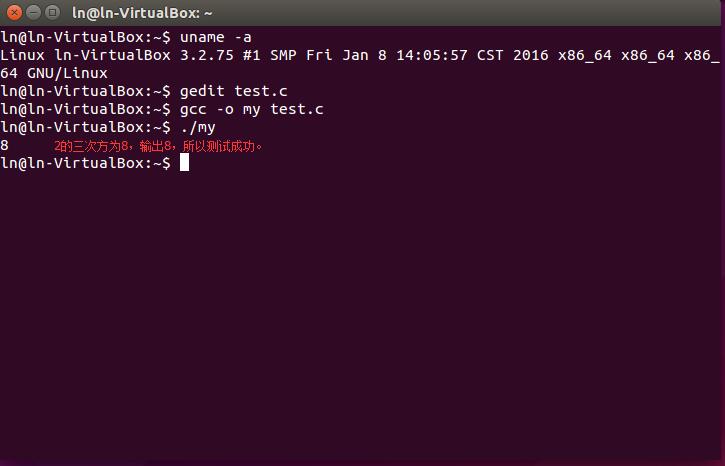
保存，关闭。

5.2测试程序编译和执行

gcc –o my test.c 测试

./my 执行

5.3执行结果



6．参考文章

<http://blog.sina.com.cn/s/blog_ad64b8200101axvl.html>

<http://blog.csdn.net/hj19870806/article/details/8725500>

<http://blog.chinaunix.net/uid-24782829-id-3211008.html>

2016/1/9