**操作系统实验课程**

**设 计 报 告**

题 目： Linux系统中进程、线程的实现机理

专业（班）： 2016级信安三班

项目组成员： 张铭诚 刘辉 赵磊 刘泽祺 陈鑫宇

课程名称： 操作系统原理及安全

任课教师： 严飞

**武汉大学国家网络安全学院**

2018年 3月

**成绩：**

|  |  |  |  |
| --- | --- | --- | --- |
| 学号 | 姓名 | 小组成绩 | 个人成绩 |
| 2016301500222 | 张铭诚 |  |  |
| 2016301500220 | 刘辉 |  |
| 2016301500213 | 刘泽祺 |  |
| 2016301500214 | 赵磊 |  |
| 2016301500252 | 陈鑫宇 |  |

**教师评语：**

**任课老师签名：　　　　　　　　时间：**

# 一、实验设计主题、内容、目标

**1、实验主题**

以Fibonacci实际问题为例，研究Linux中进程、线程的实现机理。

1. **实验内容**

以CentOS和Ubuntu为例，在Linux系统下编译内核并增添自定义的系统调用。

1. **实验环境**

VMWare Workstation Pro 14

ubuntu-16.04.3-desktop-amd64.iso

**4、实验目标**

（1）熟悉Linux系统，掌握基础Linux命令。

（2）在实验中体会Linux系统的开源性，加深对系统调用相关知识的理解，提高大家分析并解决问题的素质。

# 二、实验原理

# 实验过程

以下步骤均以Ubuntu为例。

1. **在VMWare中配置虚拟机**

从网上下载Ubuntu的iso镜像文件并按照VMWare提示完成配置。本次实验中Ubuntu内存设置为4GB，硬盘空间分配了50GB，处理器数量和每个处理器内核的数量均设置为2，网络链接选择NAT模式。

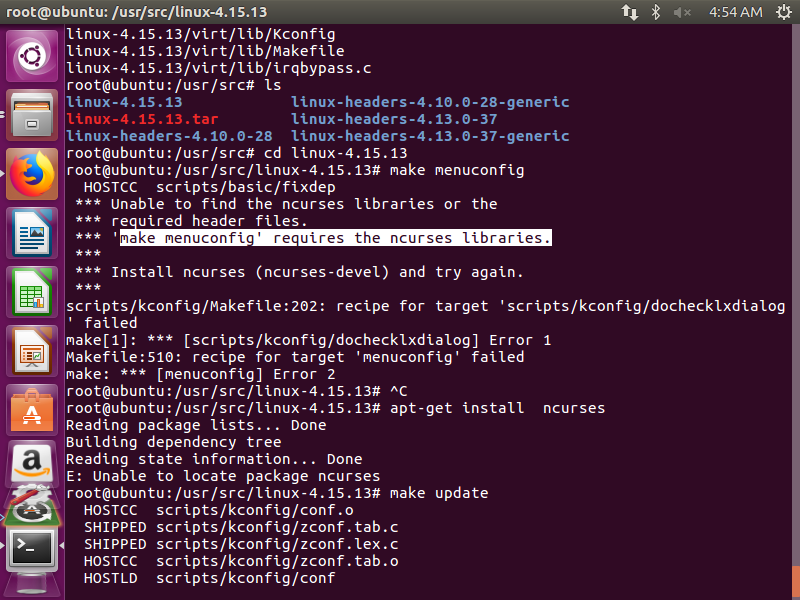
注意内存和硬盘不可分配太小，不然编译时会因为空间不足而停止。

## 2、 解压内核

利用Ubuntu自带的FireFox浏览器访问www.kernel.org并下载最新版内核，本次实验以4.15.13版本为例。

进入Documents目录下，执行sudo mv \*/linux-4.15.13.tar.xz /usr/src，将压缩包移动到/usr/src文件夹下，再依次执行 sudo xz -d linux-4.15.13.tar.xz,sudo tar -xvf linux-4.15.13.tar,最后用sudo rm linux-4.15.13.tar删除已解压的压缩包。

下图为解压完成后的src目录下的文件，可以看见已经生成了linux-4.15.13文件夹。

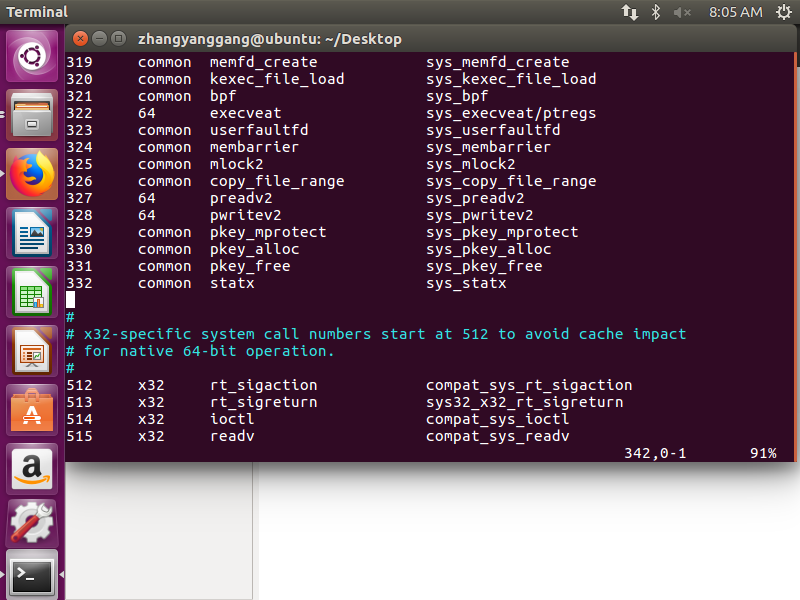


## 3、添加系统调用

## 增加系统调用号

执行命令 sudo gedit ./syscall\_64.tbl，打开并编辑系统调用入口表，找到系统调用的最大调用号，并紧接其后加上自己的调用号，此处应在光标处加上如下所示字符：

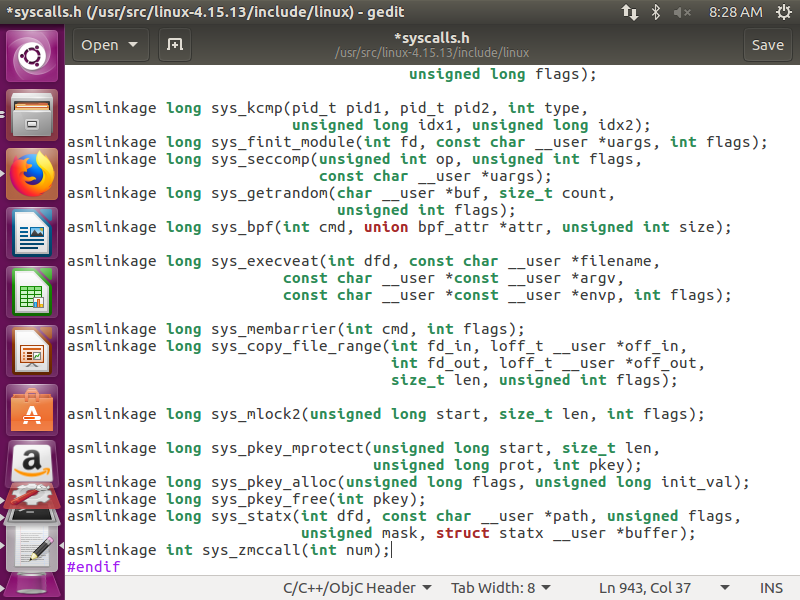
333 64 zmc sys\_zmccall，并记住我们的系统调用号是333。此处忘记截输入后的图。



（2） 声明系统调用函数

使用命令 sudo gedit  include/**[Linux](http://lib.csdn.net/base/linux" \t "_blank" \o "Linux知识库)**/syscalls.h打开系统调用表，并在#endif之前的空白处声明自己所添加的调用函数。

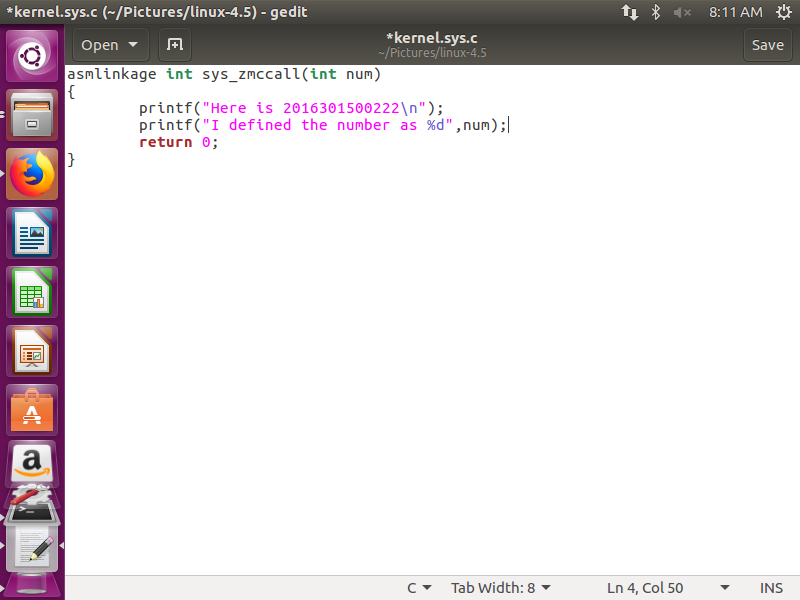
命令将系统调用添加到系统调用表中，在末端加入系统调用函数的声明。



此处如图所示，按照格式应该输入asmlinkage int sys\_zmccall(int num);

（3）设定系统调用函数

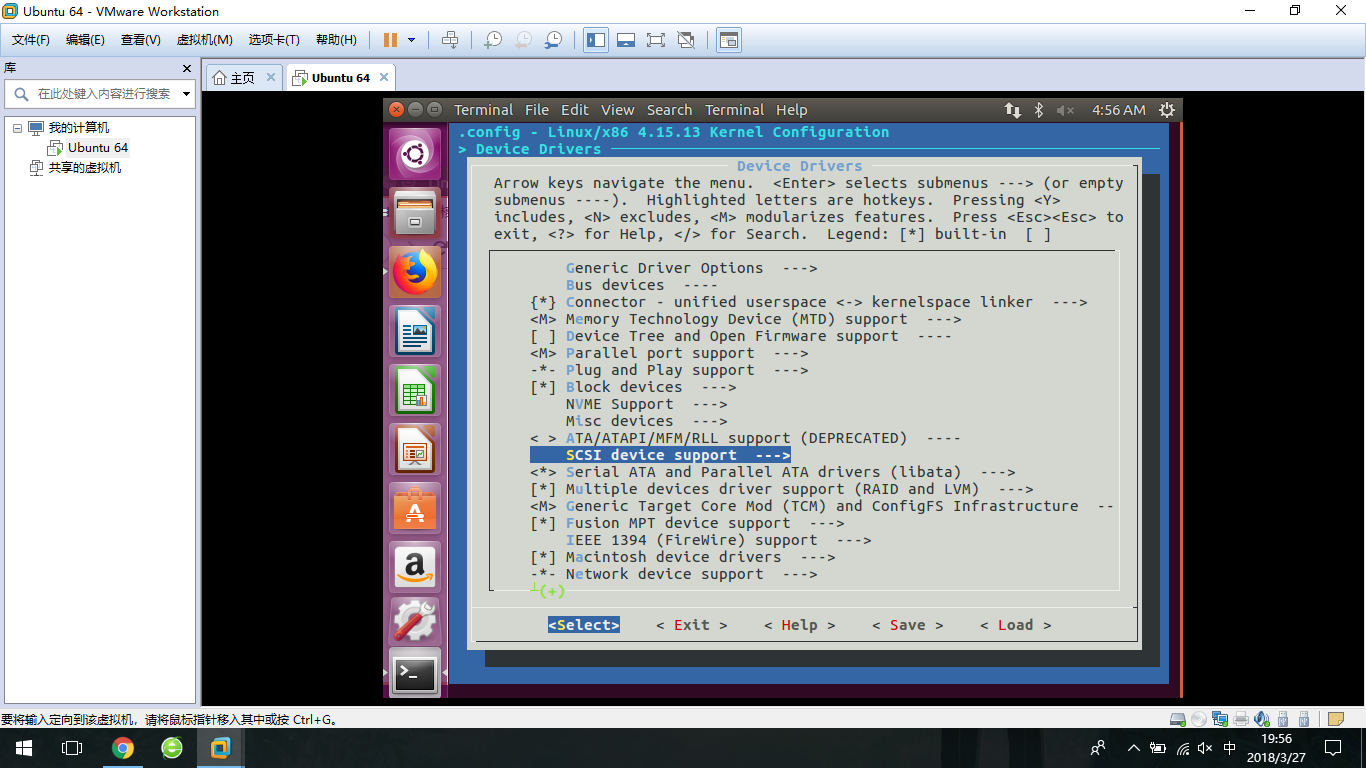
输入自己所设定的代码，利用sudo gedit ./kernel/sys.c打开文件并加入如下代码：



## 4 编译内核

### （1）配置内核

输入 sudo make menuconfig命令，按照自己的具体需求更改设置。



### （2）编译内核

在linux-4.15.13目录中，执行sudo make

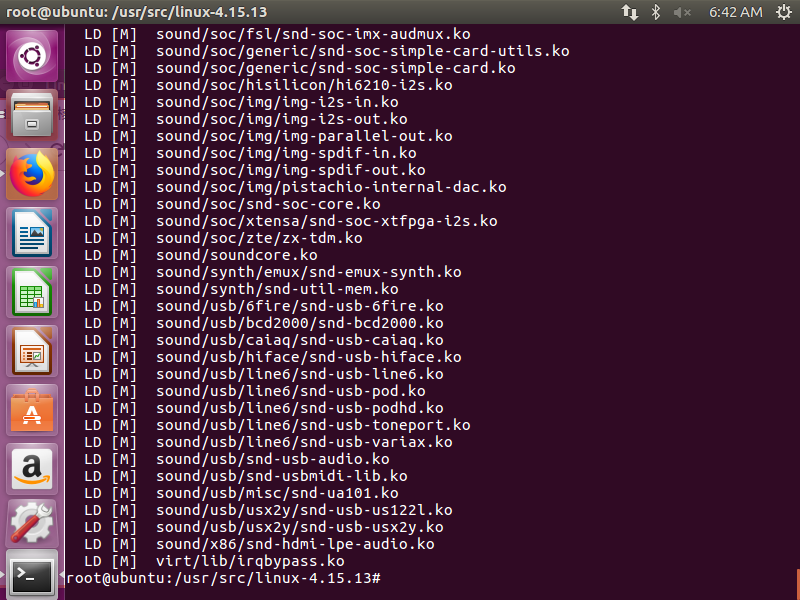


直接按照提示，执行命令sudo make -j 4 bzImage。此处-j 4指明进行四核编译。（因为前面设置的时候分配了两个处理器，每个处理器两个内核）。会编译很久。

下图为编译结束后的截图。

### Screenshot from 2018-03-27 05-29-33（3）编译模块

### 执行命令 make -j 4 modules编译模块，同样会编译很久。



此为编译模块成功后的截图。

1. 安装内核

<1>安装模块

执行sudo make -j 4 modules\_install

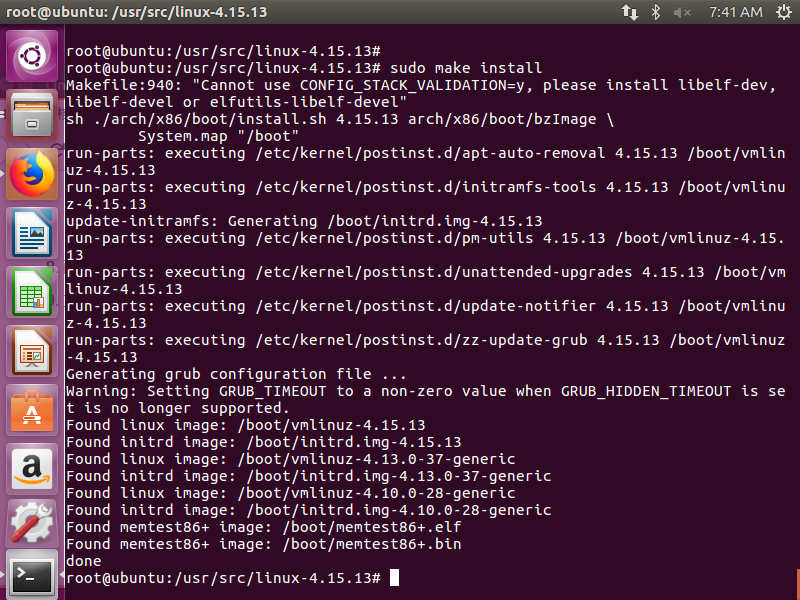
<2>建立要载入ramdisk的映像文件

执行 sudo mkinitramfs 4.15.13 -o /boot/initrd-4.15.13.img，其中，4.15.13是/lib/modules下的目录名称及内核版本号，initrd-4.15.13.img是生成的内核镜像号。

<3>安装内核

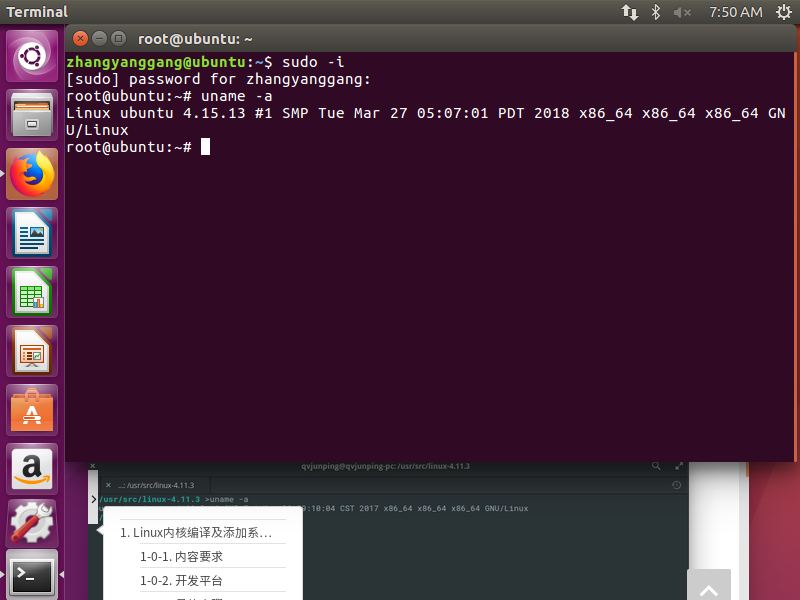
执行 sudo make install

此为执行完安装内核三条指令之后的截图。



### （5）重启系统并观察内核版本

执行 sudo reboot进行重启，并执行uname -a观察内核版本。



重启后发现内核版本已经是4.15.13。

5、验证系统调用

（1）设计并编译mykernel.c文件来使用自定义的系统调用。

#include<unistd.h>

int main()

{

syscall(333, 2);

return 0;

}

1. 编译程序

执行gcc ./mykernel.c

1. 运行程序

执行 ./a.out

1. 查看输出结果

执行dmesg，查看系统调用输出结果。

result

与预期结果相同，此次实验成功。

**四、小组分工**

1、收集资料

由张铭诚同学上网查阅实验步骤和实验原理，分享给组内的成员，大家共同学习。

1. 进行实验

组内的每个同学都应该亲自动手实践，加深理解。每个人的环境和硬件不同，每个人在实验过程中遇到的问题也不同，大家共同讨论，互相帮助，确保每个人都完成实验。

1. 总结分析

实验结束后，组长张铭诚同学进行了总结，指出了存在的不足。确定由张铭诚同学撰写本次的实验报告。

# 五、实验心得与总结

本次实验中遇到了许多实际问题。

1. 最开始时在某不知名网站下载的Ubuntu软件源是国外的，有很多现在无法访问和下载，之后将软件源全部换成阿里云的即可正常更新软件源。
2. 在实验过程中，经常会缺少一些必要的组建，如openssl,ncurses等，解决这些问题的方法几乎都是一样的，即 sudo apt-get install XXX，通过该语句可以解决绝大部分缺少文件的问题，是一种通用的解决方法。
3. 刚开始的时候内存分配不足，导致内核编译一半因为没有剩余空间而停止，要重新来过，但是内存和磁盘空间分配过大又会极大影响电脑性能，使得使用主机的体验很差，所以最终决定采用4GB内存和50GB磁盘空间的折中方案。
4. 通过本次实验，我们对“用户态”和“内核态”的区别、相互切换与协作的过程有了清晰的认识，对内核的理解也由抽象变为具体，在查阅资料的过程中加深了自己对所学知识的掌握程度，也锻炼了自己的动手能力和解决问题的能力。

总体来说，此次实验比较成功。