Project 1: 向Linux内核中添加系统调用

516030910259 刘欣鹏

1. 实验目的

以Linux内核为例，了解操作系统中系统调用的工作原理。

1. 实验原理

系统调用向用户态程序提供了由操作系统实现的各个功能的接口。在Linux系统中，一个系统调用是通过在EAX寄存器中存储系统调用号，在其他硬件寄存器中存储系统调用参数，随后执行陷阱指令而完成的。陷阱指令执行后，系统调用号被用于索引相应handler的起始地址。进程随后跳跃到该地址，并将进程由用户态转到内核态。在内核态下，进程得以执行在用户态下不能执行的指令。

1. 实验步骤
   1. 下载内核源代码并解压

访问https://www.kernel.org/。在我的实验中下载的版本是4.15.10。

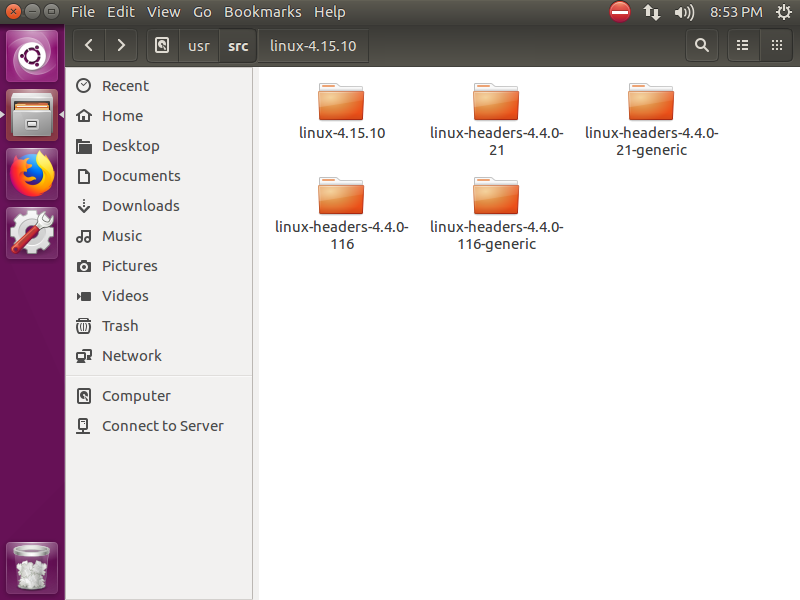
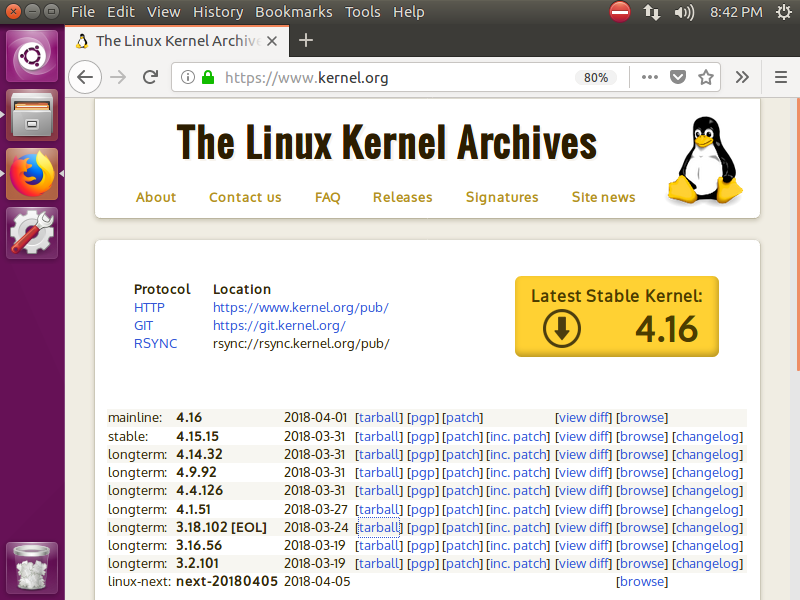
终端内输入命令：

su -

mv linux-4.15.10.tar.xz ./usr/src/linux-4.15.10.tar.xz

cd ./usr/src

tar -xvf linux-4.15.10.tar.xz



* 1. 添加系统调用号

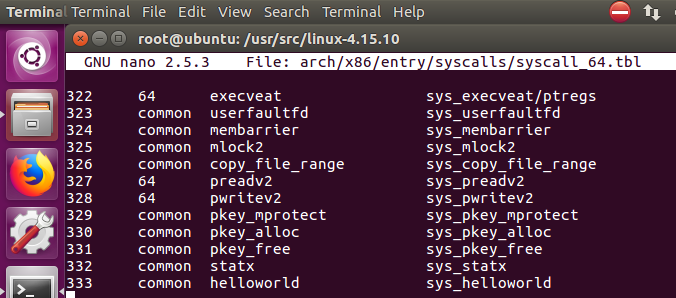
在该版本的Linux内核中，系统调用号存储在内核文件夹下 ./ arch/ x86/ entry/ syscalls/ syscall\_64.tbl 文件中。

终端内键入命令：

cd linux-4.15.10

nano arch/x86/entry/syscalls/syscall\_64.tbl

在后面添加系统调用函数helloworld及其相关信息。在我的虚拟机上，其系统调用号为333。

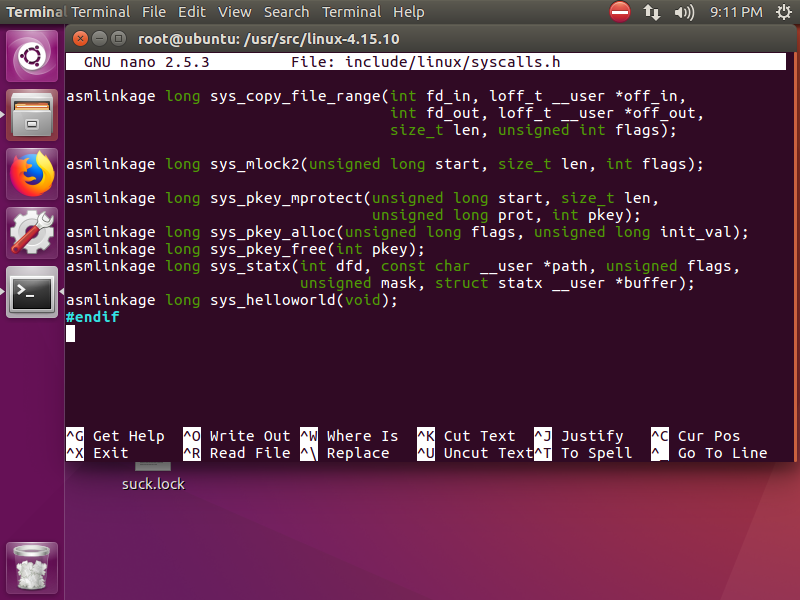


* 1. 声明系统调用函数原型

终端内键入命令：

nano include/linux/syscalls.h

在后面添加新系统调用的原型。

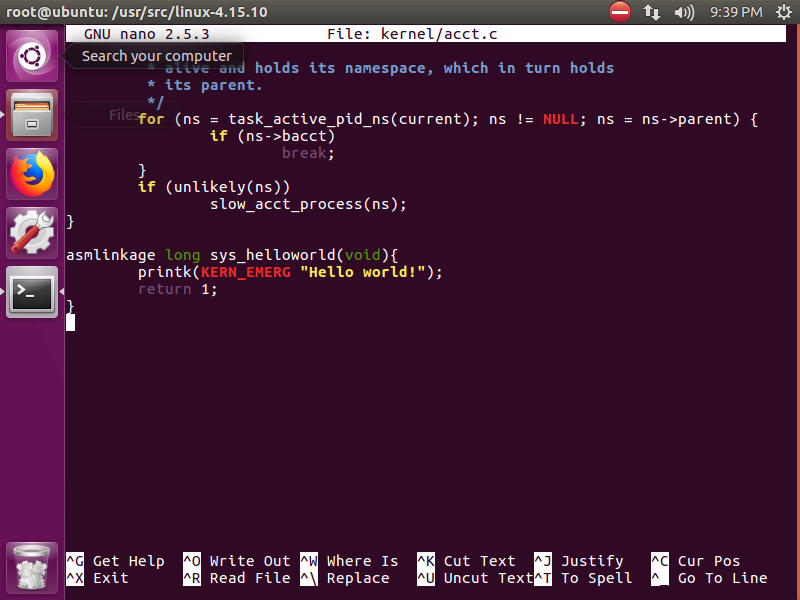


* 1. 添加系统调用函数的定义

终端内键入命令：

nano kernel/acct.c

在最后加入helloworld系统调用函数的定义。



* 1. 编译、安装内核

终端内依次键入命令：

menuconfig

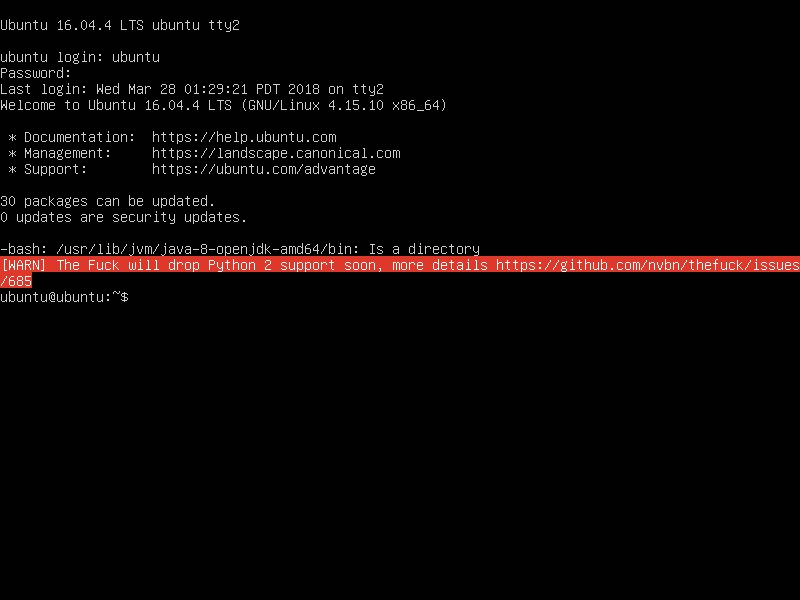
make

make modules\_install

make install

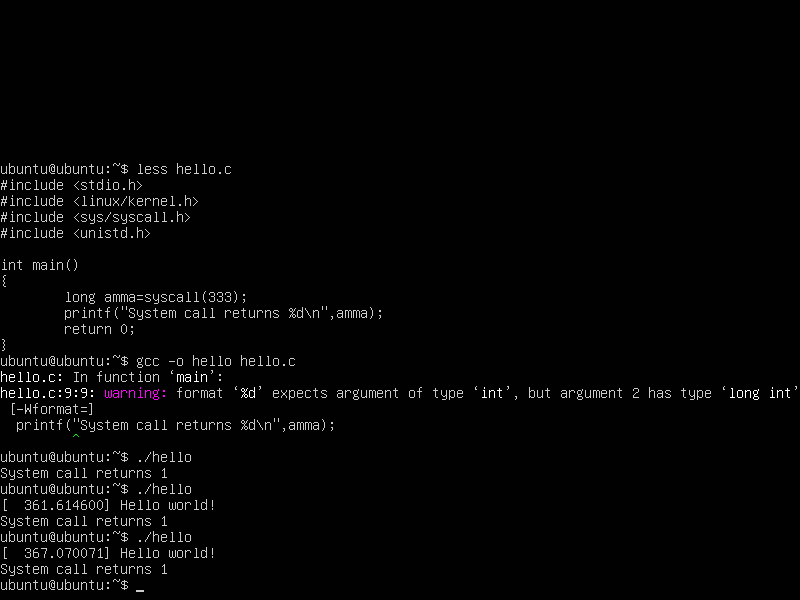
等待指令执行完毕，时间较长。

* 1. 重启进入新内核

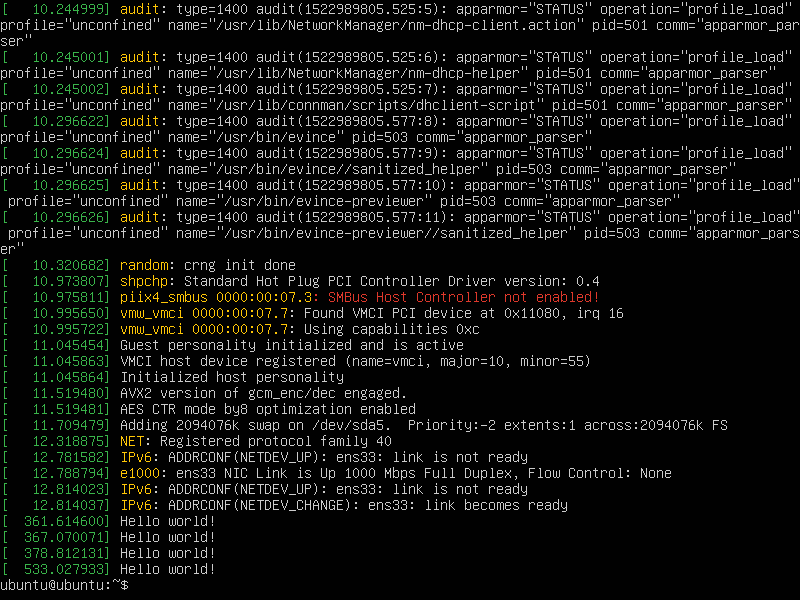


如画蓝线处所示，当前使用的内核即是添加了新系统调用的4.15.10版本内核。

* 1. 测试系统调用



创建hello.c文件，内容如上。用gcc编译并执行hello.c。输出结果如上。亦可通过键入dmesg查看系统日志。



1. 心得与体会

在实验过程中，遇到的主要问题是教科书上实验指导与实际操作的不符。教科书上使用的内核版本为2.x，而现在主流的Linux内核已经达到了4.x版本，教科书上给出的许多路径和指导已经不再适用了。因此，在实验过程中我主要是通过上网查询相关资料而最终完成的。

通过本次实验，我初步了解了如何在Linux内核中添加简单的系统调用，如何在编译并安装新的Linux内核，以及如何用C语言程序进行系统调用，在实验中加深了对Linux系统使用的了解。