**Linux系统增加系统调用**

**实验报告**

姓名： 余泽晨

学号： 71115112

东南大学计算机科学与工程学院、软件学院

School of Computer Science & Engineering

College of Software Engineering

Southeast University

二0 17 年 3 月

一、实验目的

通过实验，熟悉Linux操作系统和命令的使用，掌握构建与启动Linux内核的方法；掌握用户程序如何利用系统调用与操作系统内核实现通信的方法，加深对系统调用机制的理解；进一步掌握如何向操作系统内核增加新的系统调用的方法，以扩展操作系统的功能。

二、实验内容

1. Linux环境下的C或C++编译和调试工具的使用。

2. 向Linux内核增加新的系统调用，系统调用名称和功能自行定义，但必须实现如下输出功能：“My Student No. is 71115112，and My Name is Yu Zechen”。

3. Linux新内核的编译、安装和配置。

4. 编写应用程序以测试新的系统调用并输出测试结果。

三、实验步骤

1.环境

VMWare Player+64位Ubuntu14.04虚拟机

2. 下载Linux内核源码

在终端中使用sudo su命令并输入密码,获取root权限.

输入 apt-get install linux-source,下载linux源码(3.13.0版本)

3.解压内核

输入: cd /usr/src/

tar -xvf ./linux-source-3.13.0.tar.bz2

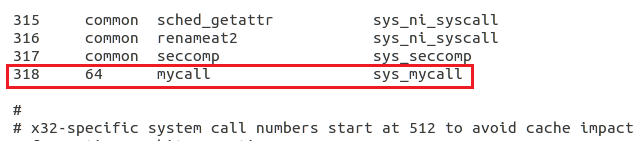
4.增添系统调用

(1) 设置系统调用号

输入: cd ./linux-source-3.13.0

gedit arch/x86/syscalls/syscall\_64.tbl

添加318号系统调用sys\_mycall



注:32位系统修改syscall\_32.tbl

(2)添加系统调用函数的定义

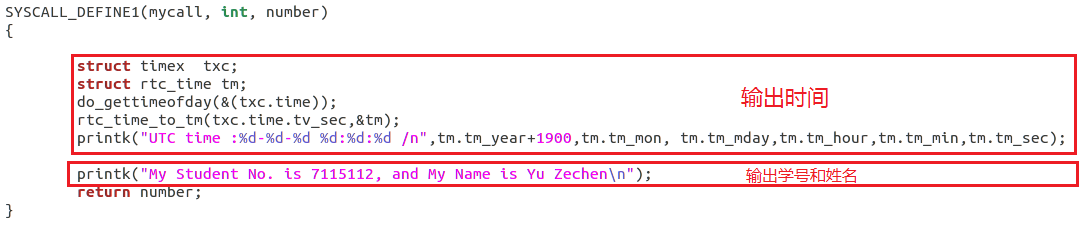
输入: gedit kernel/sys.c

编辑sys.c文件

增加包含的头文件:



定义新的系统调用函数:



注1:DEFINE1指此系统调用函数有一个参数

注2:系统调用统函数的声明在include/asm-generic/syscalls.h中已经存在,就不用再次声明了

5.编译内核

(1)安装编译所需工具

输入: apt-get install libncurses5-dev

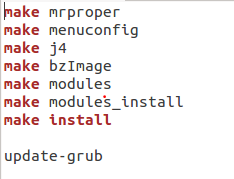
(2)编译内核

新建一个shell文件

输入: touch mycall.sh

gedit mycall.sh

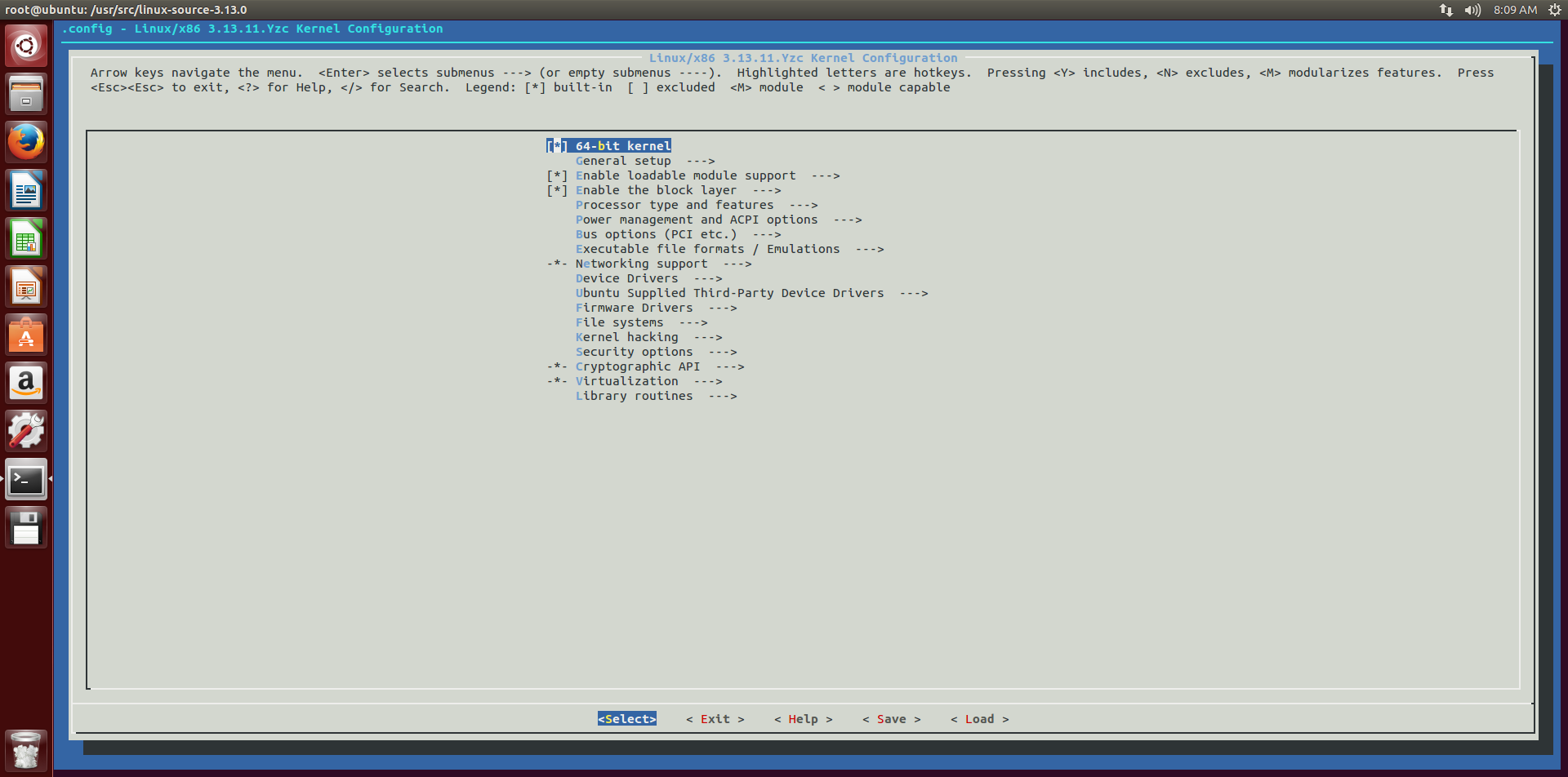
编辑mycall.sh:



输入: sh mycall.sh

开始编译内核

编译完成后重启系统并进入新内核

注:编译时进入内核的菜单式配置: 

选择Save后选择Exit退出,继续编译内核

6. 测试新系统调用

(1)新建一个test.c文件并编辑

#include<stdio.h>

#include<linux/unistd.h>

int main(){

printf("%d\n", syscall(318,71115112));

return 0;

}

(2)编译test.c

在test.c文件所在路径下输入: gcc test.c

(3) 查看输出

输入: ./a.out,查看函数结果

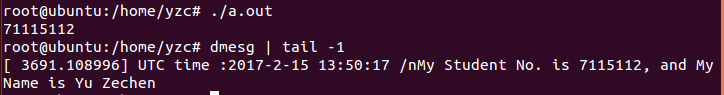
dmesg | tail -1,查看最后一条内核的信息

四、主要数据结构及其说明

系统调用表syscall\_64.tbl，记录着所有系统调用号。

系统调用定义文件sys.c，记录着所有系统调用的函数实现。

五、程序运行结果（系统截图）



六、实验体会

1.在实验中,熟悉了一些linux的命令行操作,对linux系统和系统调用也有了更深的认识.

2.实验中遇到的最大问题就是在调用新系统调用时总是返回-1,后来才发现有2个问题,

一个是我用的是64位的系统,但增加系统调用时修改了syscall\_32.tbl而不是syscall\_64.tbl,导致调用系统调用时出错,一个是系统默认进入原先的内核,必须在开机时进入grub界面选择内核,才能进入新编译的内核.

七、源程序并附上注释

1.新的系统调用函数:

#include <linux/timer.h>

#include <linux/timex.h>

#include <linux/rtc.h>

SYSCALL\_DEFINE1(mycall, int, number)

{

struct timex txc;

struct rtc\_time tm;

do\_gettimeofday(&(txc.time));

rtc\_time\_to\_tm(txc.time.tv\_sec,&tm);

printk("UTC time :%d-%d-%d %d:%d:%d /n",tm.tm\_year+1900,tm.tm\_mon, tm.tm\_mday,tm.tm\_hour,tm.tm\_min,tm.tm\_sec);**//输出时间**

printk("My Student No. is 7115112, and My Name is Yu Zechen\n");**//输出姓名和学号**

return number;

}

2.测试函数:

#include<stdio.h>

#include<linux/unistd.h>

int main(){

printf("%d\n", syscall(318,71115112));**//调用318号系统调用函数并输出其返回值**

return 0;

}