**西安电子科技大学**

**计算机学院**

**实**

**验**

**报**

**告**

**题目：** Linux添加系统功能调用

**班级：** 1403013

**姓名：** 周林茂

**学号：** 14030130098

**一、理论分析（分值：20%）**

系统调用，顾名思义，说的是操作系统提供给用户程序调用的一组“特殊”接口。用户程序可以通过这组“特殊”接口来获得操作系统内核提供的服务，比如用户可以通过文件系统相关的调用请求系统打开文件、关闭文件或读写文件，可以通过时钟相关的系统调用获得系统时间或设置定时器等。 从逻辑上来说，系统调用可被看成是一个内核与用户空间程序交互的接口——它好比一个中间人，把用户进程的请求传达给内核，待内核把请求处理完毕后再将处理结果送回给用户空间。

Linux中每个系统调用都有相应的系统调用编号作为唯一的标识，内核维护一张系统调用表sys\_call\_table，表中的元素是系统调用函数的起始地址，而系统调用号就是系统调用在调用表的偏移量。

在x86兼容机上，早期的Linux内核使用INT $0x80软中断来进行系统调用处理，用户调用系统调用时，将系统调用编号及其他参数装入寄存器中，然后出发中断，进入中断处理函数，中断处理函数将CPU上下文从用户态切换到内核态，通过终端编号从系统调用表中索引对饮调用函数的地址，去出参数进行处理，并获得调用的返回值，最后切换到调用的应用程序。

**二、设计与实现（分值：30%）**

通过重新编译内核来添加系统调用，在arch/x86/syscalls/syscall\_64.tbl表中添加一行，指定系统调用编号，调用函数入口，然后实现相应的函数，最后重新编译内核，重新加载新的内核即可实现添加系统功能调用的目的。

修改arch/x86/syscalls/syscall\_64.tbl,在文件末尾添加代码common my\_syscall

sys\_my\_syscall 并给定合适调用编号。

Sys\_my\_syscall函数实现

asmlinkage int sys\_my\_syscall(int number)

{

Printk(“this is my first syscall”);

Return number;

}

此函数向系统内核日志输出一段字符”this is my first syscall’,并返回参数number

打开/usr/src/linux-source-2.6.38/x86/kernel/syscall\_table\_32.8

添加一行，并记住其编号

.long sys\_mycall

代开/usr/src/linux-source-2.6.38/arch/x86/include/asm/unistd\_32.h

跟据上面的标号添加一行：

#define\_NR\_mycallN (N为上面的编号)

编译内核：

进入源代码目录：

【1】#cd/usr/src/linux-source-3.2.0

【2】#make mrproper //清理以前便宜留下的临时文件

【3】#make localmodconfig //自动精简内核

【4】#make-kpkg clean

#fakeroot make-kpkg --initrd --append-to-version=\_xidian kernel\_image

安装内核

便宜成功的内核在上一层目录，执行命令

#cd\_\_

#dpkg -i linux-image-3-.2.39-xidian\_3.2.39-xidian-10.00.Custom\_i386.deb

Linux-image-3.2.39-xidian-10.00.Custom\_i386.deb

#reboot //重新验证内核

#uname -4//在terminal中输入此命令，查看版本号，版本相同则替换成功

#dpkg -P linux-image-3.2.39-xidian\_3.2.39-xidian

Linux-image-3.2.39-xidian\_3.2.39-xidian

测试：

编写程序：

#include<linux/unistd.h>

#include<sys/syscall.h>

#include<stdio.h>

int main ()

{

printf(“the number is: %d\n”,syscall(233,100));

}

**三、实验结果（分值：10%）**



输出值为预期值，调用成功

**四、心得与收获（分值：40%）**

本实验首次接触Linux,感觉第一次上机什么都不知道，就连Terminal都不知道在哪里打开，一步一步，自己在linux下写了第一个C程序，Helloworld,感觉好兴奋，第二次上机尝试添加系统功能调用，在学校电脑上一直编译没有出结果，最后放弃了，在自己电脑上编译成功并且得到正确结果，虽然调用成功，但是调用过程理解还是不透测。

程序通过调用入口点system\_call进入系统调用，然后根据系统调用参数实现具体调用，并将计算结果返回给主程序。