

上海交通大学

《操作系统》课程

学生实验报告

实验名称: Project1:Introduction to Linux Kernel Modules

姓 名: 洪瑄锐

学 号: 517030910227

班 级: F1703302

手 机: 15248246044

任课老师: 吴晨涛

2019 年 5 月 26 日

目录

1. 实验要求
2. 程序设计思想及代码解释
 - 2.1 Kernel Modules Overview
 - 2.2 Loading and Removing Kernel Modules
 - 2.3 The /proc File System
3. 运行结果

一、实验要求

了解 Linux 的基本使用,学习如何创建内核模块并将其加载到 Linux 内核中。

二、程序设计思想及代码解释

I. Kernel Modules Overview

通过对课本的阅读,了解了以下几个函数和命令:

1. **lsmod**:以名称,大小,使用模块的位置的格式列出当前内核模块
2. **module_init(simple_init),module_exit(simple_exit)**:用于向内核注册模块入口点和出口点,模块入口点函数必须返回一个整数值,0 表示成功,其他任何值表示失败。模块推出点函数返回 void。
3. **prink**:是 **printtf()** 的内核等价,输出被发送到 kernel log buffer 中,可用 **dmesg** 命令读取。
4. **make**:利用 makefile 文件编译.c 文件

II. Loading and Removing Kernel Modules

通过对课本的阅读,了解到以下几个命令:

1. **sudo insmod simple.ko**:将 simple 模块加载到内核中
2. **sudo rmmod simple**:删除 simple 模块
3. **sudo dmesg -c**:清理 kernel log buffer

在这个步骤中,课本要求我们在 **simple_init()** 中打印 **GODLEN_RATIO_PRIME**,并输出 **jiffies** 和 **HZ**,在 **simple_exit()** 中打印 **jiffies**,输出 3300 和 24 的最大公约数,并利用 **jiffies** 计算从 simple 加载到移除所用的时间。

```

int simple_init(void)
{
    initJiffies = jiffies;
    printk(KERN_INFO "Loading Module\n");
    printk(KERN_INFO "%lu\n", GOLDEN_RATIO_PRIME);
    printk("%d\n", HZ);
    printk("%d\n", initJiffies);

    return 0;
}

/* This function is called when the module is removed. */
void simple_exit(void) {
    finalJiffies = jiffies;
    printk(KERN_INFO "Removing Module\n");
    unsigned long g = gcd(3300, 24);
    printk ("%lu\n",g);
    printk("%d\n", finalJiffies);
    printk("seconds:%d\n", (finalJiffies - initJiffies) / HZ);
}

```

III. The /proc File System

该步骤要求设计内核模块使得在/proc 文件系统创建新的条目。通过阅读课本可知以下几个重要函数：

1. `ssize_t proc_read(struct file*file, char __user*usr_buf, size_t count, loff_t *pos)`: 在每次/proc/hello 被读取时该函数都会被调用，注意返回值为 0 时该函数才停止调用。
2. `copy_to_user(usr_buf, buffer, rv)`: 此函数将内核内存缓冲区的内容复制到存在于用户空间的 `usr_buf` 变量中。

任务包括创建 /proc/jiffies 和 /proc/seconds，修改示例代码中的 `proc_read` 函数即可。

/proc/jiffies

```

ssize_t proc_read(struct file *file, char __user *usr_buf, size_t count, loff_t *pos)
{
    int rv = 0;
    char buffer[BUFFER_SIZE];
    static int completed = 0;

    if (completed) {
        completed = 0;
        return 0;
    }

    completed = 1;

    rv = sprintf(buffer, "%lu\n", jiffies);

    // copies the contents of buffer to userspace usr_buf
    copy_to_user(usr_buf, buffer, rv);

    return rv;
}

```

/proc/seconds

```
ssize_t proc_read(struct file *file, char __user *usr_buf, size_t count, loff_t *pos)
{
    int rv = 0;
    char buffer[BUFFER_SIZE];
    static int completed = 0;

    if (completed) {
        completed = 0;
        return 0;
    }

    completed = 1;

    finalJiffies = jiffies;
    unsigned long sec = finalJiffies - initJiffies;
    sec = sec / HZ;
    rv = sprintf(buffer, "%lu\n", sec);

    // copies the contents of buffer to userspace usr_buf
    copy_to_user(usr_buf, buffer, rv);

    return rv;
}
```

三、运行结果截图

1. simple

```
osc@ubuntu:~/final-src-osc10e/ch2$ sudo insmod simple.ko
osc@ubuntu:~/final-src-osc10e/ch2$ sudo rmmod simple
osc@ubuntu:~/final-src-osc10e/ch2$ dmesg
[ 4007.642456] Loading Module
[ 4007.642460] 11400862456688148481
[ 4007.642463] 250
[ 4007.642464] 926830
[ 4020.104493] Removing Module
[ 4020.104497] 12
[ 4020.104499] 929946
[ 4020.104501] seconds:12
```

2. /proc/jiffies

```
osc@ubuntu:~/final-src-osc10e/ch2$ sudo insmod jiffies.ko
[sudo] password for osc:
osc@ubuntu:~/final-src-osc10e/ch2$ cat /proc/jiffies
4296320361
osc@ubuntu:~/final-src-osc10e/ch2$ sudo rmmod jiffies
```

3. /proc/seconds

```
osc@ubuntu:~/final-src-osc10e/ch2$ sudo insmod seconds.ko
osc@ubuntu:~/final-src-osc10e/ch2$ cat /proc/seconds
7
osc@ubuntu:~/final-src-osc10e/ch2$ sudo rmmod seconds
```