

Linux 内核实验指导

2021年3月

实验一 模块编程

实验要求

编写四个模块，分别实现以下功能：

- \1. 模块一，加载和卸载模块时在系统日志输出信息
- \2. 模块二，支持整型、字符串、数组参数，加载时读入并打印
- \3. 模块三，在/proc下创建只读文件
- \4. 模块四，在/proc下创建文件夹，并创建一个可读可写的文件

实验提示

- \1. 模块可以写在一个c文件中，模块参数传递参考宏定义 `module_param(name, type, perm)`，需要用到头文件 `linux/moduleparam.h`。
- \2. 编写Makefile文件将c源码编译成.ko的模块。
- \3. 模块下proc目录和文件的创建参考 `proc_mkdir` 和 `proc_create` 函数。

实现思路（仅供参考）

```
//<Makefile>
```

可参考PPT

```
//<module1.c>
//Test for installing and removing of module.
#include <linux/init.h>
#include <linux/module.h>
#include <linux/kernel.h>
static int __init hello_init(void){
    //TODO: 加载模块时printk输出信息
}

static void __exit hello_exit(void){
    //TODO: 卸载模块时printk输出信息
}
module_init(hello_init);
module_exit(hello_exit);
```

```
//<module2.c>
//Support for int&str&array parameter
#include <linux/module.h>
#include <linux/moduleparam.h>
#include <linux/kernel.h>

//TODO: 声明用于接收参数的变量
```

```
//TODO: 使用module_param来指定模块可接受的参数

static int __init hello_init(void)
{
    //TODO: 加载模块时printk输出信息，打印参数的值
}
static void __exit hello_exit(void)
{
    //TODO: 卸载模块时printk输出信息
}
module_init(hello_init);
module_exit(hello_exit);
```

```
//<module3.c>
//read-only proc file
#include <linux/module.h>
#include <linux/proc_fs.h>
#include <linux/seq_file.h>

static int hello_proc_open(struct inode *inode, struct file *file) {
    //定义文件操作
}

static const struct file_operations hello_proc_fops = {
    //TODO: 指定文件操作
};

static int __init hello_proc_init(void) {
    //TODO: 加载模块时printk输出信息
}

static void __exit hello_proc_exit(void) {
    //TODO: 卸载模块时printk输出信息，删除创建的proc文件
}

module_init(hello_proc_init);
module_exit(hello_proc_exit);
```

```
//<module4.c>
#include <linux/module.h>
#include <linux/init.h>
#include <linux/kernel.h>

#include <linux/fs.h>           // for basic filesystem
#include <linux/proc_fs.h>      // for the proc filesystem
#include <linux/seq_file.h>      // for sequence files
#include <linux/slab.h>          // for kzalloc, kfree
#include <linux/uaccess.h>       // for copy_from_user
#define BUF_SIZE 10

//TODO: 可以在这里声明需要的变量，注意static关键字的使用
```

```

// file_operations -> write
static ssize_t hello_write(struct file *file, const char __user *buffer, size_t
count, loff_t *f_pos)
{
    //TODO: 定义文件操作
}

// seq_operations -> open
static int hello_open(struct inode *inode, struct file *file)
{
    //TODO: 定义文件操作
}

static const struct file_operations hello_fops =
{
    //TODO: 指定文件操作
};

// module init
static int __init hello_init(void)
{
    //TODO: 创建proc目录、创建proc文件
}

// module exit
static void __exit hello_exit(void)
{
    //TODO: 删除proc目录、删除proc文件，如有手动分配内存空间，记得释放
}

module_init(hello_init);
module_exit(hello_exit);

```

```

# 实验效果测试可能用到的指令
make
sudo insmod module1.ko
dmesg
sudo insmod module2.ko int_var=666 str_var=hello int_array=10,20,30,40
dmesg
sudo insmod module3.ko
cat /proc/hello_proc
sudo insmod module4.ko
cat /proc/hello_dir/hello
sudo su
echo nice > /proc/hello_dir/hello
cat /proc/hello_dir/hello

```

注：以上模板仅供参考，也可以完全按照自己的思路来实现，实验报告中如果能清晰具体地写出自己的实验思路和具体实现会比较有亮点。

作业验收与提交

验收方式：电院3号楼118向助教展示实验效果

提交渠道：Canvas

提交文件："学号_姓名_project1.zip"

- 源码文件夹"学号_姓名_project1_src" (*.c Makefile)
- 实验报告《学号_姓名_project1_report.pdf》，包括但不限于实验过程、实验效果截图、实验心得。