Lab Report of Project 1: Introduction to Linux Code and Kernel Modules

---

Content

1. Design a kernel module and load it into the Linux kernel.
2. Modify the module allowing it to create an entry in the `\proc` file system.

Environment

1. Vmware Station 15
2. Ubuntu-19.04-desktop
3. kernel version: linux-5.3.2

MakeFile

To compile our modules, we have to add our files to the `Makefile`

```
obj-m:=jiffies.o
simplemodule-objs:=module
KDIR:=/lib/modules/$(shell uname -r)/build
MAKE:=make
default:
    $(MAKE) -C $(KDIR) M=$(shell pwd) modules
clean:
    $(MAKE) -C $(KDIR) M=$(shell pwd) clean
```

Basic commands

1. Load kernel modules: `sudo insmod ...`
2. Remove kernel modules:`sudo rmmod ...`
3. Check the contents of this message in the kernel log buffer, enter the command: `dmesg`

The `proc` File System

Upon request, we design two modules as `jiffies` and `second` to create two entries,`\proc\jiffies` and `\proc\second`.

The `jiffies` module reports the variable `jiffies` ,the number of timer interrupts that have occurred since the system was botted, directly. Here we only need to modify function `proc_read` in the text book.

```
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 kernel space buffer to user space usr buf */
    copy_to_user(usr_buf, buffer, rv);
    return rv;
}
```

The `second` module report the number of seconds that have elapsed since the kernel module was first loaded. The modification is similar as `jiffies`'s.

```
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-t)/HZ);

    /* copies kernel space buffer to user space usr buf */
    copy_to_user(usr_buf, buffer, rv);
    return rv;
}
```

The result

jiffies

```
gavin@ubuntu:~/Documents/Project1/jiffies$ sudo insmod jiffies.ko
gavin@ubuntu:~/Documents/Project1/jiffies$ cat /proc/jiffies
4294944095
gavin@ubuntu:~/Documents/Project1/jiffies$ cat /proc/jiffies
4294944588
gavin@ubuntu:~/Documents/Project1/jiffies$ sudo rmmod jiffies.ko
```

second

```
gavin@ubuntu:~/Documents/Project1/second$ sudo insmod second.ko
gavin@ubuntu:~/Documents/Project1/second$ cat /proc/second
7
gavin@ubuntu:~/Documents/Project1/second$ cat /proc/second
8
gavin@ubuntu:~/Documents/Project1/second$ sudo rmmod second.ko
```