OS Project-1 Report

Introduction to Linux Kernel Modules

Zhou Yiyuan

516021910270
1025152261@qq.com

1 Module: simple

1.1 Requirements

(a) Print out the value of GOLDEN RATIO PRIME in the simple init() function.

(b) Print out the greatest common divisor of 3,300 and 24 in the simple exit() function.

(c) Print out the values of jiffies and HZ in the simple init() function.

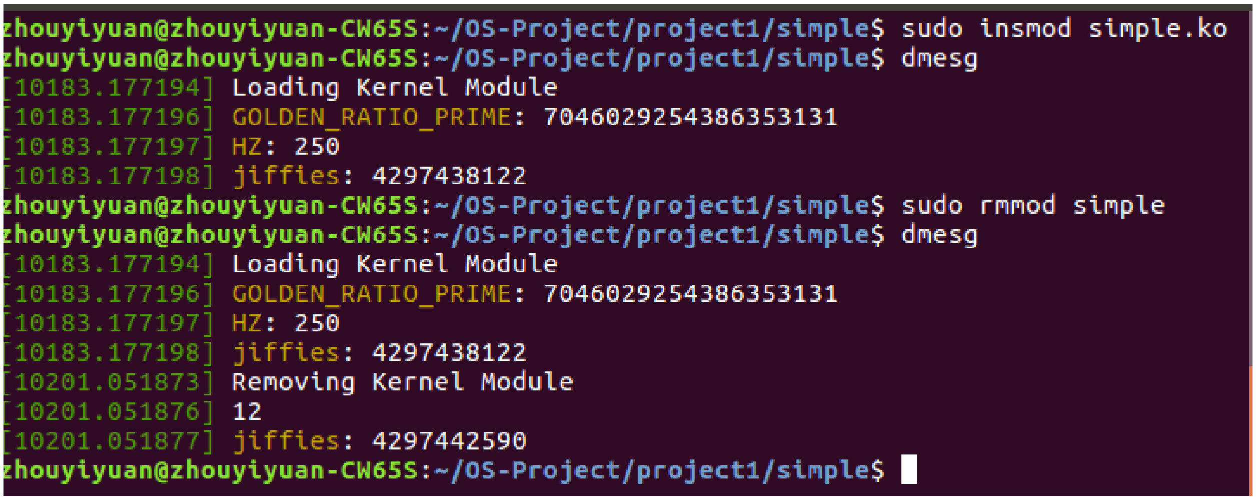
(d) Print out the value of jiffies in the simple exit() function.

1.2 Code

```c
#include <linux/init.h>
#include <linux/kernel.h>
#include <linux/module.h>
#include <linux/hash.h>
#include <linux/gcd.h>
#include <asm/param.h>
#include <linux/jiffies.h>

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

    return 0;
}

void simple_exit(void)
{
    printk(KERN_INFO "Removing Kernel Module\n");
    printk(KERN_INFO "%d\n", gcd(3300, 24));
    printk(KERN_INFO "jiffies: %lu\n", jiffies);
}

module_init(simple_init);
module_exit(simple_exit);

MODULE_LICENSE("GPL");
MODULE_DESCRIPTION("Simple Module");
MODULE_AUTHOR("ZYY");
```

1.3 Result

The result is shown as Figure 1.



```
zhouyiyuan@zhouyiyuan-CW65S:~/OS-Project/project1/simple$ sudo insmod simple.ko
zhouyiyuan@zhouyiyuan-CW65S:~/OS-Project/project1/simple$ dmesg
[10183.177194] Loading Kernel Module
[10183.177196] GOLDEN_RATIO_PRIME: 7046029254386353131
[10183.177197] HZ: 250
[10183.177198] jiffies: 4297438122
zhouyiyuan@zhouyiyuan-CW65S:~/OS-Project/project1/simple$ sudo rmmod simple
zhouyiyuan@zhouyiyuan-CW65S:~/OS-Project/project1/simple$ dmesg
[10183.177194] Loading Kernel Module
[10183.177196] GOLDEN_RATIO_PRIME: 7046029254386353131
[10183.177197] HZ: 250
[10183.177198] jiffies: 4297438122
[10201.051873] Removing Kernel Module
[10201.051876] 12
[10201.051877] jiffies: 4297442590
zhouyiyuan@zhouyiyuan-CW65S:~/OS-Project/project1/simple$ 
```

Figure 1: Result of simple module

2 Module: hello

2.1 Requirements

Print out ”Hello World” when the command ”cat /proc/hello” is entered.

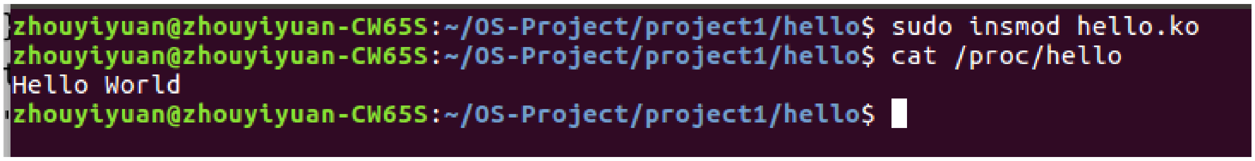
2.2 Code

```
#include <linux/init.h>
#include <linux/kernel.h>
#include <linux/module.h>
#include <linux/proc_fs.h>
#include <linux/uaccess.h>

#define BUFFER_SIZE 128
#define PROC_NAME "hello"

ssize_t proc_read(struct file *file, char __user *usr_buf, size_t count, loff_t
 ↪  *pos);

static struct file_operations proc_ops = {
    .owner = THIS_MODULE,
    .read = proc_read,
};
```

```
int proc_init(void)
{
    proc_create(PROC_NAME, 0666, NULL, &proc_ops);

    return 0;
}


void proc_exit(void)
{
    remove_proc_entry(PROC_NAME, NULL);
}

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, "Hello World\n");

    copy_to_user(usr_buf, buffer, rv);

    return rv;
}
module_init(proc_init);
module_exit(proc_exit);

MODULE_LICENSE("GPL");
MODULE_DESCRIPTION("HELLO MODULE");
MODULE_AUTHOR("ZYY");
```

2.3 Result

The result is shown as Figure 2.



```
zhouyiyuan@zhouyiyuan-CW65S:~/OS-Project/project1/hello$ sudo insmod hello.ko
zhouyiyuan@zhouyiyuan-CW65S:~/OS-Project/project1/hello$ cat /proc/hello
Hello World
zhouyiyuan@zhouyiyuan-CW65S:~/OS-Project/project1/hello$ 
```

Figure 2: Result of hello module

3 Module: jiffies

3.1 Requirements

Design a kernel module that creates a /proc file named /proc/jiffies that reports the current value of jiffies when the /proc/jiffies file is read, such as with the command

```
$ cat /proc/jiffies
```

Be sure to remove /proc/jiffies when the module is removed.

3.2 Code

```
#include <linux/init.h>
#include <linux/kernel.h>
#include <linux/module.h>
#include <linux/proc_fs.h>
#include <linux/uaccess.h>
#include <linux/jiffies.h>

#define BUFFER_SIZE 128
#define PROC_NAME "jiffies"

ssize_t proc_read(struct file *file, char __user *usr_buf, size_t count, loff_t
↪  *pos);

static struct file_operations proc_ops = {
    .owner = THIS_MODULE,
    .read = proc_read,
};

int proc_init(void)
{
    proc_create(PROC_NAME, 0666, NULL, &proc_ops);

    return 0;
}


void proc_exit(void)
```

```
{
    remove_proc_entry(PROC_NAME, NULL);
}

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, "jiffies: %lu\n", jiffies);

    copy_to_user(usr_buf, buffer, rv);

    return rv;
}
module_init(proc_init);
module_exit(proc_exit);

MODULE_LICENSE("GPL");
MODULE_DESCRIPTION("JIFFIES MODULE");
MODULE_AUTHOR("ZYY");
```

3.3 Result

The result is shown as Figure 3

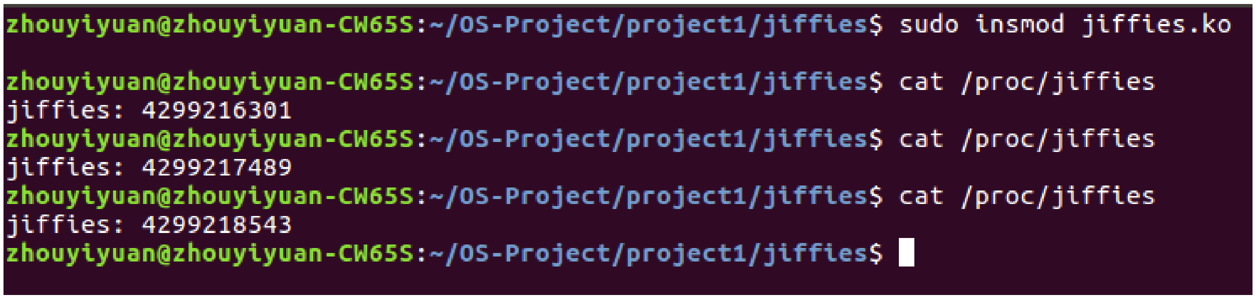
4 Module: seconds

4.1 Requirements

Design a kernel module that creates a proc file named /proc/seconds that reports the number of elapsed seconds since the kernel module was loaded. This will involve using the value of jiffies as well as the HZ rate. When a user enters the command

```
$ cat /proc/seconds
```

your kernel module will report the number of seconds that have elapsed since the kernel module was first loaded. Be sure to remove /proc/seconds when the module is removed.



```
zhouyiyuan@zhouyiyuan-CW65S:~/OS-Project/project1/jiffies$ sudo insmod jiffies.ko

zhouyiyuan@zhouyiyuan-CW65S:~/OS-Project/project1/jiffies$ cat /proc/jiffies
jiffies: 4299216301
zhouyiyuan@zhouyiyuan-CW65S:~/OS-Project/project1/jiffies$ cat /proc/jiffies
jiffies: 4299217489
zhouyiyuan@zhouyiyuan-CW65S:~/OS-Project/project1/jiffies$ cat /proc/jiffies
jiffies: 4299218543
zhouyiyuan@zhouyiyuan-CW65S:~/OS-Project/project1/jiffies$ 
```

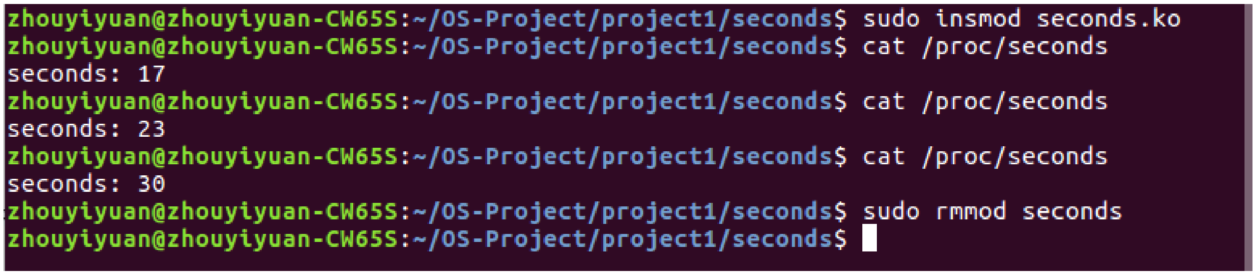
Figure 3: Result of jiffies module

4.2 Code

```c
#include <linux/init.h>
#include <linux/kernel.h>
#include <linux/module.h>
#include <linux/proc_fs.h>
#include <linux/uaccess.h>
#include <linux/jiffies.h>
#include <asm/param.h>

#define BUFFER_SIZE 128
#define PROC_NAME "seconds"

ssize_t proc_read(struct file *file, char __user *usr_buf, size_t count, loff_t
↪ *pos);

long unsigned init_jiffies = 0;

static struct file_operations proc_ops = {
    .owner = THIS_MODULE,
    .read = proc_read,
};

int proc_init(void)
{
    init_jiffies = jiffies;
    proc_create(PROC_NAME, 0666, NULL, &proc_ops);

    return 0;
}


void proc_exit(void)
{
    remove_proc_entry(PROC_NAME, NULL);
}

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, "seconds: %lu\n", (jiffies - init_jiffies) / HZ);

    copy_to_user(usr_buf, buffer, rv);

    return rv;
}
module_init(proc_init);
module_exit(proc_exit);

MODULE_LICENSE("GPL");
MODULE_DESCRIPTION("SECONDS MODULE");
MODULE_AUTHOR("ZYY");
```

4.3 Result

The result is shown as Figure 4



```
zhouyiyuan@zhouyiyuan-CW65S:~/OS-Project/project1/seconds$ sudo insmod seconds.ko
zhouyiyuan@zhouyiyuan-CW65S:~/OS-Project/project1/seconds$ cat /proc/seconds
seconds: 17
zhouyiyuan@zhouyiyuan-CW65S:~/OS-Project/project1/seconds$ cat /proc/seconds
seconds: 23
zhouyiyuan@zhouyiyuan-CW65S:~/OS-Project/project1/seconds$ cat /proc/seconds
seconds: 30
zhouyiyuan@zhouyiyuan-CW65S:~/OS-Project/project1/seconds$ sudo rmmod seconds
zhouyiyuan@zhouyiyuan-CW65S:~/OS-Project/project1/seconds$ 
```

Figure 4: Result of seconds module