操作系统(D)

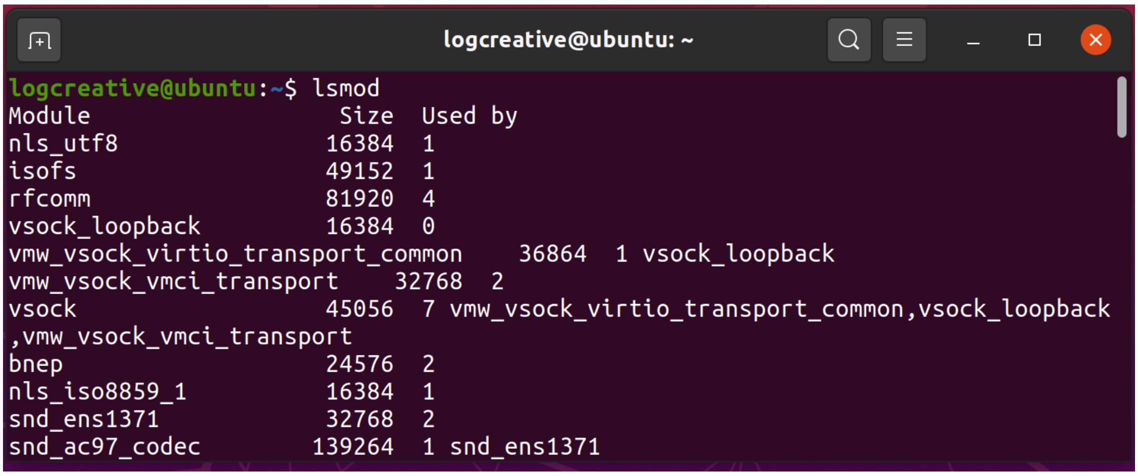
项目 1

李子龙 518070910095

2021 年 3 月 13 日

一　创建内核模块

1. 列出当前加载的所有内核模块。



```
logcreative@ubuntu:~$ lsmod
Module                  Size  Used by
nls_utf8               16384  1
isofs                  49152  1
rfcomm                 81920  4
vsock_loopback         16384  0
vmw_vsock_virtio_transport_common    36864  1 vsock_loopback
vmw_vsock_vmci_transport    32768  2
vsock                  45056  7 vmw_vsock_virtio_transport_common,vsock_loopback
,vmw_vsock_vmci_transport
bnep                   24576  2
nls_iso8859_1          16384  1
snd_ens1371            32768  2
snd_ac97_codec        139264  1 snd_ens1371
```

2. 编写模块以输出提示：

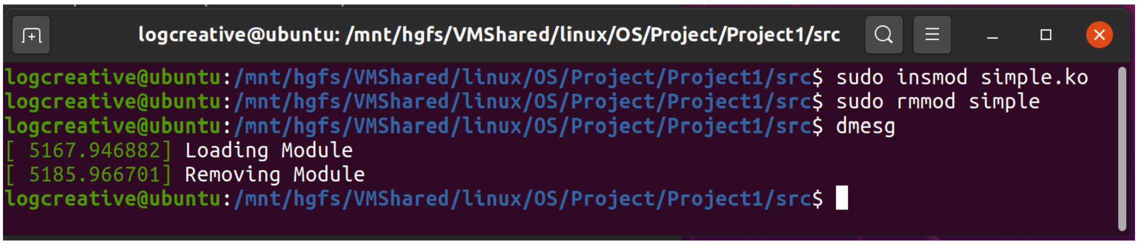
```
int simple_init(void){
    printk(KERN_INFO "Loading Module\n");
    return 0;
}

void simple_exit(void){
    printk(KERN_INFO "Removing Module\n");
}
```

使用 MakeFile 编译：

Listing 1: src/Makefile

```
obj-m:=simple.o hello.o jiffies.o seconds.o
KDIR:=/lib/modules/$(shell uname -r)/build
PWD:=$(shell pwd)

all:
    make -C $(KDIR) M=$(PWD) modules
clean:
    rm *.o *.ko *.mod.c Modules.symvers modules.order -f
```

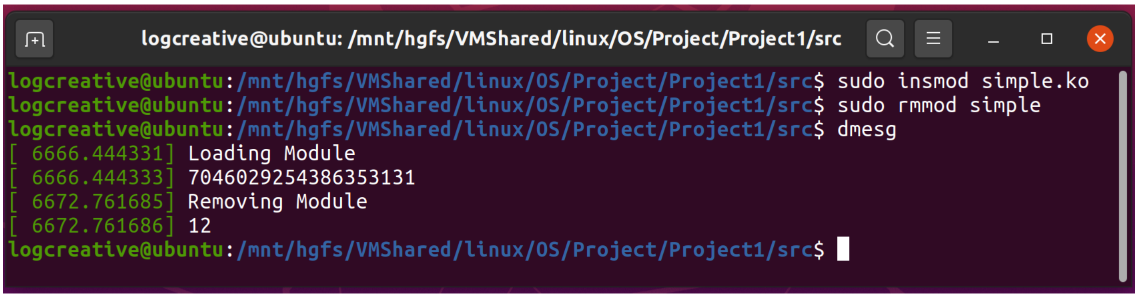
3. 加载与卸载内核模块


logcreative@ubuntu: /mnt/hgfs/VMShared/linux/OS/Project/Project1/src

```
logcreative@ubuntu:/mnt/hgfs/VMShared/linux/OS/Project/Project1/src$ sudo insmod simple.ko
logcreative@ubuntu:/mnt/hgfs/VMShared/linux/OS/Project/Project1/src$ sudo rmmod simple
logcreative@ubuntu:/mnt/hgfs/VMShared/linux/OS/Project/Project1/src$ dmesg
[ 5167.946882] Loading Module
[ 5185.966701] Removing Module
logcreative@ubuntu:/mnt/hgfs/VMShared/linux/OS/Project/Project1/src$ 
```

添加了两行代码用于打印 GOLDEN_RATIO_PRIME 和 3300 与 24 的最大公因数：

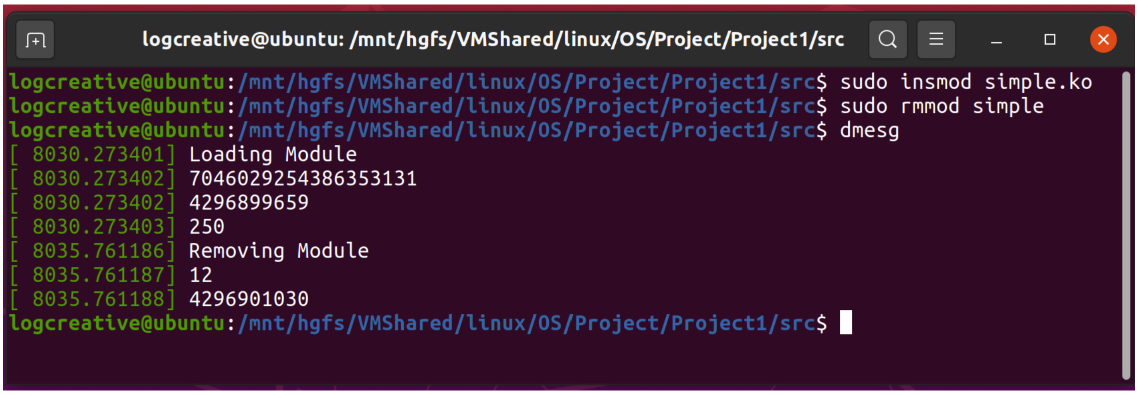
```
// @ simple_init(void)
printk(KERN_INFO "%lu\n", GOLDEN_RATIO_PRIME);

// @ simple_exit(void)
printk(KERN_INFO "%lu\n", gcd(3300,24));
```


logcreative@ubuntu: /mnt/hgfs/VMShared/linux/OS/Project/Project1/src

```
logcreative@ubuntu:/mnt/hgfs/VMShared/linux/OS/Project/Project1/src$ sudo insmod simple.ko
logcreative@ubuntu:/mnt/hgfs/VMShared/linux/OS/Project/Project1/src$ sudo rmmod simple
logcreative@ubuntu:/mnt/hgfs/VMShared/linux/OS/Project/Project1/src$ dmesg
[ 6666.444331] Loading Module
[ 6666.444333] 7046029254386353131
[ 6672.761685] Removing Module
[ 6672.761686] 12
logcreative@ubuntu:/mnt/hgfs/VMShared/linux/OS/Project/Project1/src$ 
```

继续添加代码用于打印 jiffies 和 HZ。

```
// @ simple_init(void)
printk(KERN_INFO "%lu\n", jiffies);
printk(KERN_INFO "%u\n", HZ);

// @ simple_exit(void)
printk(KERN_INFO "%lu\n", jiffies);
```


logcreative@ubuntu: /mnt/hgfs/VMShared/linux/OS/Project/Project1/src

```
logcreative@ubuntu:/mnt/hgfs/VMShared/linux/OS/Project/Project1/src$ sudo insmod simple.ko
logcreative@ubuntu:/mnt/hgfs/VMShared/linux/OS/Project/Project1/src$ sudo rmmod simple
logcreative@ubuntu:/mnt/hgfs/VMShared/linux/OS/Project/Project1/src$ dmesg
[ 8030.273401] Loading Module
[ 8030.273402] 7046029254386353131
[ 8030.273402] 4296899659
[ 8030.273403] 250
[ 8035.761186] Removing Module
[ 8035.761187] 12
[ 8035.761188] 4296901030
logcreative@ubuntu:/mnt/hgfs/VMShared/linux/OS/Project/Project1/src$ 
```

最后该部分所有的代码如下：

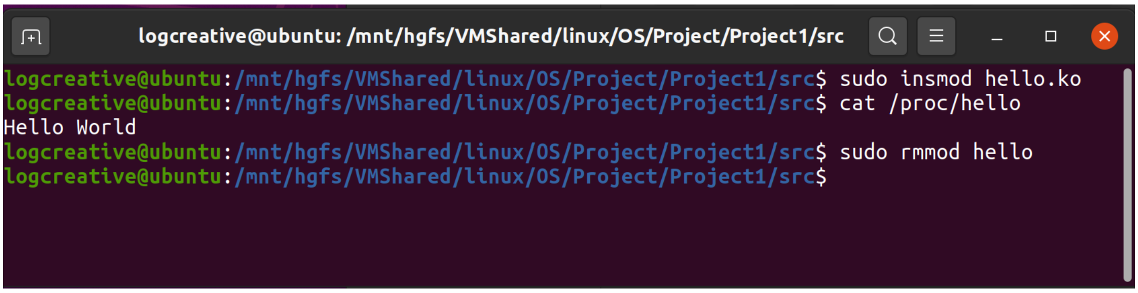
Listing 2: src/simple.c

```
#include<linux/kernel.h>
#include<linux/module.h>
#include<linux/init.h>
#include<linux/hash.h>
#include<linux/gcd.h>
```

```
#include<linux/jiffies.h>

int simple_init(void){
    printk(KERN_INFO "Loading Module\n");
    printk(KERN_INFO "%lu\n", GOLDEN_RATIO_PRIME);
    printk(KERN_INFO "%lu\n", jiffies);
    printk(KERN_INFO "%u\n", HZ);
    return 0;
}

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

module_init(simple_init);
module_exit(simple_exit);

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

二 \proc 文件系统

使用 \proc 文件系统打印 Hello World。



logcreative@ubuntu: /mnt/hgfs/VMShared/linux/OS/Project/Project1/src

```
logcreative@ubuntu:/mnt/hgfs/VMShared/linux/OS/Project/Project1/src$ sudo insmod hello.ko
logcreative@ubuntu:/mnt/hgfs/VMShared/linux/OS/Project/Project1/src$ cat /proc/hello
Hello World
logcreative@ubuntu:/mnt/hgfs/VMShared/linux/OS/Project/Project1/src$ sudo rmmod hello
logcreative@ubuntu:/mnt/hgfs/VMShared/linux/OS/Project/Project1/src$
```

由于采用了 Ubuntu 20.04 系统，所以使用的是 Linux 内核，因此需要使用 proc_ops 结构体来传入 proc_create 的第四参数。

Listing 3: src/hello.c

```
#include <linux/init.h>
#include <linux/kernel.h>
#include <linux/module.h>
#include <linux/proc_fs.h>
#include <asm/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 proc_ops proc_ops = {
    .proc_read = proc_read
};
```

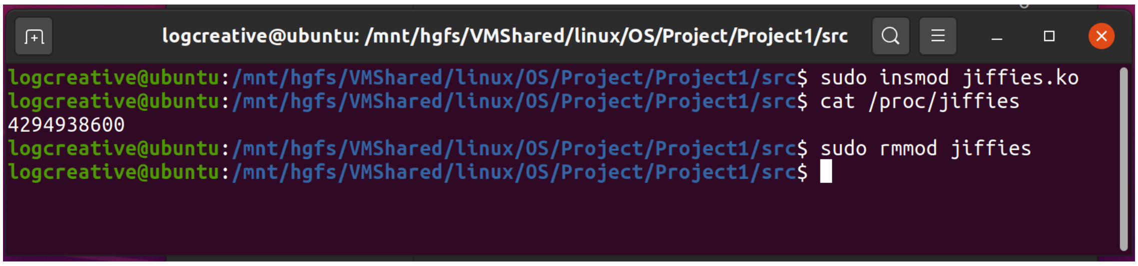
```
int proc_init(void){
    proc_create(PROC_NAME, 0666, NULL, &proc_ops);
    return 0;
}

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("LogCreative");
```

三 作业

1. 使用 \proc 打印 jiffies。

   最重要的部分是更改了 sprintf 的所在行。

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



logcreative@ubuntu: /mnt/hgfs/VMShared/linux/OS/Project/Project1/src

```
logcreative@ubuntu:/mnt/hgfs/VMShared/linux/OS/Project/Project1/src$ sudo insmod jiffies.ko
logcreative@ubuntu:/mnt/hgfs/VMShared/linux/OS/Project/Project1/src$ cat /proc/jiffies
4294938600
logcreative@ubuntu:/mnt/hgfs/VMShared/linux/OS/Project/Project1/src$ sudo rmmod jiffies
logcreative@ubuntu:/mnt/hgfs/VMShared/linux/OS/Project/Project1/src$ 
```

Listing 4: src/jiffies.c

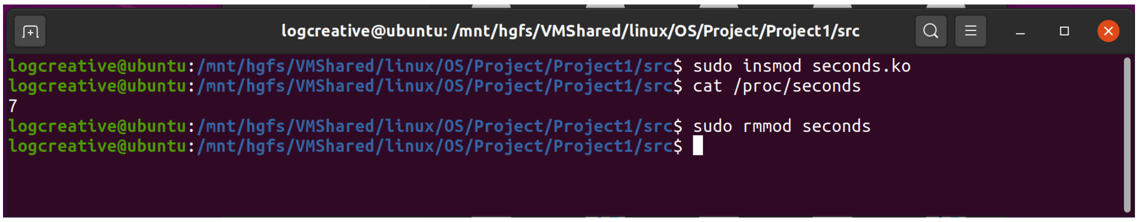
```
#include <linux/init.h>
#include <linux/kernel.h>
```

```c
#include <linux/module.h>
#include <linux/proc_fs.h>
#include <asm/uaccess.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 proc_ops proc_ops = {
    .proc_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,"%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("LogCreative");
```

2. 使用 \proc 打印模块运行秒数。

   初始时初始化 init_jiff 变量，秒数由 HZ 计算得到：

$$\texttt{seconds} = \frac{\texttt{jiffies} - \texttt{init\_jiff}}{\texttt{HZ}}$$



logcreative@ubuntu: /mnt/hgfs/VMShared/linux/OS/Project/Project1/src

```
logcreative@ubuntu:/mnt/hgfs/VMShared/linux/OS/Project/Project1/src$ sudo insmod seconds.ko
logcreative@ubuntu:/mnt/hgfs/VMShared/linux/OS/Project/Project1/src$ cat /proc/seconds
7
logcreative@ubuntu:/mnt/hgfs/VMShared/linux/OS/Project/Project1/src$ sudo rmmod seconds
logcreative@ubuntu:/mnt/hgfs/VMShared/linux/OS/Project/Project1/src$ 
```

Listing 5: src/seconds.c

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

static struct proc_ops proc_ops = {
    .proc_read = proc_read
};

unsigned long init_jiff;

int proc_init(void){
    proc_create(PROC_NAME, 0666, NULL, &proc_ops);
    init_jiff = jiffies;
    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;
    unsigned long diffjiff = jiffies - init_jiff;
    int seconds = diffjiff / HZ;
    rv = sprintf(buffer,"%d\n",seconds);

    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("LogCreative");
```