Environment：Linux

Instructions：

1. make

2. lsmod to check existing linux kernel

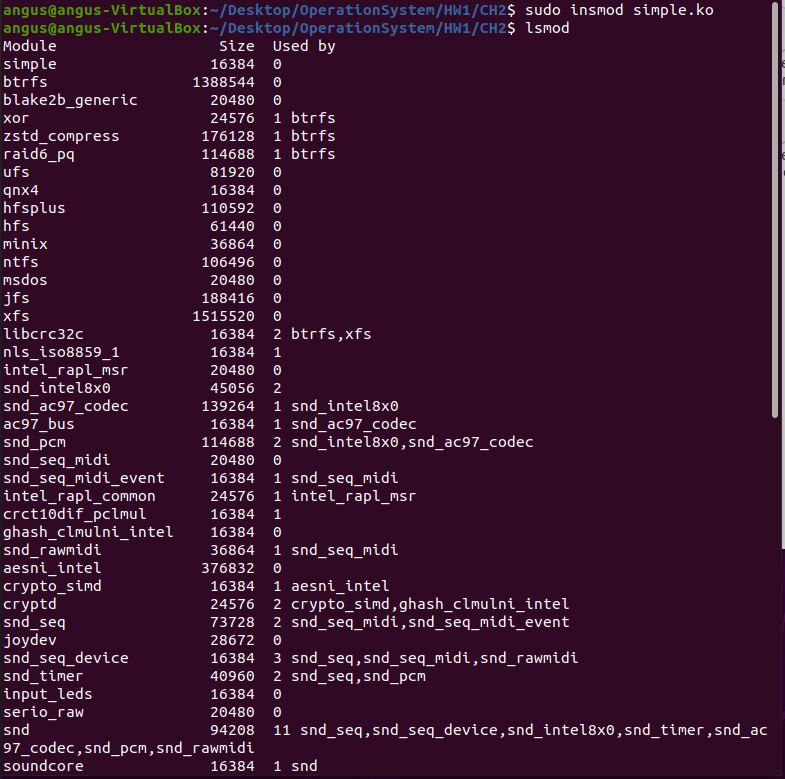
3. sudo insmod simple.ko

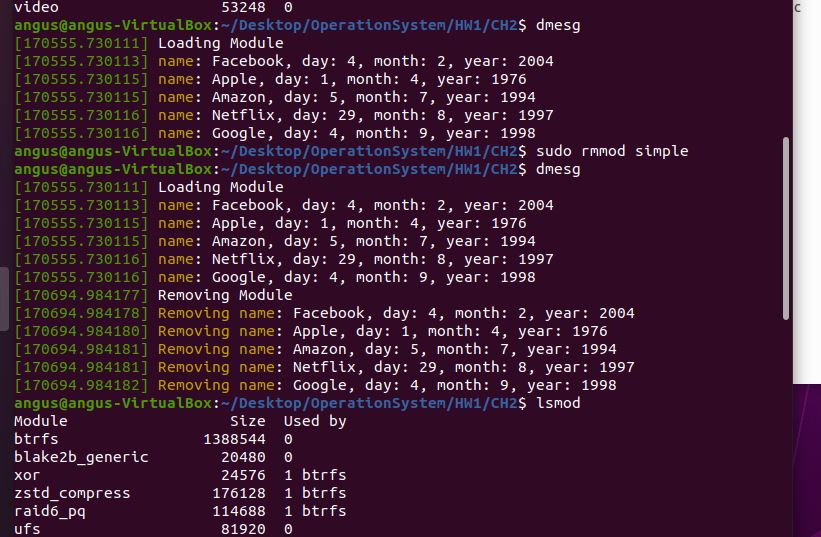
4. lsmod check whether the kernel has been loaded

5. dmesg to print content of the message in the kernel log buffer

6. sudo rmmod simple to remove kernel module

Load kernel



Remove kernel

Source code：

#include <linux/init.h>

#include <linux/module.h>

#include <linux/kernel.h>

#include <linux/slab.h>

#include <linux/list.h>

static LIST\_HEAD(birthday\_list);

struct birthday

{

char \*name;

int day;

int month;

int year;

struct list\_head list;

};

void set\_birthday(struct birthday \*person, char \*name, int day, int month, int year);

/\* This function is called when the module is loaded. \*/

int simple\_init(void)

{

struct birthday \*Facebook, \*Apple, \*Amazon, \*Netflix, \*Google;

struct birthday \*ptr;

printk(KERN\_INFO "Loading Module\n");

Facebook = kmalloc(sizeof(\*Facebook), GFP\_KERNEL);

Apple = kmalloc(sizeof(\*Apple), GFP\_KERNEL);

Amazon = kmalloc(sizeof(\*Amazon), GFP\_KERNEL);

Netflix = kmalloc(sizeof(\*Netflix), GFP\_KERNEL);

Google = kmalloc(sizeof(\*Google), GFP\_KERNEL);

set\_birthday(Facebook, "Facebook", 4, 2, 2004);

set\_birthday(Apple, "Apple", 1, 4, 1976);

set\_birthday(Amazon, "Amazon", 5, 7, 1994);

set\_birthday(Netflix, "Netflix", 29, 8, 1997);

set\_birthday(Google, "Google", 4, 9, 1998);

list\_for\_each\_entry(ptr, &birthday\_list, list)

{

printk(KERN\_INFO "name: %s, day: %d, month: %d, year: %d \n", ptr->name, ptr->day, ptr->month, ptr->year);

}

return 0;

}

void set\_birthday(struct birthday \*person, char \*name, int day, int month, int year)

{

person->name = name;

person->day = day;

person->month = month;

person->year = year;

INIT\_LIST\_HEAD(&person->list);

list\_add\_tail(&person->list, &birthday\_list);

}

/\* This function is called when the module is removed. \*/

void simple\_exit(void)

{

struct birthday \*ptr, \*next;

printk(KERN\_INFO "Removing Module\n");

list\_for\_each\_entry\_safe(ptr, next, &birthday\_list, list)

{

printk(KERN\_INFO "Removing name: %s, day: %d, month: %d, year: %d \n", ptr->name, ptr->day, ptr->month, ptr->year);

list\_del(&ptr->list);

kfree(ptr);

}

}

/\* Macros for registering module entry and exit points. \*/

module\_init(simple\_init);

module\_exit(simple\_exit);

MODULE\_LICENSE("GPL");

MODULE\_DESCRIPTION("Simple Module");

MODULE\_AUTHOR("SGG");