School of Computer and Information Science

《**Operating-System-Concepts(10th)**》

Project Report

Student information：

Name：　　　张乐之　　　　　　　　　　　ID:　　222022321102120　　　　
Major/Grade:　　　2022 计科中外 04　　　　　　　　Time:　　　2024-10-17

Experiment information：

Topic:

Introduction to Linux Kernel modules

Requirements:

1. Understand the basic concepts and organizational structure of the operating system, understand the characteristics of linux operating system, the user interface of linux operating system, and the general shell commands.
2. Master the loading / unloading method of linux kernel module and understand the development of operating system.

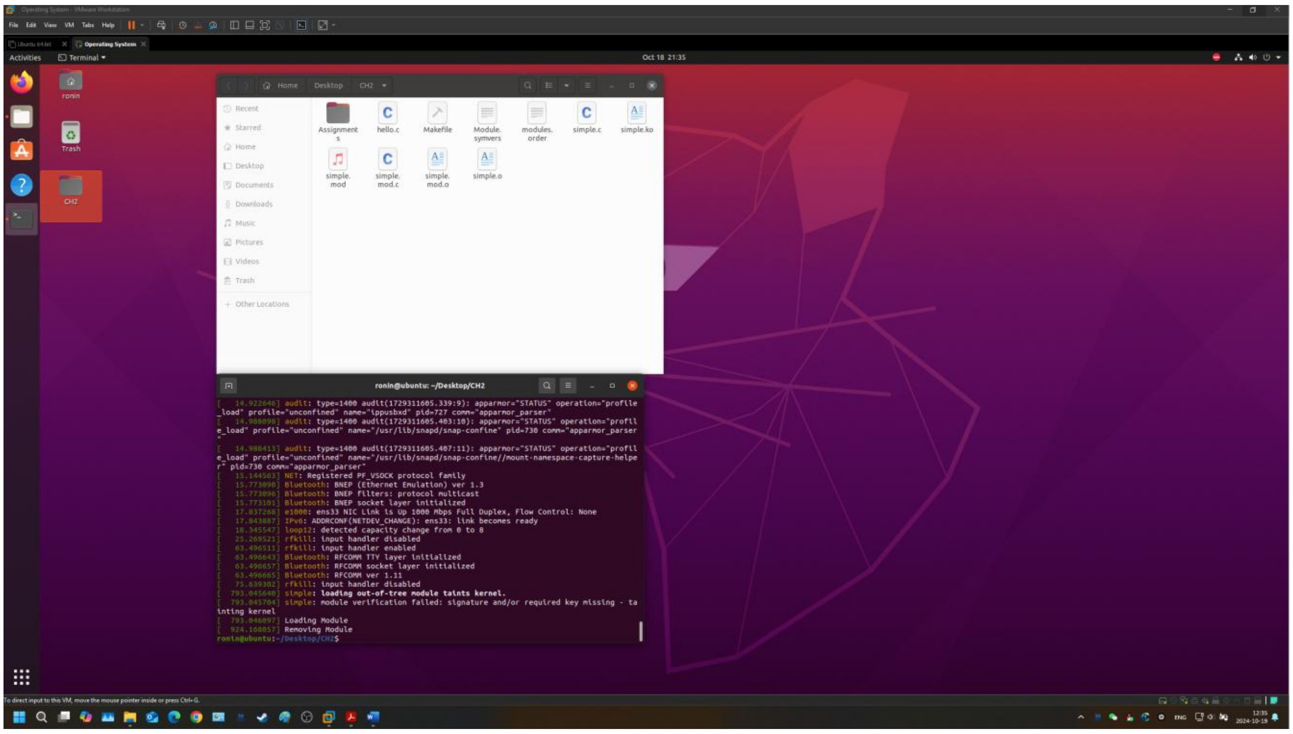
Procedure：

1. Write a kernel module.
2. Compile the kernel module with ‘make’ command.
3. Load the kernel module.
4. Remove the kernel module.

Results：

（Code and Figures）

1. For PART I, we compiled, loaded and removed a simple module, given the source codes and makefile:

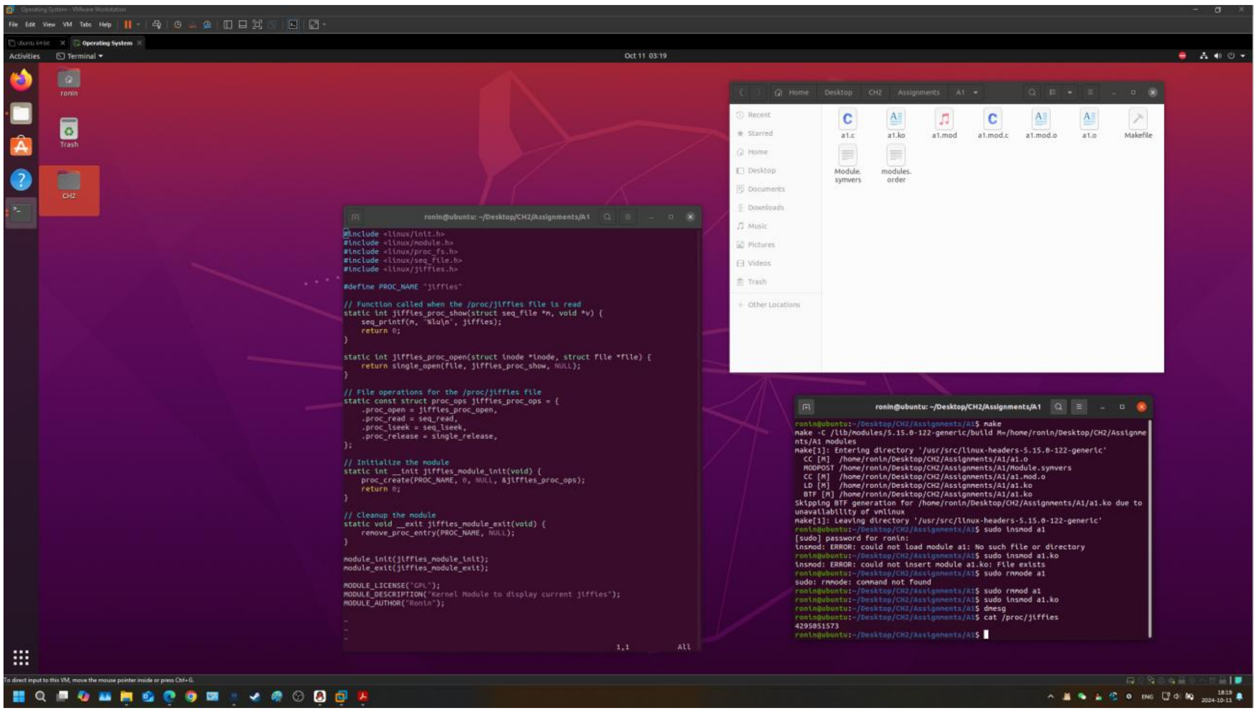


```
[   14.922646] audit: type=1400 audit(1729311605.339:9): apparmor="STATUS" operation="profile
_load" profile="unconfined" name="ippusbxd" pid=727 comm="apparmor_parser"
[   14.988098] audit: type=1400 audit(1729311605.403:10): apparmor="STATUS" operation="profil
e_load" profile="unconfined" name="/usr/lib/snapd/snap-confine" pid=730 comm="apparmor_parser
"
[   14.988413] audit: type=1400 audit(1729311605.407:11): apparmor="STATUS" operation="profil
e_load" profile="unconfined" name="/usr/lib/snapd/snap-confine//mount-namespace-capture-helpe
r" pid=730 comm="apparmor_parser"
[   15.144563] NET: Registered PF_VSOCK protocol family
[   15.773090] Bluetooth: BNEP (Ethernet Emulation) ver 1.3
[   15.773096] Bluetooth: BNEP filters: protocol multicast
[   15.773101] Bluetooth: BNEP socket layer initialized
[   17.837268] e1000: ens33 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: None
[   17.843887] IPv6: ADDRCONF(NETDEV_CHANGE): ens33: link becomes ready
[   18.345547] loop12: detected capacity change from 0 to 8
[   25.269521] rfkill: input handler disabled
[   63.496511] rfkill: input handler enabled
[   63.496643] Bluetooth: RFCOMM TTY layer initialized
[   63.496657] Bluetooth: RFCOMM socket layer initialized
[   63.496665] Bluetooth: RFCOMM ver 1.11
[   75.639302] rfkill: input handler disabled
[  793.045640] simple: loading out-of-tree module taints kernel.
[  793.045704] simple: module verification failed: signature and/or required key missing - ta
inting kernel
[  793.046097] Loading Module
[  924.168057] Removing Module
ronin@ubuntu:~/Desktop/CH2$
```

2. For PART II, we have 2 tasks. The first is to 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. My codes are as follows:

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

#define PROC_NAME "jiffies"

// Function called when the /proc/jiffies file is read
static int jiffies_proc_show(struct seq_file *m, void *v) {
    seq_printf(m, "%lu\n", jiffies);
    return 0;
}

static int jiffies_proc_open(struct inode *inode, struct file *file) {
    return single_open(file, jiffies_proc_show, NULL);
}

// File operations for the /proc/jiffies file
static const struct proc_ops jiffies_proc_ops = {
    .proc_open = jiffies_proc_open,
    .proc_read = seq_read,
    .proc_lseek = seq_lseek,
    .proc_release = single_release,
};

// Initialize the module
static int __init jiffies_module_init(void) {
    proc_create(PROC_NAME, 0, NULL, &jiffies_proc_ops);
    return 0;
}

// Cleanup the module
static void __exit jiffies_module_exit(void) {
    remove_proc_entry(PROC_NAME, NULL);
}

module_init(jiffies_module_init);
module_exit(jiffies_module_exit);

MODULE_LICENSE("GPL");
MODULE_DESCRIPTION("Kernel Module to display current jiffies");
MODULE_AUTHOR("Ronin");
```

After coding, we have to adjust the make file and compile it. After compiling, we can load it to the kernel and try find that file:



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

#define PROC_NAME "jiffies"

// Function called when the /proc/jiffies file is read
static int jiffies_proc_show(struct seq_file *m, void *v) {
    seq_printf(m, "%lu\n", jiffies);
    return 0;
}

static int jiffies_proc_open(struct inode *inode, struct file *file) {
    return single_open(file, jiffies_proc_show, NULL);
}

// File operations for the /proc/jiffies file
static const struct proc_ops jiffies_proc_ops = {
    .proc_open = jiffies_proc_open,
    .proc_read = seq_read,
    .proc_lseek = seq_lseek,
    .proc_release = single_release,
};

// Initialize the module
static int __init jiffies_module_init(void) {
    proc_create(PROC_NAME, 0, NULL, &jiffies_proc_ops);
    return 0;
}

// Cleanup the module
static void __exit jiffies_module_exit(void) {
    remove_proc_entry(PROC_NAME, NULL);
}

module_init(jiffies_module_init);
module_exit(jiffies_module_exit);

MODULE_LICENSE("GPL");
MODULE_DESCRIPTION("Kernel Module to display current jiffies");
MODULE_AUTHOR("Ronin");
~
~
~
                                                  1,1           All
```

```
ronin@ubuntu:~/Desktop/CH2/Assignments/A1$ make
make -C /lib/modules/5.15.0-122-generic/build M=/home/ronin/Desktop/CH2/Assignme
nts/A1 modules
make[1]: Entering directory '/usr/src/linux-headers-5.15.0-122-generic'
  CC [M]  /home/ronin/Desktop/CH2/Assignments/A1/a1.o
  MODPOST /home/ronin/Desktop/CH2/Assignments/A1/Module.symvers
  CC [M]  /home/ronin/Desktop/CH2/Assignments/A1/a1.mod.o
  LD [M]  /home/ronin/Desktop/CH2/Assignments/A1/a1.ko
  BTF [M] /home/ronin/Desktop/CH2/Assignments/A1/a1.ko
Skipping BTF generation for /home/ronin/Desktop/CH2/Assignments/A1/a1.ko due to
unavailability of vmlinux
make[1]: Leaving directory '/usr/src/linux-headers-5.15.0-122-generic'
ronin@ubuntu:~/Desktop/CH2/Assignments/A1$ sudo insmod a1
[sudo] password for ronin:
insmod: ERROR: could not load module a1: No such file or directory
ronin@ubuntu:~/Desktop/CH2/Assignments/A1$ sudo insmod a1.ko
insmod: ERROR: could not insert module a1.ko: File exists
ronin@ubuntu:~/Desktop/CH2/Assignments/A1$ sudo rmmode a1
sudo: rmmode: command not found
ronin@ubuntu:~/Desktop/CH2/Assignments/A1$ sudo rmmod a1
ronin@ubuntu:~/Desktop/CH2/Assignments/A1$ sudo insmod a1.ko
ronin@ubuntu:~/Desktop/CH2/Assignments/A1$ dmesg
ronin@ubuntu:~/Desktop/CH2/Assignments/A1$ cat /proc/jiffies
4295851573
ronin@ubuntu:~/Desktop/CH2/Assignments/A1$
```

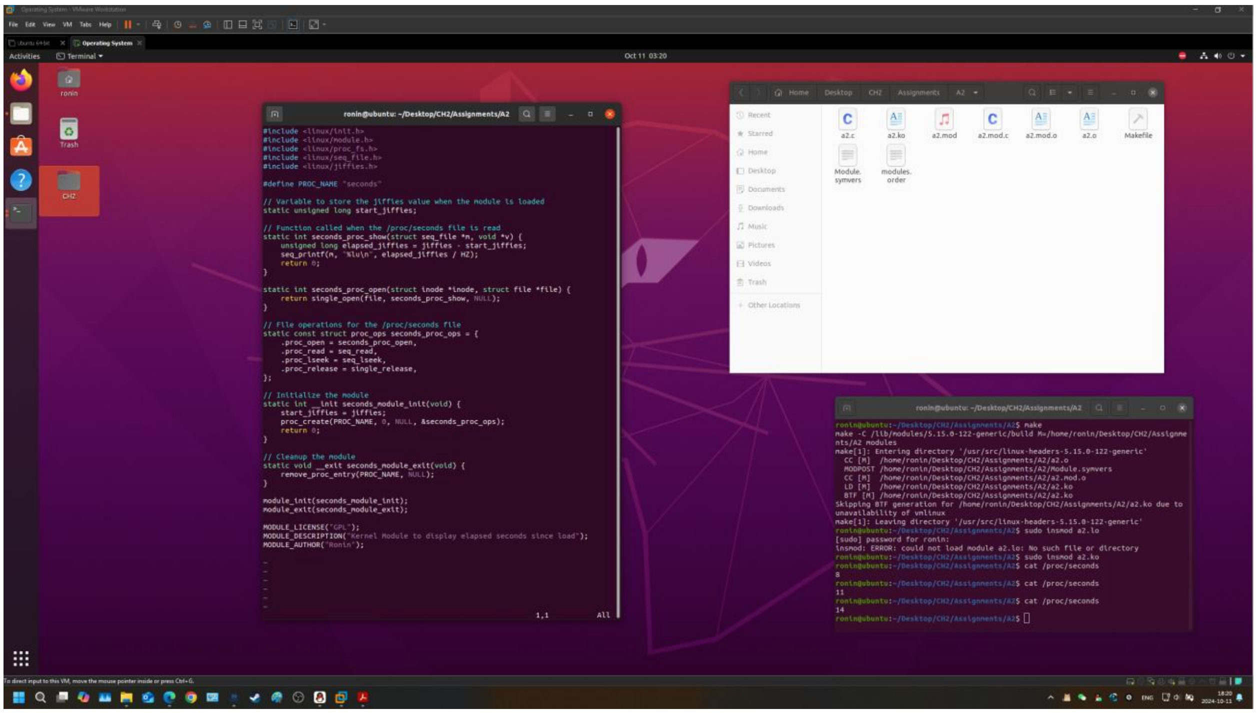
3. The second task is to count the seconds since the module is loaded. Codes are as follows:

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

```
#define PROC_NAME "seconds"

// Variable to store the jiffies value when the module is loaded
static unsigned long start_jiffies;

// Function called when the /proc/seconds file is read
static int seconds_proc_show(struct seq_file *m, void *v) {
    unsigned long elapsed_jiffies = jiffies - start_jiffies;
    seq_printf(m, "%lu\n", elapsed_jiffies / HZ);
    return 0;
}

static int seconds_proc_open(struct inode *inode, struct file *file) {
    return single_open(file, seconds_proc_show, NULL);
}

// File operations for the /proc/seconds file
static const struct proc_ops seconds_proc_ops = {
    .proc_open = seconds_proc_open,
    .proc_read = seq_read,
    .proc_lseek = seq_lseek,
    .proc_release = single_release,
};

// Initialize the module
static int __init seconds_module_init(void) {
    start_jiffies = jiffies;
    proc_create(PROC_NAME, 0, NULL, &seconds_proc_ops);
    return 0;
}

// Cleanup the module
static void __exit seconds_module_exit(void) {
    remove_proc_entry(PROC_NAME, NULL);
}

module_init(seconds_module_init);
module_exit(seconds_module_exit);

MODULE_LICENSE("GPL");
MODULE_DESCRIPTION("Kernel Module to display elapsed seconds since load");
MODULE_AUTHOR("Ronin");
```

As mentioned above, I modified the makefile and compiled it. After loading the module, we can find that each time I inspect the file, the content changes with time.

Results:



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

#define PROC_NAME "seconds"

// Variable to store the jiffies value when the module is loaded
static unsigned long start_jiffies;

// Function called when the /proc/seconds file is read
static int seconds_proc_show(struct seq_file *m, void *v) {
    unsigned long elapsed_jiffies = jiffies - start_jiffies;
    seq_printf(m, "%lu\n", elapsed_jiffies / HZ);
    return 0;
}

static int seconds_proc_open(struct inode *inode, struct file *file) {
    return single_open(file, seconds_proc_show, NULL);
}

// File operations for the /proc/seconds file
static const struct proc_ops seconds_proc_ops = {
    .proc_open = seconds_proc_open,
    .proc_read = seq_read,
    .proc_lseek = seq_lseek,
    .proc_release = single_release,
};

// Initialize the module
static int __init seconds_module_init(void) {
    start_jiffies = jiffies;
    proc_create(PROC_NAME, 0, NULL, &seconds_proc_ops);
    return 0;
}

// Cleanup the module
static void __exit seconds_module_exit(void) {
    remove_proc_entry(PROC_NAME, NULL);
}

module_init(seconds_module_init);
module_exit(seconds_module_exit);

MODULE_LICENSE("GPL");
MODULE_DESCRIPTION("Kernel Module to display elapsed seconds since load");
MODULE_AUTHOR("Ronin");

~
~
~
~
~
~
                                                    1,1           All
```

```
ronin@ubuntu:~/Desktop/CH2/Assignments/A2$ make
make -C /lib/modules/5.15.0-122-generic/build M=/home/ronin/Desktop/CH2/Assignme
nts/A2 modules
make[1]: Entering directory '/usr/src/linux-headers-5.15.0-122-generic'
  CC [M]  /home/ronin/Desktop/CH2/Assignments/A2/a2.o
  MODPOST /home/ronin/Desktop/CH2/Assignments/A2/Module.symvers
  CC [M]  /home/ronin/Desktop/CH2/Assignments/A2/a2.mod.o
  LD [M]  /home/ronin/Desktop/CH2/Assignments/A2/a2.ko
  BTF [M] /home/ronin/Desktop/CH2/Assignments/A2/a2.ko
Skipping BTF generation for /home/ronin/Desktop/CH2/Assignments/A2/a2.ko due to
unavailability of vmlinux
make[1]: Leaving directory '/usr/src/linux-headers-5.15.0-122-generic'
ronin@ubuntu:~/Desktop/CH2/Assignments/A2$ sudo insmod a2.lo
[sudo] password for ronin:
insmod: ERROR: could not load module a2.lo: No such file or directory
ronin@ubuntu:~/Desktop/CH2/Assignments/A2$ sudo insmod a2.ko
ronin@ubuntu:~/Desktop/CH2/Assignments/A2$ cat /proc/seconds
8
ronin@ubuntu:~/Desktop/CH2/Assignments/A2$ cat /proc/seconds
11
ronin@ubuntu:~/Desktop/CH2/Assignments/A2$ cat /proc/seconds
14
ronin@ubuntu:~/Desktop/CH2/Assignments/A2$ 
```