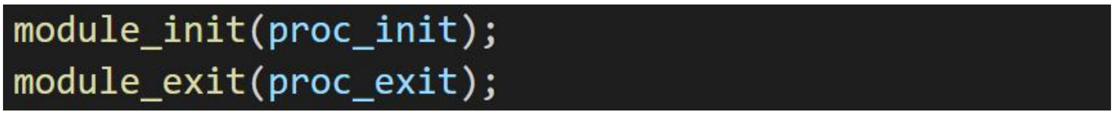
Assignment1 – *jiffies.c* and *seconds.c* kernel modules

1930026143　　薛劭杰

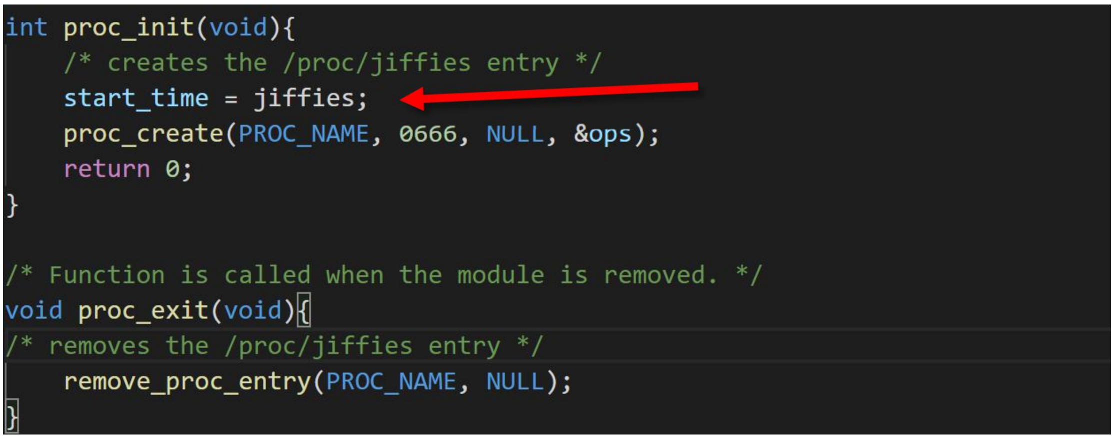
a. How to solve question:

First, we should define the functions which can be called when module is loaded into the kernel and is removed from the kernel.



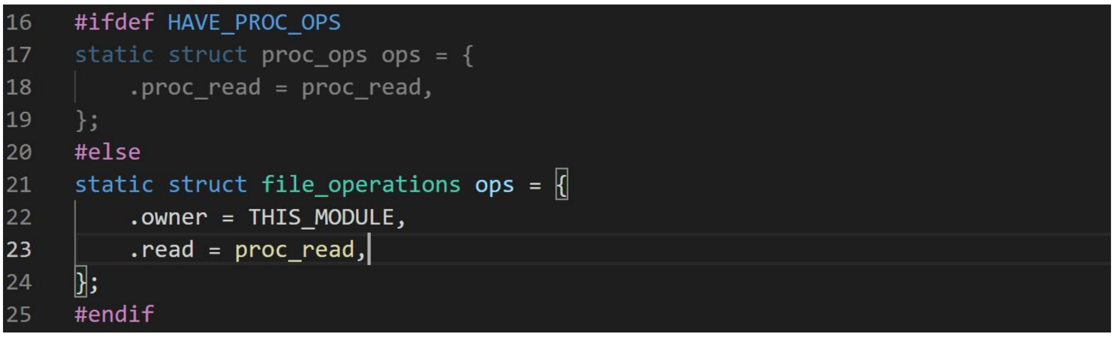
```
module_init(proc_init);
module_exit(proc_exit);
```

If push the command *insmod* in the terminal, it will use the *module_init* to load into the kernel and call the *proc_init* function. If push the command *rmmod* in the terminal, it will use the *module_exit* to remove from the kernel and call the *proc_exit* function. Here we load module into the kernel and record the jiffies at that time which is the kernel module was first loaded before creating the process.



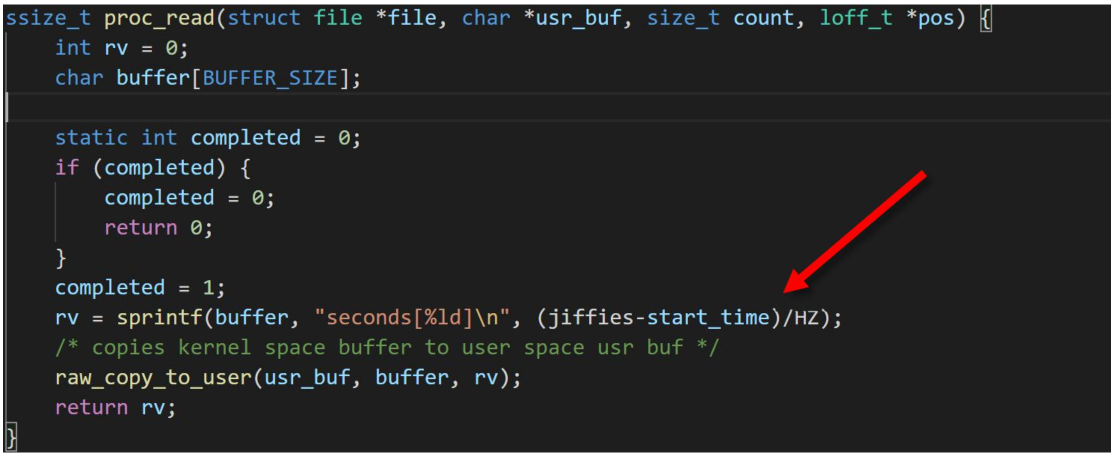
```
int proc_init(void){
    /* creates the /proc/jiffies entry */
    start_time = jiffies;
    proc_create(PROC_NAME, 0666, NULL, &ops);
    return 0;
}

/* Function is called when the module is removed. */
void proc_exit(void){
/* removes the /proc/jiffies entry */
    remove_proc_entry(PROC_NAME, NULL);
}
```

In the *proc_init* function, it will create a process "jiffies" in the /proc directory, and we find the ops address by *&ops* which is a structure. In this structure, we can assign which operation we want to do, here we call the *proc_read* method.



```
#ifdef HAVE_PROC_OPS
static struct proc_ops ops = {
    .proc_read = proc_read,
};
#else
static struct file_operations ops = {
    .owner = THIS_MODULE,
    .read = proc_read,
};
#endif
```

In the *proc_read* function, we should assign a buffer which is use to copy the message in kernel space buffer to user space *usr buf*. And the message is the number of seconds that have elapsed since the kernel module was first loaded. Jiffies is the number of times the timer is triggered, which means that once interrupted it will increment by one. HZ means that how many interrupts occur per second. So the seconds that have elapsed since the kernel module was first loaded equal to $\frac{jiffies_{now}-jiffies_{start}}{HZ}$. Remember that the $start\_time$ is assigned at the $proc\_init$ function.



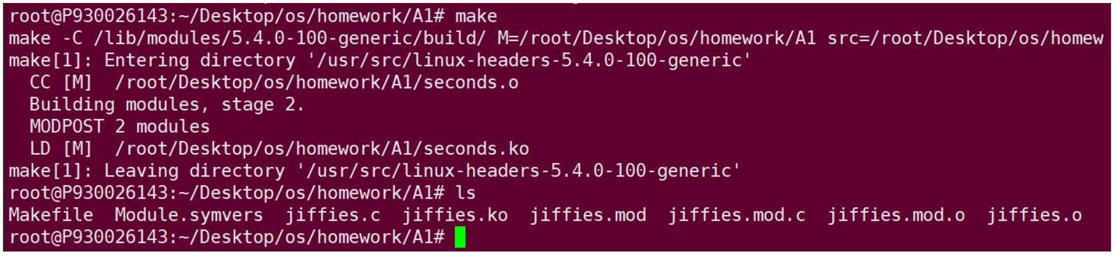
```
ssize_t proc_read(struct file *file, char *usr_buf, size_t count, loff_t *pos) {
    int rv = 0;
    char buffer[BUFFER_SIZE];

    static int completed = 0;
    if (completed) {
        completed = 0;
        return 0;
    }
    completed = 1;
    rv = sprintf(buffer, "seconds[%ld]\n", (jiffies-start_time)/HZ);
    /* copies kernel space buffer to user space usr buf */
    raw_copy_to_user(usr_buf, buffer, rv);
    return rv;
}
```

The $jiffies.c$ task is similar with $seconds.c$, the difference is the message in kernel space buffer to user space $usr\ buf$ which is just the current jiffies.

a. Show the result:

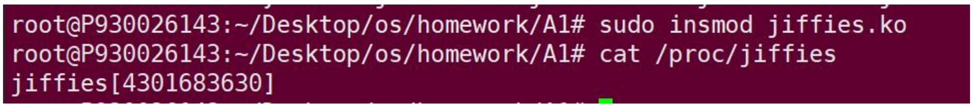
1. $jiffies.c$

make **& Show the output in the directory**



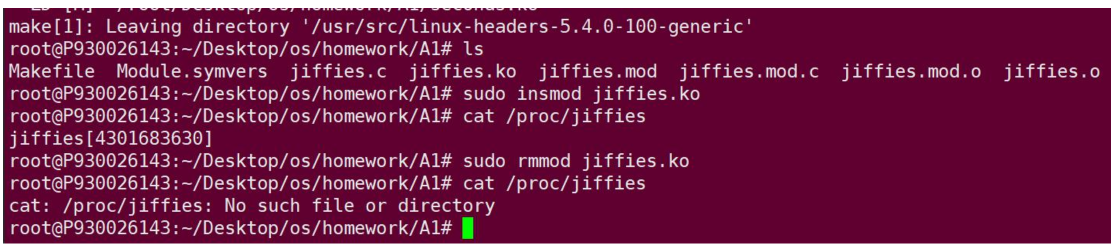
```
root@P930026143:~/Desktop/os/homework/A1# make
make -C /lib/modules/5.4.0-100-generic/build/ M=/root/Desktop/os/homework/A1 src=/root/Desktop/os/homew
make[1]: Entering directory '/usr/src/linux-headers-5.4.0-100-generic'
  CC [M]  /root/Desktop/os/homework/A1/seconds.o
  Building modules, stage 2.
  MODPOST 2 modules
  LD [M]  /root/Desktop/os/homework/A1/seconds.ko
make[1]: Leaving directory '/usr/src/linux-headers-5.4.0-100-generic'
root@P930026143:~/Desktop/os/homework/A1# ls
Makefile  Module.symvers  jiffies.c  jiffies.ko  jiffies.mod  jiffies.mod.c  jiffies.mod.o  jiffies.o
root@P930026143:~/Desktop/os/homework/A1# 
```

$\boldsymbol{sudo\ insmod\ jiffies.ko}$ & $\boldsymbol{cat\ /proc/jiffies}$



```
root@P930026143:~/Desktop/os/homework/A1# sudo insmod jiffies.ko
root@P930026143:~/Desktop/os/homework/A1# cat /proc/jiffies
jiffies[4301683630]
```

$\boldsymbol{sudo\ rmmod\ jiffies.c}$

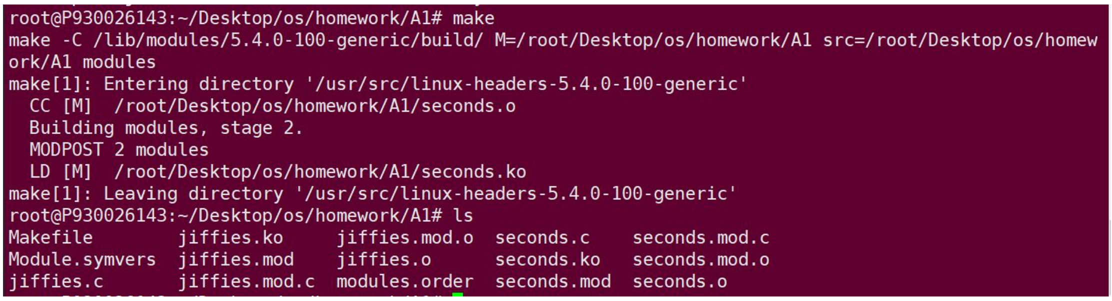


```
make[1]: Leaving directory '/usr/src/linux-headers-5.4.0-100-generic'
root@P930026143:~/Desktop/os/homework/A1# ls
Makefile  Module.symvers  jiffies.c  jiffies.ko  jiffies.mod  jiffies.mod.c  jiffies.mod.o  jiffies.o
root@P930026143:~/Desktop/os/homework/A1# sudo insmod jiffies.ko
root@P930026143:~/Desktop/os/homework/A1# cat /proc/jiffies
jiffies[4301683630]
root@P930026143:~/Desktop/os/homework/A1# sudo rmmod jiffies.ko
root@P930026143:~/Desktop/os/homework/A1# cat /proc/jiffies
cat: /proc/jiffies: No such file or directory
root@P930026143:~/Desktop/os/homework/A1# 
```

We can see that the $\boldsymbol{/proc/jiffies}$ be removed successfully.

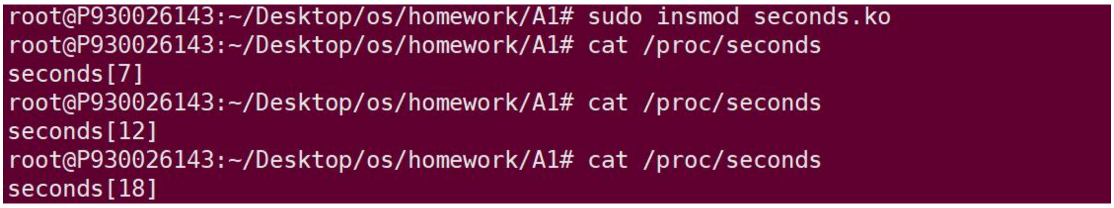
2. *seconds.c*

make & Show the output in the directory



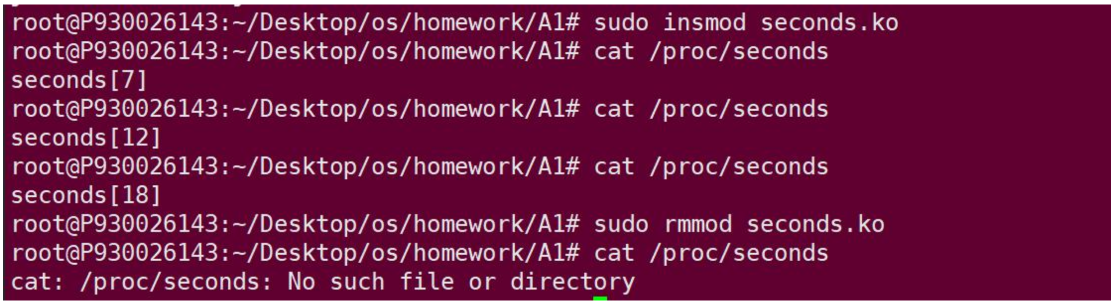
```
root@P930026143:~/Desktop/os/homework/A1# make
make -C /lib/modules/5.4.0-100-generic/build/ M=/root/Desktop/os/homework/A1 src=/root/Desktop/os/homew
ork/A1 modules
make[1]: Entering directory '/usr/src/linux-headers-5.4.0-100-generic'
  CC [M]  /root/Desktop/os/homework/A1/seconds.o
  Building modules, stage 2.
  MODPOST 2 modules
  LD [M]  /root/Desktop/os/homework/A1/seconds.ko
make[1]: Leaving directory '/usr/src/linux-headers-5.4.0-100-generic'
root@P930026143:~/Desktop/os/homework/A1# ls
Makefile        jiffies.ko     jiffies.mod.o  seconds.c    seconds.mod.c
Module.symvers  jiffies.mod    jiffies.o      seconds.ko   seconds.mod.o
jiffies.c       jiffies.mod.c  modules.order  seconds.mod  seconds.o
```

sudo insmod seconds.ko & *cat /proc/seconds*



```
root@P930026143:~/Desktop/os/homework/A1# sudo insmod seconds.ko
root@P930026143:~/Desktop/os/homework/A1# cat /proc/seconds
seconds[7]
root@P930026143:~/Desktop/os/homework/A1# cat /proc/seconds
seconds[12]
root@P930026143:~/Desktop/os/homework/A1# cat /proc/seconds
seconds[18]
```

sudo rmmod seconds.c

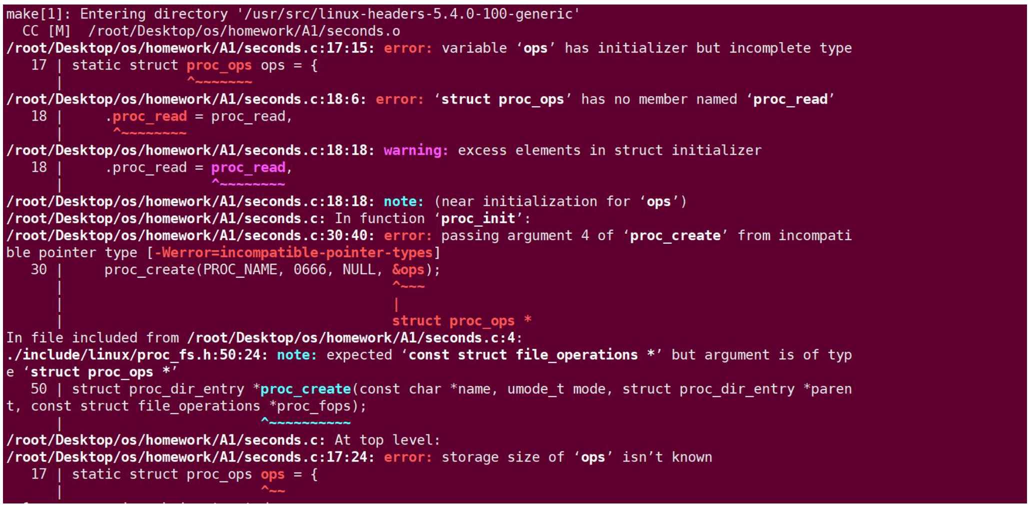


```
root@P930026143:~/Desktop/os/homework/A1# sudo insmod seconds.ko
root@P930026143:~/Desktop/os/homework/A1# cat /proc/seconds
seconds[7]
root@P930026143:~/Desktop/os/homework/A1# cat /proc/seconds
seconds[12]
root@P930026143:~/Desktop/os/homework/A1# cat /proc/seconds
seconds[18]
root@P930026143:~/Desktop/os/homework/A1# sudo rmmod seconds.ko
root@P930026143:~/Desktop/os/homework/A1# cat /proc/seconds
cat: /proc/seconds: No such file or directory
```

Checking his progress every few seconds, we can see that it is successfully timing and return the seconds. And we also can see that the *I/proc/seconds* be removed successfully.

b. problem encountered and solution

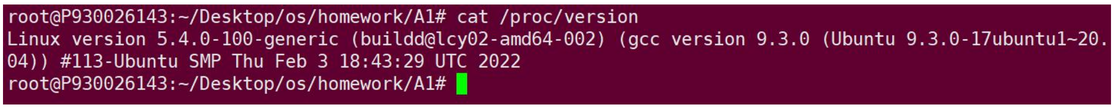
1. Unable to identify *ops* structure which is form the define **HAVE_PROC_OPS.**



```
make[1]: Entering directory '/usr/src/linux-headers-5.4.0-100-generic'
  CC [M]  /root/Desktop/os/homework/A1/seconds.o
/root/Desktop/os/homework/A1/seconds.c:17:15: error: variable ‘ops’ has initializer but incomplete type
   17 | static struct proc_ops ops = {
      |               ^~~~~~~~
/root/Desktop/os/homework/A1/seconds.c:18:6: error: ‘struct proc_ops’ has no member named ‘proc_read’
   18 |     .proc_read = proc_read,
      |      ^~~~~~~~~
/root/Desktop/os/homework/A1/seconds.c:18:18: warning: excess elements in struct initializer
   18 |     .proc_read = proc_read,
      |                  ^~~~~~~~~
/root/Desktop/os/homework/A1/seconds.c:18:18: note: (near initialization for ‘ops’)
/root/Desktop/os/homework/A1/seconds.c: In function ‘proc_init’:
/root/Desktop/os/homework/A1/seconds.c:30:40: error: passing argument 4 of ‘proc_create’ from incompati
ble pointer type [-Werror=incompatible-pointer-types]
   30 |     proc_create(PROC_NAME, 0666, NULL, &ops);
      |                                        ^~~~
      |                                        |
      |                                        struct proc_ops *
In file included from /root/Desktop/os/homework/A1/seconds.c:4:
./include/linux/proc_fs.h:50:24: note: expected ‘const struct file_operations *’ but argument is of typ
e ‘struct proc_ops *’
   50 | struct proc_dir_entry *proc_create(const char *name, umode_t mode, struct proc_dir_entry *paren
t, const struct file_operations *proc_fops);
      |                        ^~~~~~~~~~~
/root/Desktop/os/homework/A1/seconds.c: At top level:
/root/Desktop/os/homework/A1/seconds.c:17:24: error: storage size of ‘ops’ isn’t known
   17 | static struct proc_ops ops = {
      |                        ^~~
```

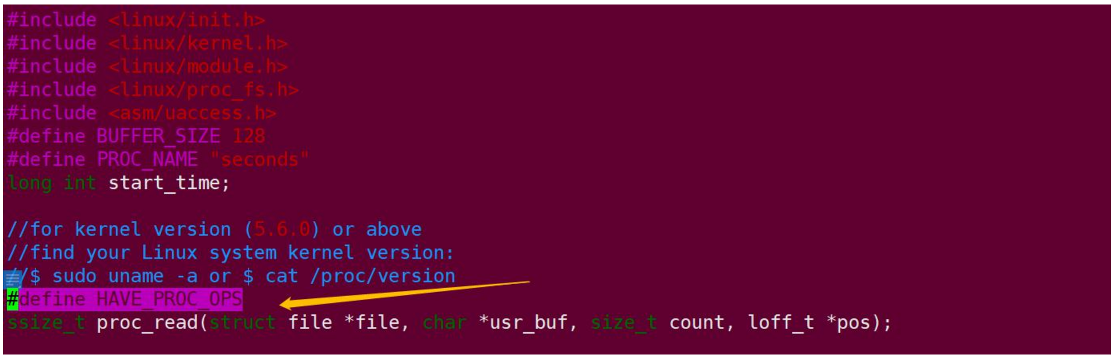
This is the process version problem. we can check the version by this command:

cat /proc/version



```
root@P930026143:~/Desktop/os/homework/A1# cat /proc/version
Linux version 5.4.0-100-generic (buildd@lcy02-amd64-002) (gcc version 9.3.0 (Ubuntu 9.3.0-17ubuntu1~20.
04)) #113-Ubuntu SMP Thu Feb 3 18:43:29 UTC 2022
root@P930026143:~/Desktop/os/homework/A1#
```

In this version, we do not need to add the **#define HAVE_PROC_OPS**. **As follow:**



```
#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"
long int start_time;

//for kernel version (5.6.0) or above
//find your Linux system kernel version:
//$ sudo uname -a or $ cat /proc/version
#define HAVE_PROC_OPS
ssize_t proc_read(struct file *file, char *usr_buf, size_t count, loff_t *pos);
```

Just remove and it will '*make*' successfully and generate the object file.