RedHat Linux Kernel Internals Laboratory Exercises

Lab 4: Our First Module

Objective: Revisit Lab 3 and build and load a hello world module into the RHEL kernel. The code for this module will be provided. Module licensing will be introduced and finally you will verify the module was loaded correctly by observing the debug system log for the running kernel.

1. Start Eclipse from Applications, Programming, Eclipse.

2. In the Project Explorer, right-click and select New, C-Project. In the dialogue window call the Project Name “Lab4” and selection Linux GCC for the toolchain:

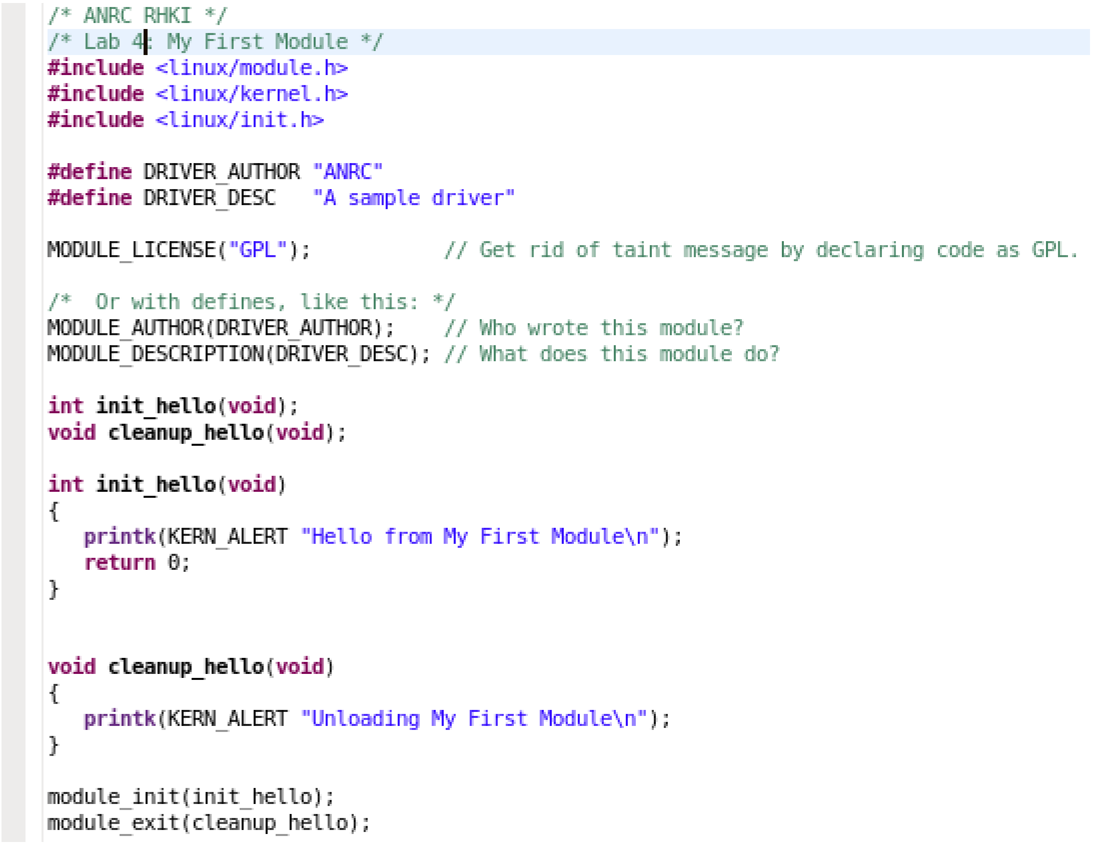
3. Click next and keep both “Debug” and “Release” configurations. Click “Finish”.

4. Right click on the “Lab4” project in Project Explorer, select New, File and call the file “lab4.c”. Create another file and call it “Makefile”.

5. The contents of Makefile will be the same as the one given in Lab 3 with the exception of the obj file name “lab4.o”.

6. In kernel 2.4 and later, a mechanism was devised to identify code licensed under the GPL so people can be warned that the code is non open-source. This is accomplished by the MODULE_LICENSE() macro which is demonstrated in this first module. By setting the license to GPL, you can keep the warning from being printed. This license mechanism is defined and documented in linux/module.h.

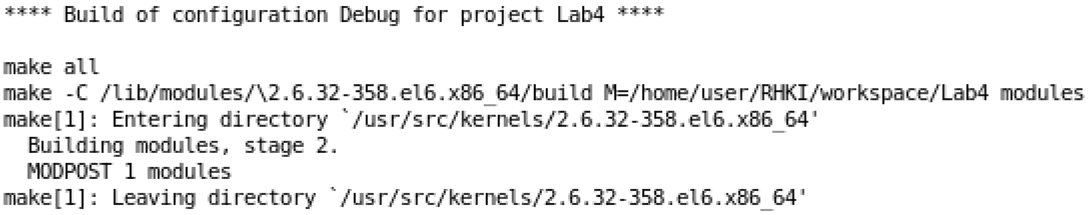
7. The following code will create our first module:



```
/* ANRC RHKI */
/* Lab 4: My First Module */
#include <linux/module.h>
#include <linux/kernel.h>
#include <linux/init.h>

#define DRIVER_AUTHOR "ANRC"
#define DRIVER_DESC   "A sample driver"

MODULE_LICENSE("GPL");           // Get rid of taint message by declaring code as GPL.

/*  Or with defines, like this: */
MODULE_AUTHOR(DRIVER_AUTHOR);    // Who wrote this module?
MODULE_DESCRIPTION(DRIVER_DESC); // What does this module do?

int init_hello(void);
void cleanup_hello(void);

int init_hello(void)
{
   printk(KERN_ALERT "Hello from My First Module\n");
   return 0;
}


void cleanup_hello(void)
{
   printk(KERN_ALERT "Unloading My First Module\n");
}

module_init(init_hello);
module_exit(cleanup_hello);
```

8. Create Make Targets for “all” and “clean”.

9. Finally run the “make all” target and observe the console window. You should have a clean compilation free of errors. If you do not, please let the instructor know as you probably have an issue with your development environment which needs to be fixed for future labs.



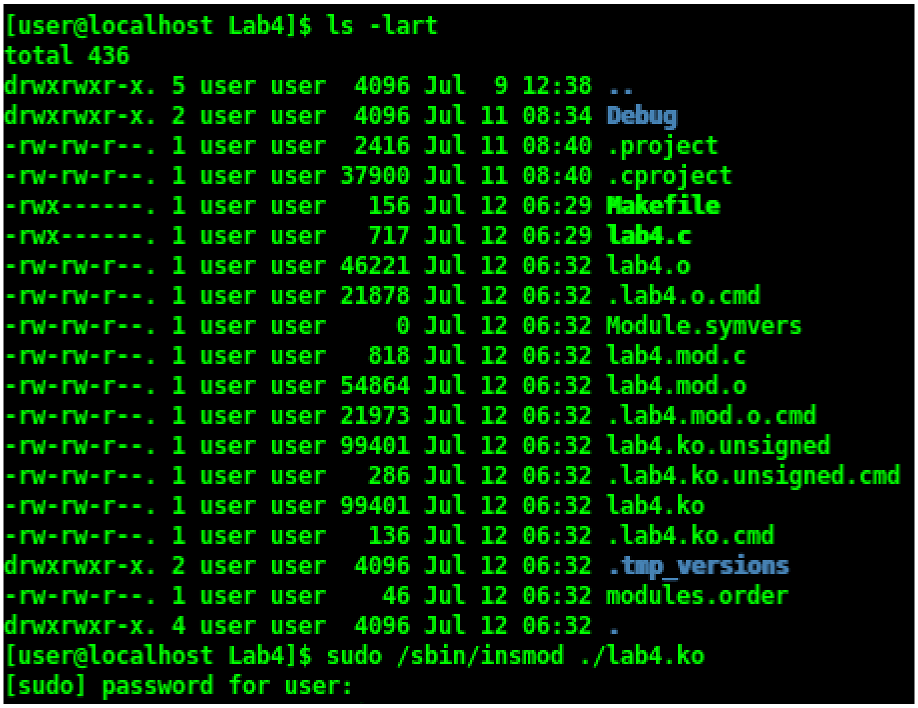
```
**** Build of configuration Debug for project Lab4 ****

make all
make -C /lib/modules/\2.6.32-358.el6.x86_64/build M=/home/user/RHKI/workspace/Lab4 modules
make[1]: Entering directory `/usr/src/kernels/2.6.32-358.el6.x86_64'
  Building modules, stage 2.
  MODPOST 1 modules
make[1]: Leaving directory `/usr/src/kernels/2.6.32-358.el6.x86_64'
```

10. To load your module start a command console (right-click on your desktop and Open Terminal).

11. Change directory to your workspace “Lab4” folder (see the console build output from step 9 for your specific directory).

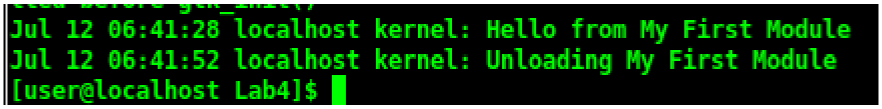
12. The kernel module is called “lab4.ko”. To load it you will need to run the command “sudo /sbin/insmod ./lab4.ko” and enter the root password “pa22w0rd”.

13. To observe our debug print statement you can dump the messages log with the command “sudo tail /var/log/messages”.



```
[user@localhost Lab4]$ ls -lart
total 436
drwxrwxr-x. 5 user user  4096 Jul  9 12:38 ..
drwxrwxr-x. 2 user user  4096 Jul 11 08:34 Debug
-rw-rw-r--. 1 user user  2416 Jul 11 08:40 .project
-rw-rw-r--. 1 user user 37900 Jul 11 08:40 .cproject
-rwx------. 1 user user   156 Jul 12 06:29 Makefile
-rwx------. 1 user user   717 Jul 12 06:29 lab4.c
-rw-rw-r--. 1 user user 46221 Jul 12 06:32 lab4.o
-rw-rw-r--. 1 user user 21878 Jul 12 06:32 .lab4.o.cmd
-rw-rw-r--. 1 user user     0 Jul 12 06:32 Module.symvers
-rw-rw-r--. 1 user user   818 Jul 12 06:32 lab4.mod.c
-rw-rw-r--. 1 user user 54864 Jul 12 06:32 lab4.mod.o
-rw-rw-r--. 1 user user 21973 Jul 12 06:32 .lab4.mod.o.cmd
-rw-rw-r--. 1 user user 99401 Jul 12 06:32 lab4.ko.unsigned
-rw-rw-r--. 1 user user   286 Jul 12 06:32 .lab4.ko.unsigned.cmd
-rw-rw-r--. 1 user user 99401 Jul 12 06:32 lab4.ko
-rw-rw-r--. 1 user user   136 Jul 12 06:32 .lab4.ko.cmd
drwxrwxr-x. 2 user user  4096 Jul 12 06:32 .tmp_versions
-rw-rw-r--. 1 user user    46 Jul 12 06:32 modules.order
drwxrwxr-x. 4 user user  4096 Jul 12 06:32 .
[user@localhost Lab4]$ sudo /sbin/insmod ./lab4.ko
[sudo] password for user:
```

14. Finally to unload the module and observe the exit printing out to the debug log issue the command “sudo /sbin/rmmod lab4” and dump the log file as before.



```
Jul 12 06:41:28 localhost kernel: Hello from My First Module
Jul 12 06:41:52 localhost kernel: Unloading My First Module
[user@localhost Lab4]$
```