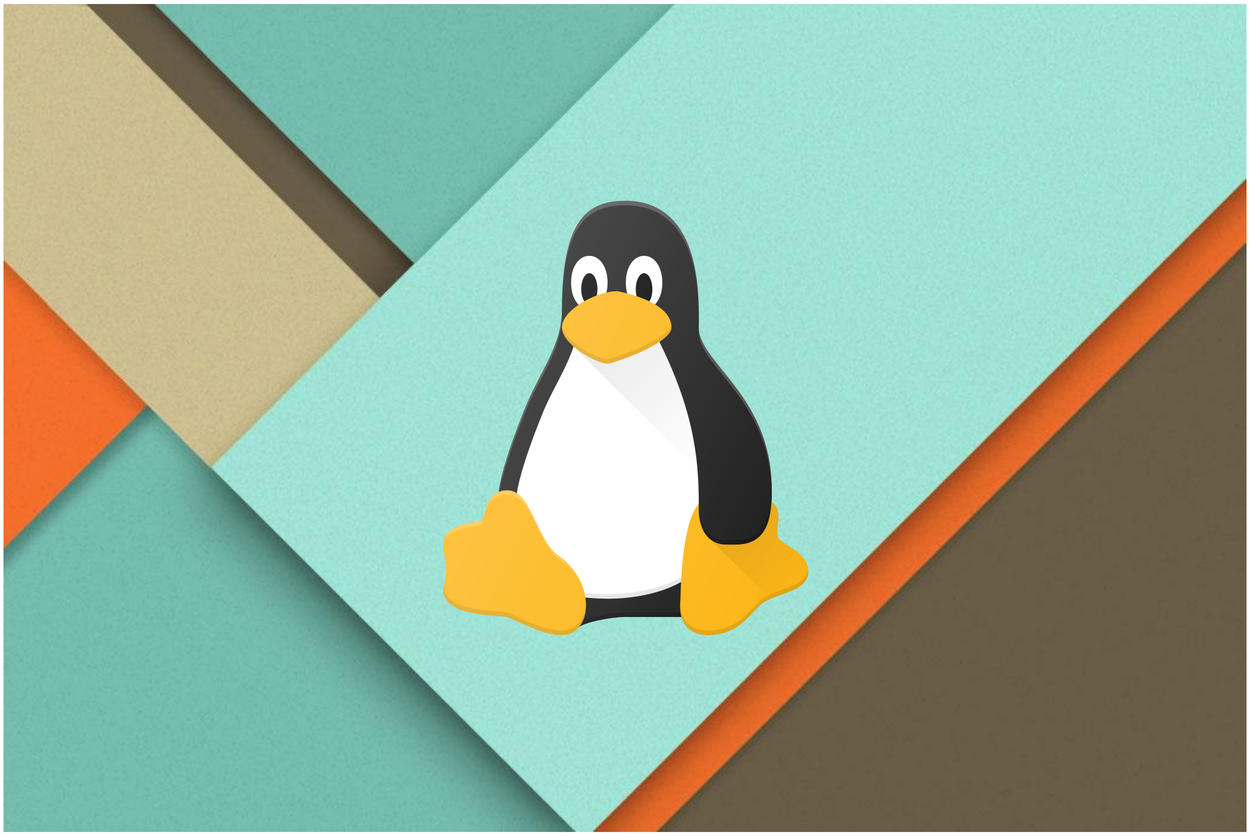


Operating Systems Lab

Experiment No. 3

01.02.2022

Professor - Dr. Shrinivas Khedkar.

Pratham Loya
201080068
IT
prloya_b20@it.vjti.ac.in

Linux Kernel Modules

In this project, you will learn how to create a kernel module and load it into the Linux kernel. The project can be completed using the Linux virtual machine. Although you may use an editor to write these C programs, you will have to use the terminal application to compile the programs, and you will have to enter commands on the command line to manage the modules in the kernel. As you'll discover, the advantage of developing kernel modules is that it is a relatively easy method of interacting with the kernel, thus allowing you to write programs that directly invoke kernel functions. It is important for you to keep in mind that you are indeed writing kernel code that directly interacts with the kernel. That normally means that any errors in the code could crash the system! However, since you will be using a virtual machine, any failures will at worst only require rebooting the system.

Assignment

Proceed through the steps described below to create the kernel module and to load and unload the module. Be sure to check the contents of the kernel log buffer using dmesg to ensure you have properly followed the steps.

Creating Kernel Modules

The first part of this project involves following a series of steps for creating and inserting a module into the Linux kernel.

You can list all kernel modules that are currently loaded by entering the command:

```
lsmod
```

This command will list the current kernel modules in three columns: name, size, and where the module is being used.

The following program (named simple.c and available with the source code for this text) illustrates a very basic kernel module that prints appropriate messages when the kernel module is loaded and unloaded

Program

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

/* This function is called when the module is loaded. */
int simple_init(void)
{
printk(KERN_INFO "Simple module is being Loaded \n");
return 0;
}

/* This function is called when the module is removed. */
void simple_exit(void)
{
printk(KERN_INFO "Removing Simple module \n");
}

/* 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");
```

Theory

1. The function **simple init()** is the *module entry point*, which represents the function that is invoked when the module is loaded into the kernel.

2. Similarly, the **simple exit()** function is the *module exit point*—the function that is called when the module is removed from the kernel.

3. The module entry point function must return an integer value, with *0 representing success and any other value representing failure.* The module exit point function returns void. Neither the module entry point nor the module exit point is passed any parameters.

4. The two following macros are used for registering the module entry and exit points with the kernel:

   module init()

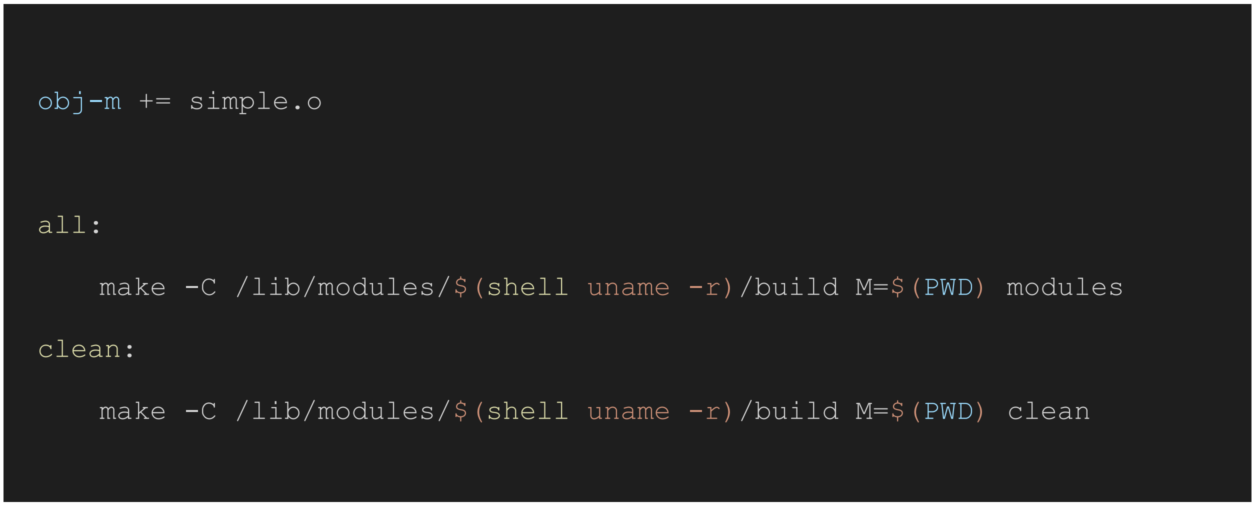
   module exit()

5. Notice how both the module entry and exit point functions make calls to the **printk()** function. *printk() is the kernel equivalent of printf()*, yet its output is sent to a kernel log buffer whose contents can be read by the **dmesg** command.

   One difference between printf() and printk() is that printk() allows us to specify a priority flag whose values are given in the include file. In this instance, the priority is KERN_INFO, which is defined as an informational message.

6. The final lines—MODULE LICENSE(), MODULE DESCRIPTION(), and MODULE AUTHOR()— represent details regarding the software license, description of the module, and author. For our purposes, we do not depend on this information, but we include it because it is standard practice in developing kernel modules.The two following macros are used for registering the module entry and exit points with the kernel.

7. This kernel module simple.c is compiled using the Makefile accompanying the source code with this project.



```
obj-m += simple.o


all:
    make -C /lib/modules/$(shell uname -r)/build M=$(PWD) modules
clean:
    make -C /lib/modules/$(shell uname -r)/build M=$(PWD) clean
```

8. To compile the module, enter the following on the command line:

   make

   The compilation produces several files. The file **simple.ko** represents the compiled kernel module.

   The following step illustrates inserting this module into the Linux kernel.

Loading and Removing Kernel Modules

Kernel modules are loaded using the **insmod** command, which is run as follows:

```
sudo insmod simple.ko
```

To check whether the module has loaded, enter the lsmod command and search for the module simple. Recall that the module entry point is invoked when the module is inserted into the kernel.

To check the contents of this message in the kernel log buffer, enter the command:

```
dmesg
```

You should see the message "Loading Module."

Removing the kernel module involves invoking the **rmmod** command

(notice that the .ko suffix is unnecessary):

```
sudo rmmod simple
```

Be sure to check with the **dmesg** command to ensure the module has been removed.

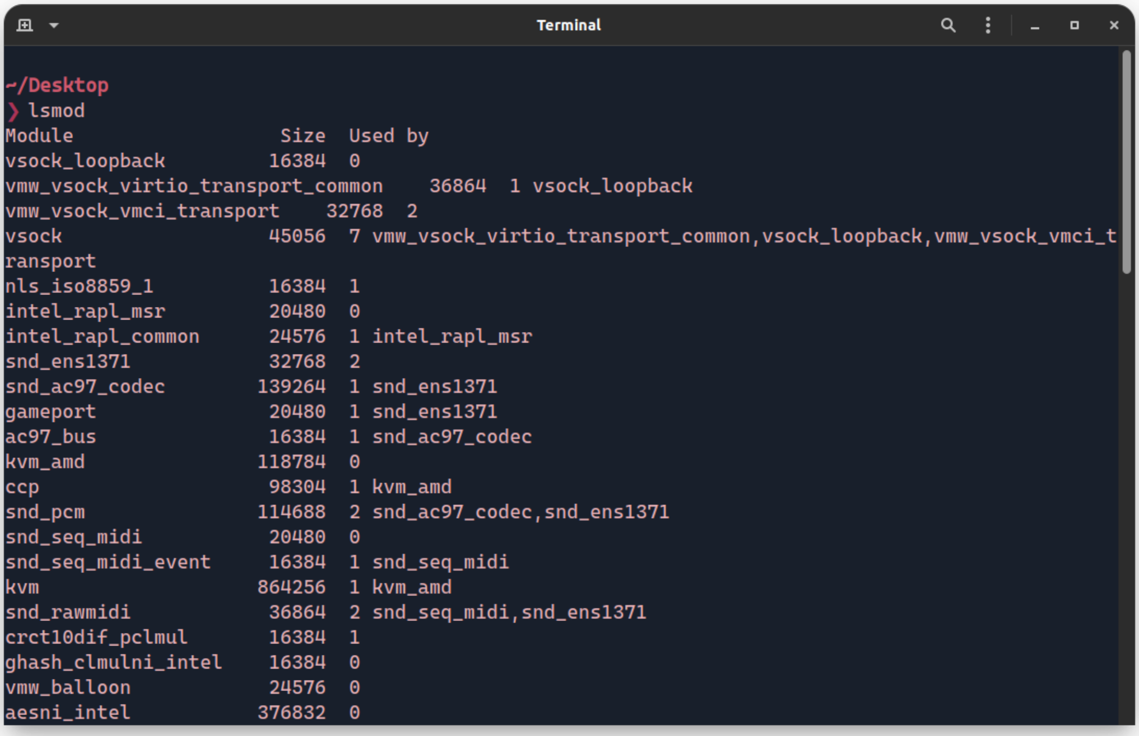
Because the kernel log buffer can fill up quickly, it often makes sense to clear the buffer periodically.

This can be accomplished as follows:

```
sudo dmesg –c
```

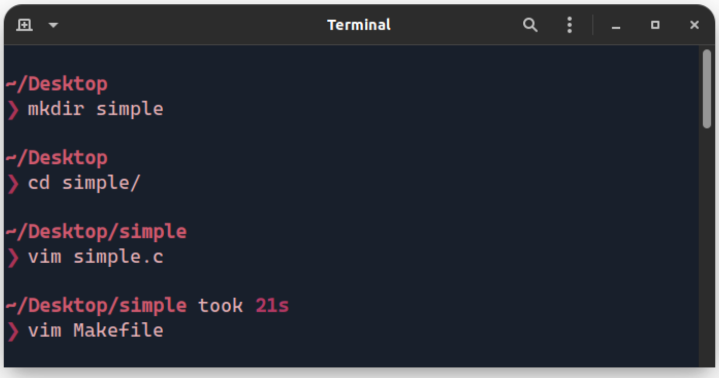
Screenshots

lsmod

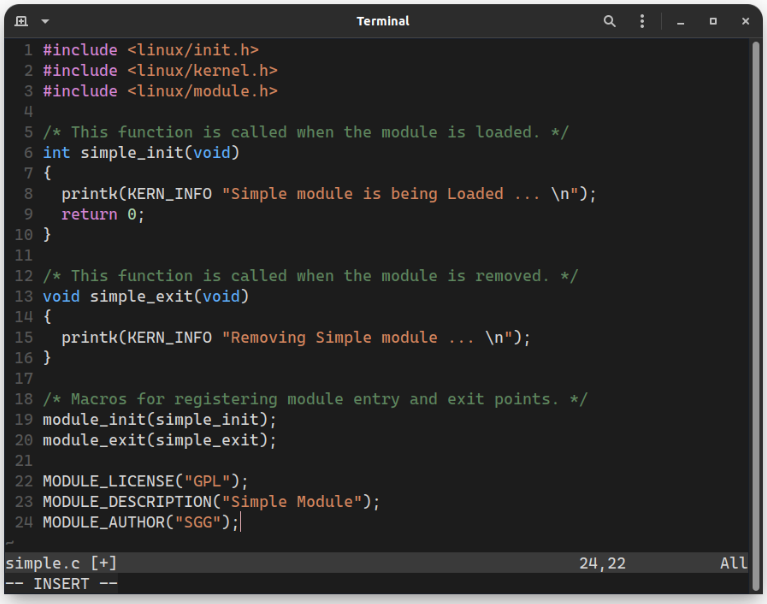


```
~/Desktop
❯ lsmod
Module                  Size  Used by
vsock_loopback         16384  0
vmw_vsock_virtio_transport_common    36864  1 vsock_loopback
vmw_vsock_vmci_transport    32768  2
vsock                  45056  7 vmw_vsock_virtio_transport_common,vsock_loopback,vmw_vsock_vmci_t
ransport
nls_iso8859_1          16384  1
intel_rapl_msr         20480  0
intel_rapl_common      24576  1 intel_rapl_msr
snd_ens1371            32768  2
snd_ac97_codec        139264  1 snd_ens1371
gameport               20480  1 snd_ens1371
ac97_bus               16384  1 snd_ac97_codec
kvm_amd               118784  0
ccp                    98304  1 kvm_amd
snd_pcm               114688  2 snd_ac97_codec,snd_ens1371
snd_seq_midi           20480  0
snd_seq_midi_event     16384  1 snd_seq_midi
kvm                   864256  1 kvm_amd
snd_rawmidi            36864  2 snd_seq_midi,snd_ens1371
crct10dif_pclmul       16384  1
ghash_clmulni_intel    16384  0
vmw_balloon            24576  0
aesni_intel           376832  0
```

creating .c file and Makefile

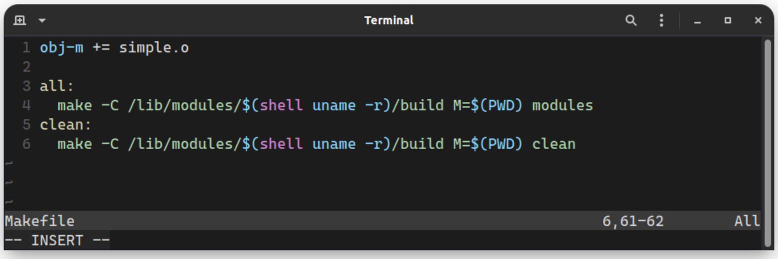


```
~/Desktop
❯ mkdir simple

~/Desktop
❯ cd simple/

~/Desktop/simple
❯ vim simple.c

~/Desktop/simple took 21s
❯ vim Makefile
```



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

/* This function is called when the module is loaded. */
int simple_init(void)
{
  printk(KERN_INFO "Simple module is being Loaded ... \n");
  return 0;
}

/* This function is called when the module is removed. */
void simple_exit(void)
{
  printk(KERN_INFO "Removing Simple module ... \n");
}

/* 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");
```

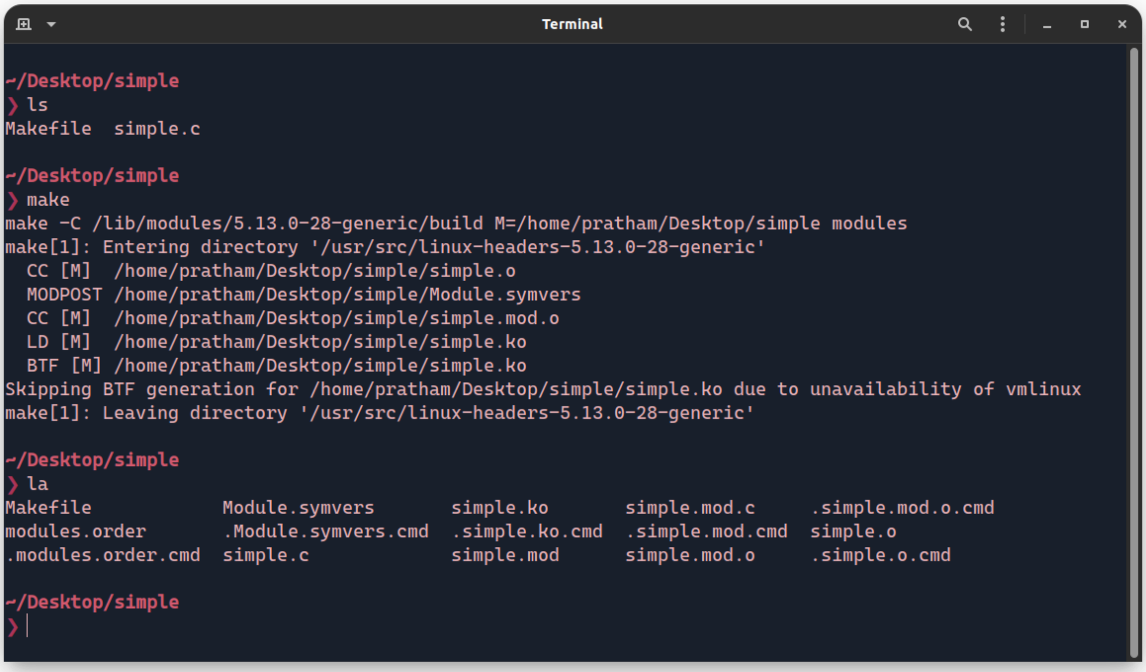
```
simple.c [+]                                              24,22          All
-- INSERT --
```



```makefile
obj-m += simple.o

all:
	make -C /lib/modules/$(shell uname -r)/build M=$(PWD) modules
clean:
	make -C /lib/modules/$(shell uname -r)/build M=$(PWD) clean
```

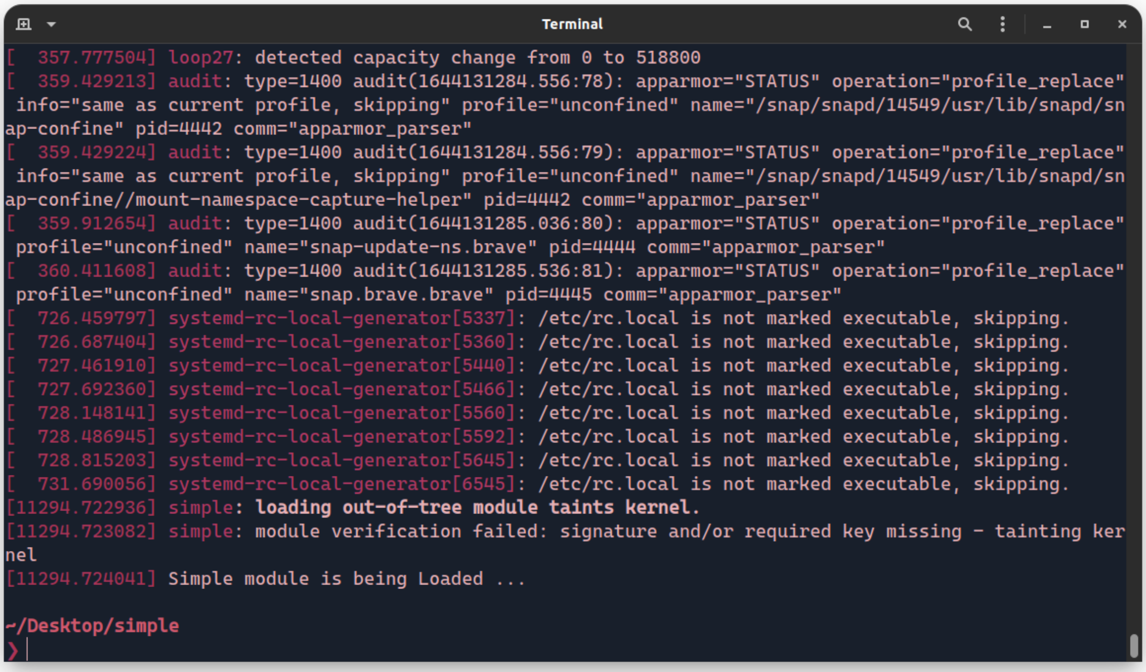
```
Makefile                                                     6,61-62       All
-- INSERT --
```

make

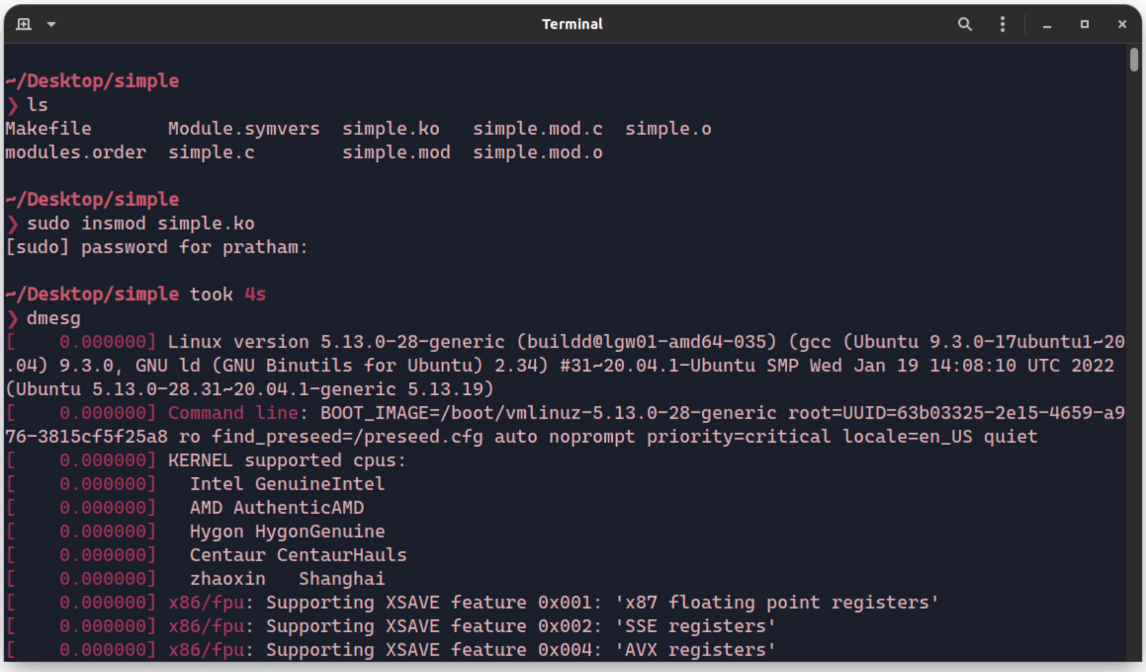


```
~/Desktop/simple
❯ ls
Makefile  simple.c

~/Desktop/simple
❯ make
make -C /lib/modules/5.13.0-28-generic/build M=/home/pratham/Desktop/simple modules
make[1]: Entering directory '/usr/src/linux-headers-5.13.0-28-generic'
  CC [M]  /home/pratham/Desktop/simple/simple.o
  MODPOST /home/pratham/Desktop/simple/Module.symvers
  CC [M]  /home/pratham/Desktop/simple/simple.mod.o
  LD [M]  /home/pratham/Desktop/simple/simple.ko
  BTF [M] /home/pratham/Desktop/simple/simple.ko
Skipping BTF generation for /home/pratham/Desktop/simple/simple.ko due to unavailability of vmlinux
make[1]: Leaving directory '/usr/src/linux-headers-5.13.0-28-generic'

~/Desktop/simple
❯ la
Makefile           Module.symvers       simple.ko       simple.mod.c     .simple.mod.o.cmd
modules.order      .Module.symvers.cmd  .simple.ko.cmd  .simple.mod.cmd  simple.o
.modules.order.cmd  simple.c            simple.mod      simple.mod.o     .simple.o.cmd

~/Desktop/simple
❯ 
```

Loading Module

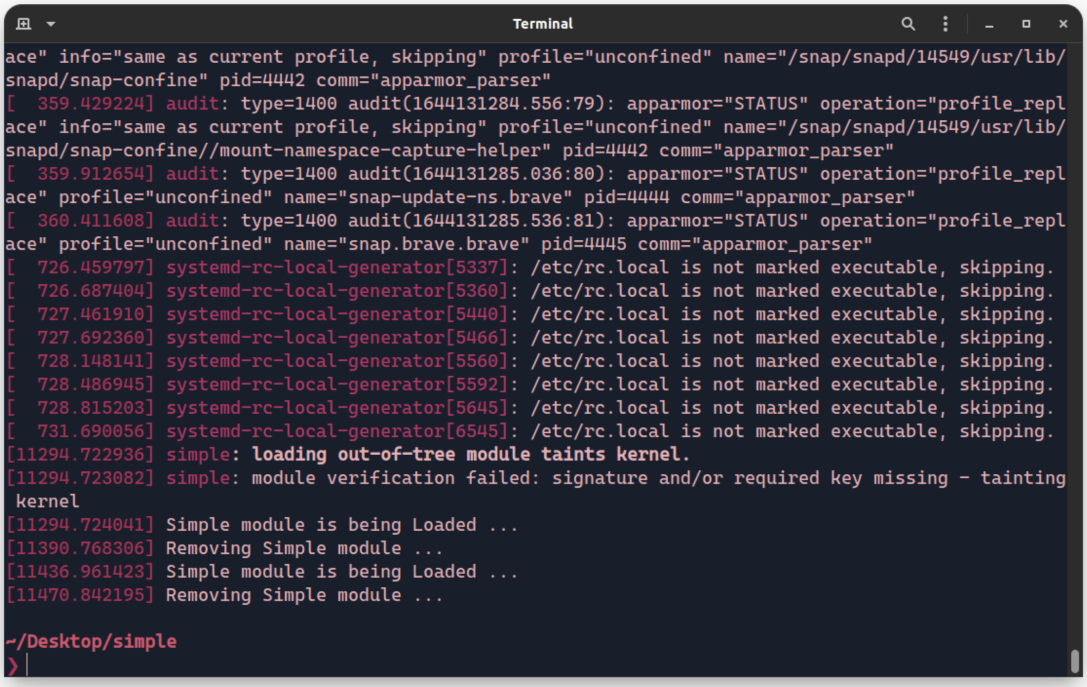


```
[  357.777504] loop27: detected capacity change from 0 to 518800
[  359.429213] audit: type=1400 audit(1644131284.556:78): apparmor="STATUS" operation="profile_replace"
 info="same as current profile, skipping" profile="unconfined" name="/snap/snapd/14549/usr/lib/snapd/sn
ap-confine" pid=4442 comm="apparmor_parser"
[  359.429224] audit: type=1400 audit(1644131284.556:79): apparmor="STATUS" operation="profile_replace"
 info="same as current profile, skipping" profile="unconfined" name="/snap/snapd/14549/usr/lib/snapd/sn
ap-confine//mount-namespace-capture-helper" pid=4442 comm="apparmor_parser"
[  359.912654] audit: type=1400 audit(1644131285.036:80): apparmor="STATUS" operation="profile_replace"
 profile="unconfined" name="snap-update-ns.brave" pid=4444 comm="apparmor_parser"
[  360.411608] audit: type=1400 audit(1644131285.536:81): apparmor="STATUS" operation="profile_replace"
 profile="unconfined" name="snap.brave.brave" pid=4445 comm="apparmor_parser"
[  726.459797] systemd-rc-local-generator[5337]: /etc/rc.local is not marked executable, skipping.
[  726.687404] systemd-rc-local-generator[5360]: /etc/rc.local is not marked executable, skipping.
[  727.461910] systemd-rc-local-generator[5440]: /etc/rc.local is not marked executable, skipping.
[  727.692360] systemd-rc-local-generator[5466]: /etc/rc.local is not marked executable, skipping.
[  728.148141] systemd-rc-local-generator[5560]: /etc/rc.local is not marked executable, skipping.
[  728.486945] systemd-rc-local-generator[5592]: /etc/rc.local is not marked executable, skipping.
[  728.815203] systemd-rc-local-generator[5645]: /etc/rc.local is not marked executable, skipping.
[  731.690056] systemd-rc-local-generator[6545]: /etc/rc.local is not marked executable, skipping.
[11294.722936] simple: loading out-of-tree module taints kernel.
[11294.723082] simple: module verification failed: signature and/or required key missing - tainting ker
nel
[11294.724041] Simple module is being Loaded ...

~/Desktop/simple
❯
```

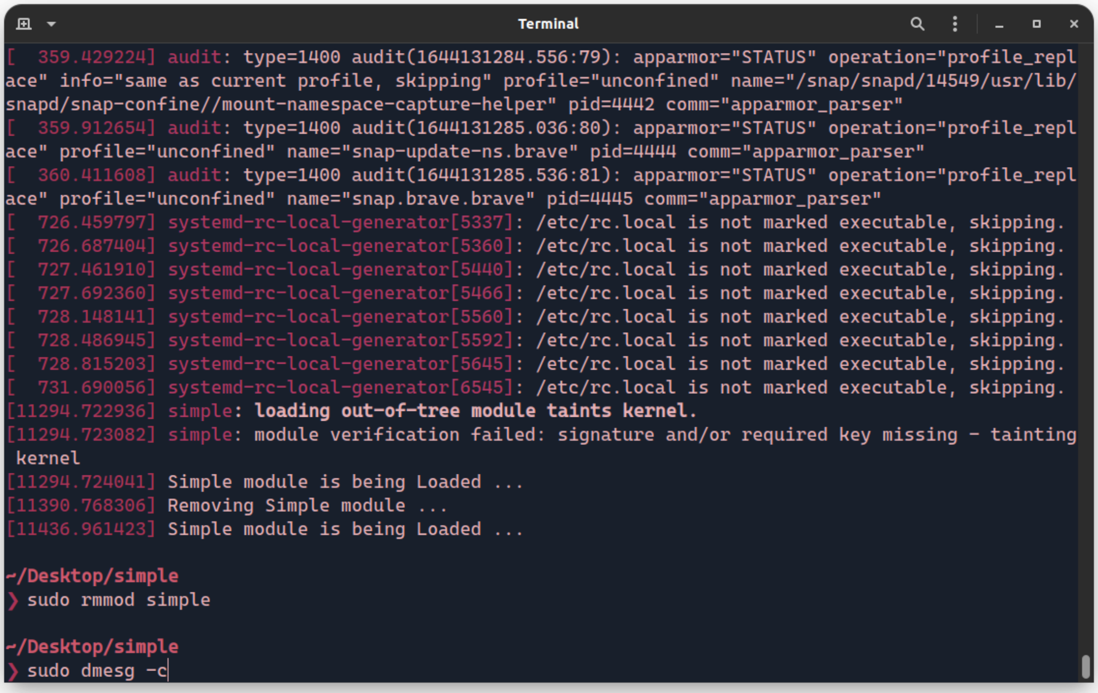


```
~/Desktop/simple
❯ ls
Makefile       Module.symvers  simple.ko   simple.mod.c  simple.o
modules.order  simple.c        simple.mod  simple.mod.o

~/Desktop/simple
❯ sudo insmod simple.ko
[sudo] password for pratham:

~/Desktop/simple took 4s
❯ dmesg
[    0.000000] Linux version 5.13.0-28-generic (buildd@lgw01-amd64-035) (gcc (Ubuntu 9.3.0-17ubuntu1~20
.04) 9.3.0, GNU ld (GNU Binutils for Ubuntu) 2.34) #31~20.04.1-Ubuntu SMP Wed Jan 19 14:08:10 UTC 2022
(Ubuntu 5.13.0-28.31~20.04.1-generic 5.13.19)
[    0.000000] Command line: BOOT_IMAGE=/boot/vmlinuz-5.13.0-28-generic root=UUID=63b03325-2e15-4659-a9
76-3815cf5f25a8 ro find_preseed=/preseed.cfg auto noprompt priority=critical locale=en_US quiet
[    0.000000] KERNEL supported cpus:
[    0.000000]   Intel GenuineIntel
[    0.000000]   AMD AuthenticAMD
[    0.000000]   Hygon HygonGenuine
[    0.000000]   Centaur CentaurHauls
[    0.000000]   zhaoxin   Shanghai
[    0.000000] x86/fpu: Supporting XSAVE feature 0x001: 'x87 floating point registers'
[    0.000000] x86/fpu: Supporting XSAVE feature 0x002: 'SSE registers'
[    0.000000] x86/fpu: Supporting XSAVE feature 0x004: 'AVX registers'
```

Removing Module

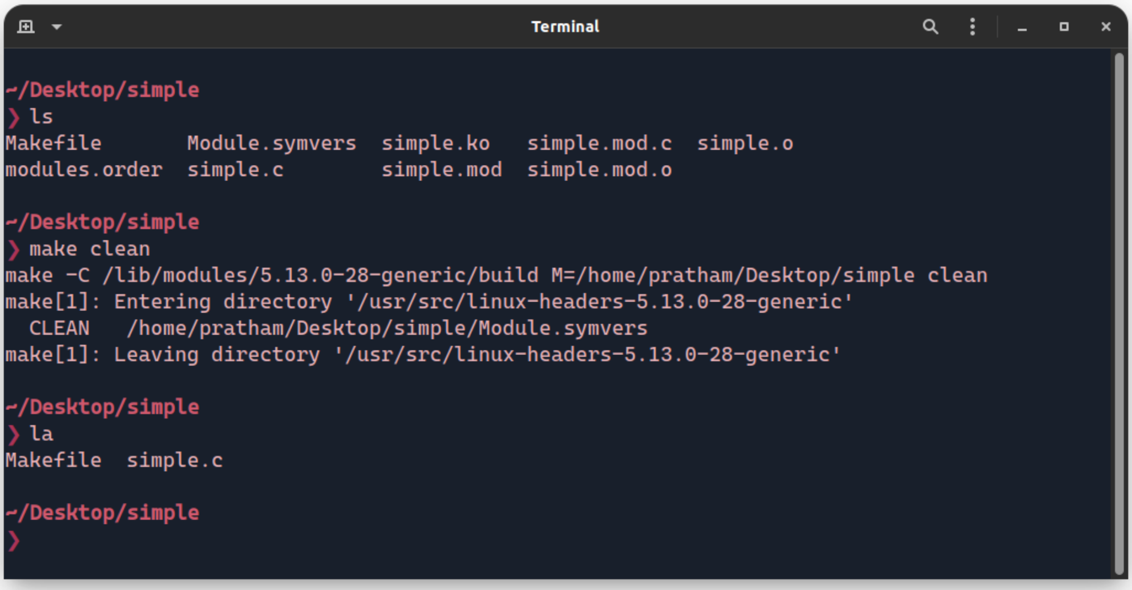


```
ace" info="same as current profile, skipping" profile="unconfined" name="/snap/snapd/14549/usr/lib/
snapd/snap-confine" pid=4442 comm="apparmor_parser"
[  359.429224] audit: type=1400 audit(1644131284.556:79): apparmor="STATUS" operation="profile_repl
ace" info="same as current profile, skipping" profile="unconfined" name="/snap/snapd/14549/usr/lib/
snapd/snap-confine//mount-namespace-capture-helper" pid=4442 comm="apparmor_parser"
[  359.912654] audit: type=1400 audit(1644131285.036:80): apparmor="STATUS" operation="profile_repl
ace" profile="unconfined" name="snap-update-ns.brave" pid=4444 comm="apparmor_parser"
[  360.411608] audit: type=1400 audit(1644131285.536:81): apparmor="STATUS" operation="profile_repl
ace" profile="unconfined" name="snap.brave.brave" pid=4445 comm="apparmor_parser"
[  726.459797] systemd-rc-local-generator[5337]: /etc/rc.local is not marked executable, skipping.
[  726.687404] systemd-rc-local-generator[5360]: /etc/rc.local is not marked executable, skipping.
[  727.461910] systemd-rc-local-generator[5440]: /etc/rc.local is not marked executable, skipping.
[  727.692360] systemd-rc-local-generator[5466]: /etc/rc.local is not marked executable, skipping.
[  728.148141] systemd-rc-local-generator[5560]: /etc/rc.local is not marked executable, skipping.
[  728.486945] systemd-rc-local-generator[5592]: /etc/rc.local is not marked executable, skipping.
[  728.815203] systemd-rc-local-generator[5645]: /etc/rc.local is not marked executable, skipping.
[  731.690056] systemd-rc-local-generator[6545]: /etc/rc.local is not marked executable, skipping.
[11294.722936] simple: loading out-of-tree module taints kernel.
[11294.723082] simple: module verification failed: signature and/or required key missing - tainting
 kernel
[11294.724041] Simple module is being Loaded ...
[11390.768306] Removing Simple module ...
[11436.961423] Simple module is being Loaded ...
[11470.842195] Removing Simple module ...

~/Desktop/simple
❯
```



```
[  359.429224] audit: type=1400 audit(1644131284.556:79): apparmor="STATUS" operation="profile_repl
ace" info="same as current profile, skipping" profile="unconfined" name="/snap/snapd/14549/usr/lib/
snapd/snap-confine//mount-namespace-capture-helper" pid=4442 comm="apparmor_parser"
[  359.912654] audit: type=1400 audit(1644131285.036:80): apparmor="STATUS" operation="profile_repl
ace" profile="unconfined" name="snap-update-ns.brave" pid=4444 comm="apparmor_parser"
[  360.411608] audit: type=1400 audit(1644131285.536:81): apparmor="STATUS" operation="profile_repl
ace" profile="unconfined" name="snap.brave.brave" pid=4445 comm="apparmor_parser"
[  726.459797] systemd-rc-local-generator[5337]: /etc/rc.local is not marked executable, skipping.
[  726.687404] systemd-rc-local-generator[5360]: /etc/rc.local is not marked executable, skipping.
[  727.461910] systemd-rc-local-generator[5440]: /etc/rc.local is not marked executable, skipping.
[  727.692360] systemd-rc-local-generator[5466]: /etc/rc.local is not marked executable, skipping.
[  728.148141] systemd-rc-local-generator[5560]: /etc/rc.local is not marked executable, skipping.
[  728.486945] systemd-rc-local-generator[5592]: /etc/rc.local is not marked executable, skipping.
[  728.815203] systemd-rc-local-generator[5645]: /etc/rc.local is not marked executable, skipping.
[  731.690056] systemd-rc-local-generator[6545]: /etc/rc.local is not marked executable, skipping.
[11294.722936] simple: loading out-of-tree module taints kernel.
[11294.723082] simple: module verification failed: signature and/or required key missing - tainting
 kernel
[11294.724041] Simple module is being Loaded ...
[11390.768306] Removing Simple module ...
[11436.961423] Simple module is being Loaded ...

~/Desktop/simple
❯ sudo rmmod simple

~/Desktop/simple
❯ sudo dmesg -c
```

clean



```
~/Desktop/simple
❯ ls
Makefile       Module.symvers  simple.ko   simple.mod.c  simple.o
modules.order  simple.c        simple.mod  simple.mod.o

~/Desktop/simple
❯ make clean
make -C /lib/modules/5.13.0-28-generic/build M=/home/pratham/Desktop/simple clean
make[1]: Entering directory '/usr/src/linux-headers-5.13.0-28-generic'
  CLEAN   /home/pratham/Desktop/simple/Module.symvers
make[1]: Leaving directory '/usr/src/linux-headers-5.13.0-28-generic'

~/Desktop/simple
❯ la
Makefile  simple.c

~/Desktop/simple
❯
```

Conclusion

In this experiment we have learned how to write, compile(i.e make), load, remove and clean kernel modules in linux.

Thank You