LAB 4
LINUX KERNEL DEVELOPMENT

Fullname: Trương Đặng Trúc Lâm

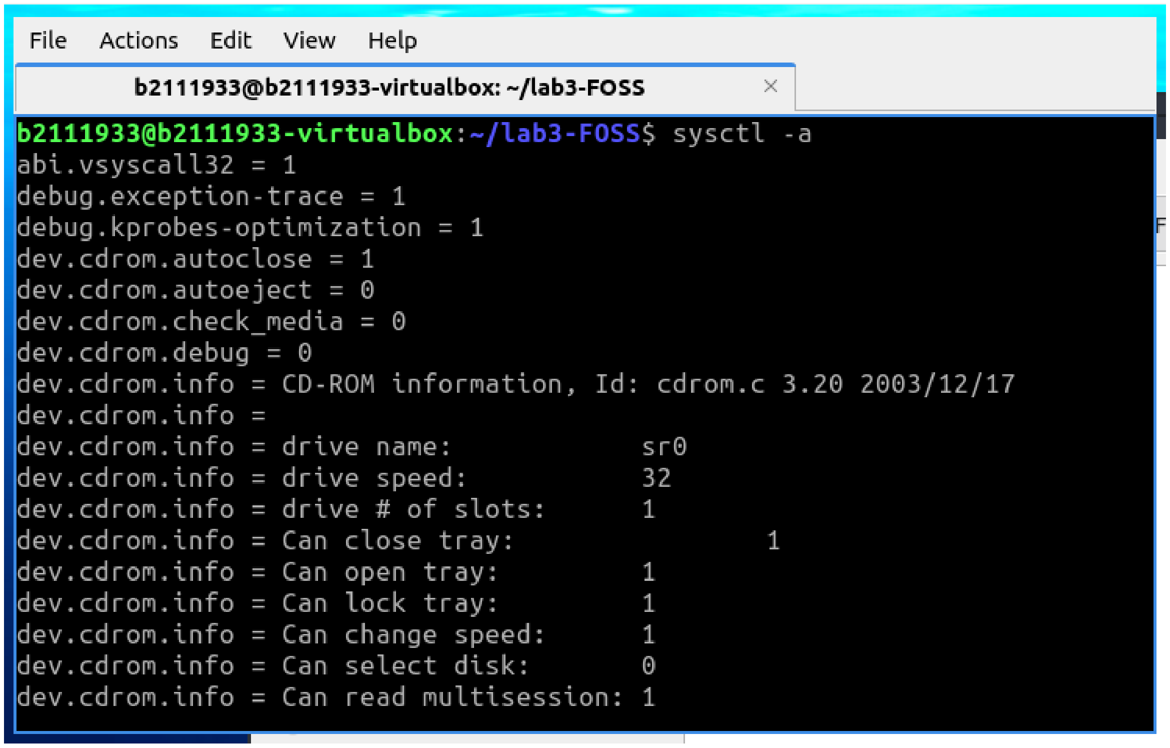
Student ID: B2111933

- Note: screenshots need to be clear and good-looking; submissions must be in PDF format.

1. Modify kernel parameters and install new modules

- List all linux kernel parameters on your OS:

```
sysctl -a
```

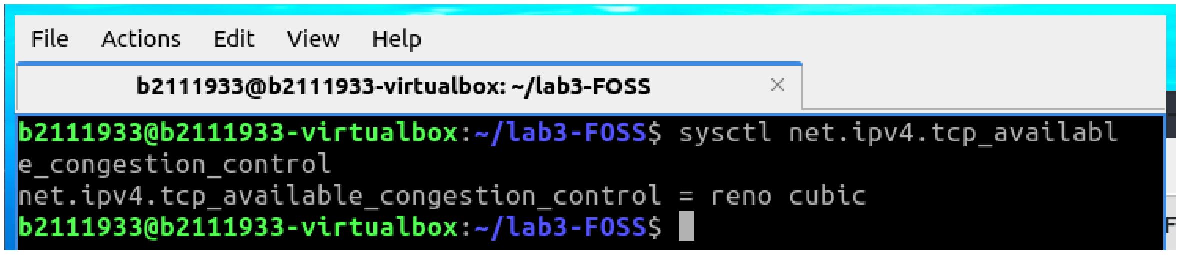


```
b2111933@b2111933-virtualbox:~/lab3-FOSS$ sysctl -a
abi.vsyscall32 = 1
debug.exception-trace = 1
debug.kprobes-optimization = 1
dev.cdrom.autoclose = 1
dev.cdrom.autoeject = 0
dev.cdrom.check_media = 0
dev.cdrom.debug = 0
dev.cdrom.info = CD-ROM information, Id: cdrom.c 3.20 2003/12/17
dev.cdrom.info =
dev.cdrom.info = drive name:            sr0
dev.cdrom.info = drive speed:           32
dev.cdrom.info = drive # of slots:      1
dev.cdrom.info = Can close tray:                1
dev.cdrom.info = Can open tray:         1
dev.cdrom.info = Can lock tray:         1
dev.cdrom.info = Can change speed:      1
dev.cdrom.info = Can select disk:       0
dev.cdrom.info = Can read multisession: 1
```

List all linux kernel parameters

- List all available TCP congestion control algorithms:

```
sysctl net.ipv4.tcp_available_congestion_control
```

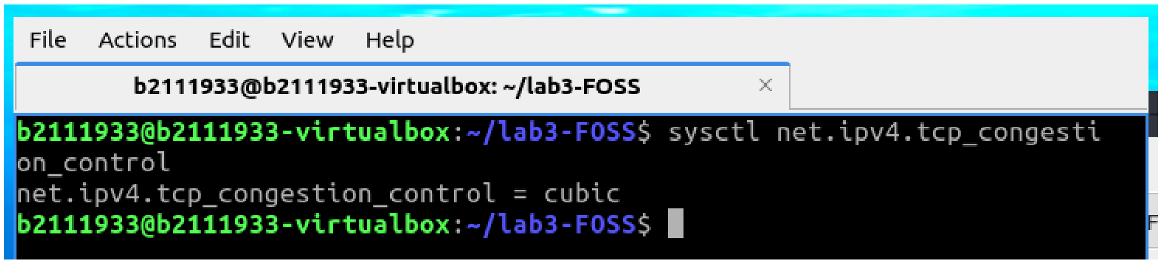

File Actions Edit View Help
b2111933@b2111933-virtualbox: ~/lab3-FOSS

```
b2111933@b2111933-virtualbox:~/lab3-FOSS$ sysctl net.ipv4.tcp_availabl
e_congestion_control
net.ipv4.tcp_available_congestion_control = reno cubic
b2111933@b2111933-virtualbox:~/lab3-FOSS$
```

List all available TCP congestion control algorithms

- Show which TCP congestion control algorithm is using:

```
sysctl net.ipv4.tcp_congestion_control
```

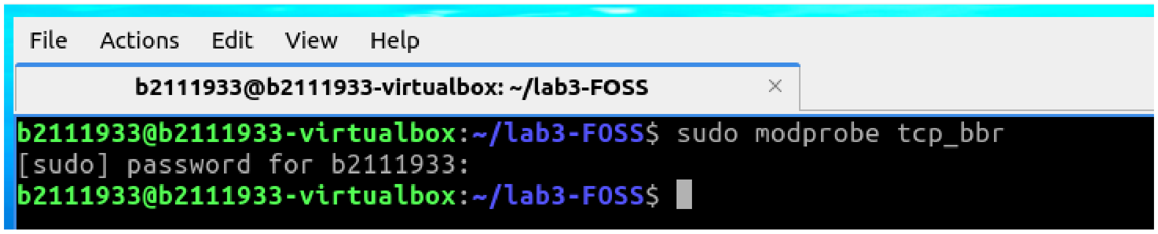

File Actions Edit View Help
b2111933@b2111933-virtualbox: ~/lab3-FOSS

```
b2111933@b2111933-virtualbox:~/lab3-FOSS$ sysctl net.ipv4.tcp_congesti
on_control
net.ipv4.tcp_congestion_control = cubic
b2111933@b2111933-virtualbox:~/lab3-FOSS$
```

The using TCP congestion control algorithm is **cubic**

- Install `bbr` TCP congestion control algorithm module:

```
sudo modprobe tcp_bbr
```

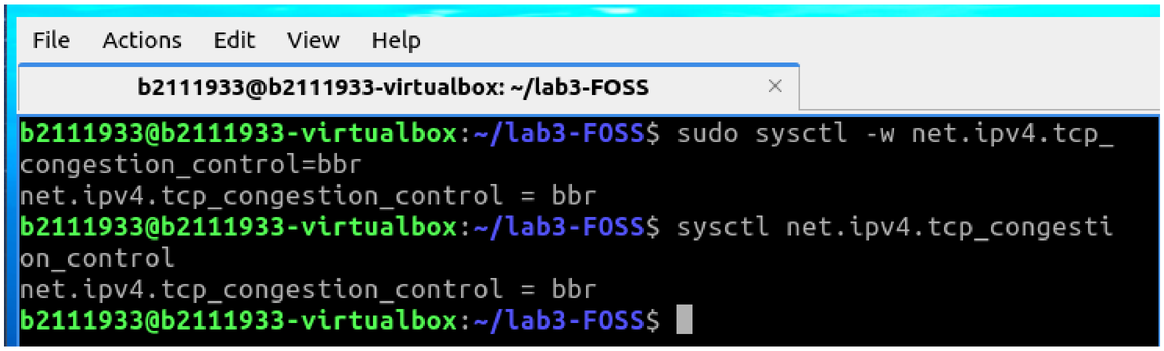

File Actions Edit View Help
b2111933@b2111933-virtualbox: ~/lab3-FOSS

```
b2111933@b2111933-virtualbox:~/lab3-FOSS$ sudo modprobe tcp_bbr
[sudo] password for b2111933:
b2111933@b2111933-virtualbox:~/lab3-FOSS$
```

Install **bbr** TCP congestion control algorithm module

- Switch to the `bbr` TCP congestion control algorithm:

```
sudo sysctl -w net.ipv4.tcp_congestion_control=bbr
sysctl net.ipv4.tcp_congestion_control
```


File Actions Edit View Help
b2111933@b2111933-virtualbox: ~/lab3-FOSS

```
b2111933@b2111933-virtualbox:~/lab3-FOSS$ sudo sysctl -w net.ipv4.tcp_
congestion_control=bbr
net.ipv4.tcp_congestion_control = bbr
b2111933@b2111933-virtualbox:~/lab3-FOSS$ sysctl net.ipv4.tcp_congesti
on_control
net.ipv4.tcp_congestion_control = bbr
b2111933@b2111933-virtualbox:~/lab3-FOSS$
```

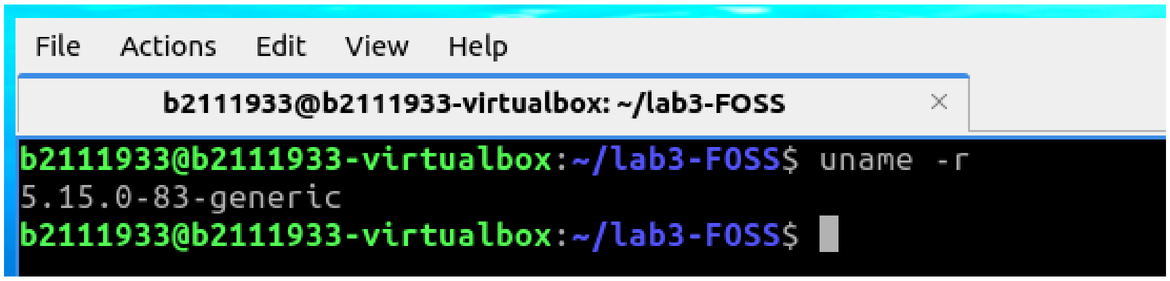
Switch to the **bbr** TCP congestion control algorithm & check the result

(take screenshots to show that you finish this exercise)

2. Install new kernel version

- Show your current kernel version:

```
uname -r
```

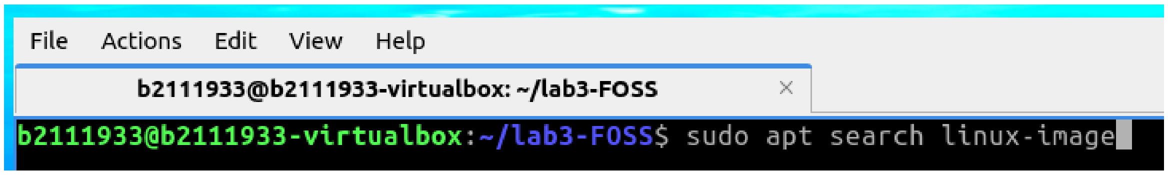

File Actions Edit View Help
b2111933@b2111933-virtualbox: ~/lab3-FOSS

```
b2111933@b2111933-virtualbox:~/lab3-FOSS$ uname -r
5.15.0-83-generic
b2111933@b2111933-virtualbox:~/lab3-FOSS$
```

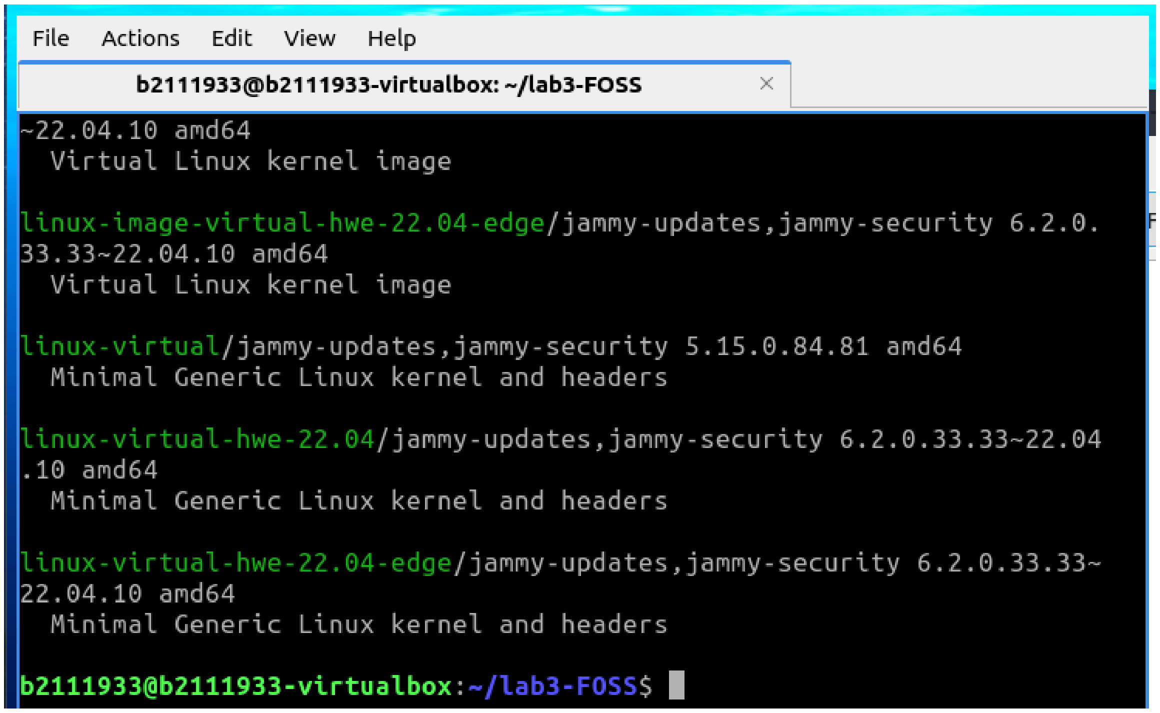
The current kernel version is **5.15.0-83-generic**

- Search for newer versions:

```
sudo apt search linux-image
```


File Actions Edit View Help
b2111933@b2111933-virtualbox: ~/lab3-FOSS

```
b2111933@b2111933-virtualbox:~/lab3-FOSS$ sudo apt search linux-image
```

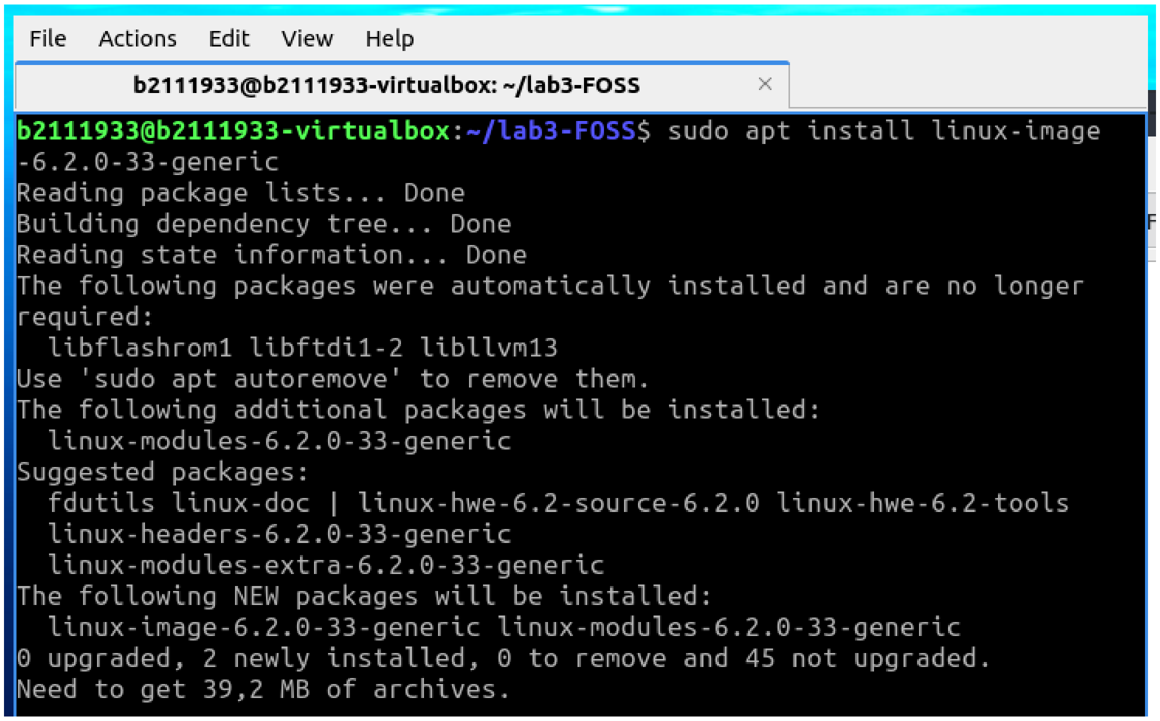

File Actions Edit View Help
b2111933@b2111933-virtualbox: ~/lab3-FOSS

```
~22.04.10 amd64
  Virtual Linux kernel image

linux-image-virtual-hwe-22.04-edge/jammy-updates,jammy-security 6.2.0.
33.33~22.04.10 amd64
  Virtual Linux kernel image

linux-virtual/jammy-updates,jammy-security 5.15.0.84.81 amd64
  Minimal Generic Linux kernel and headers

linux-virtual-hwe-22.04/jammy-updates,jammy-security 6.2.0.33.33~22.04
.10 amd64
  Minimal Generic Linux kernel and headers

linux-virtual-hwe-22.04-edge/jammy-updates,jammy-security 6.2.0.33.33~
22.04.10 amd64
  Minimal Generic Linux kernel and headers

b2111933@b2111933-virtualbox:~/lab3-FOSS$
```

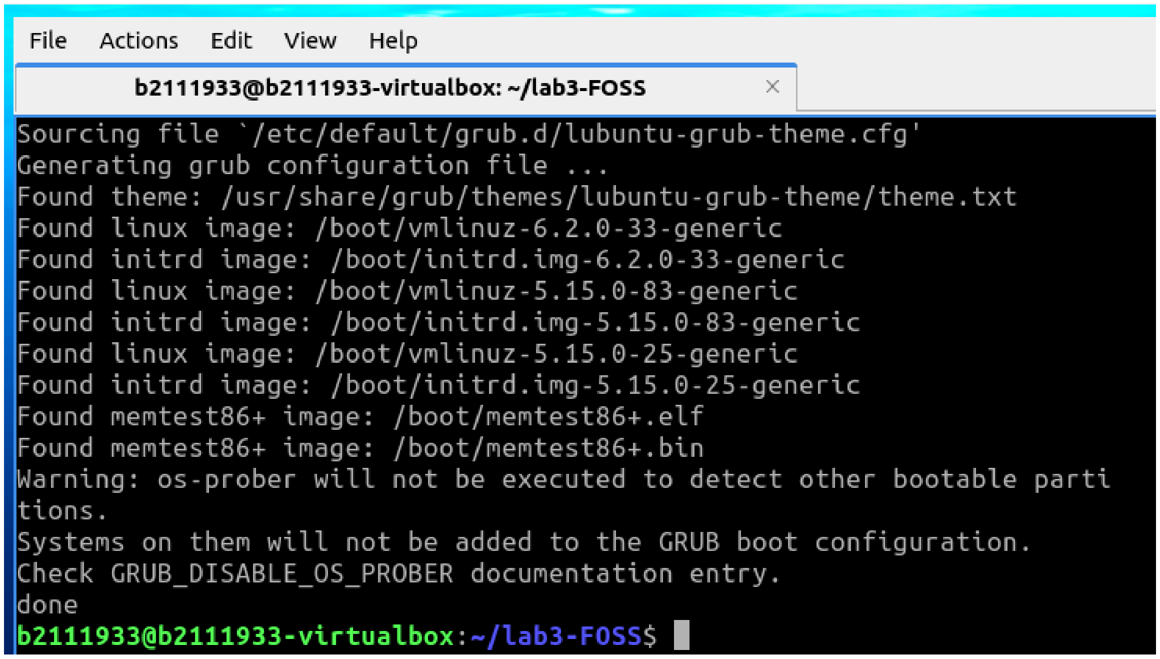
There are various newer versions of kernel

- Install the latest version you find:

```
sudo apt install linux-image-x.x.x-x-generic
```


File Actions Edit View Help
b2111933@b2111933-virtualbox: ~/lab3-FOSS

```
b2111933@b2111933-virtualbox:~/lab3-FOSS$ sudo apt install linux-image
-6.2.0-33-generic
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer
required:
  libflashrom1 libftdi1-2 libllvm13
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  linux-modules-6.2.0-33-generic
Suggested packages:
  fdutils linux-doc | linux-hwe-6.2-source-6.2.0 linux-hwe-6.2-tools
  linux-headers-6.2.0-33-generic
  linux-modules-extra-6.2.0-33-generic
The following NEW packages will be installed:
  linux-image-6.2.0-33-generic linux-modules-6.2.0-33-generic
0 upgraded, 2 newly installed, 0 to remove and 45 not upgraded.
Need to get 39,2 MB of archives.
```

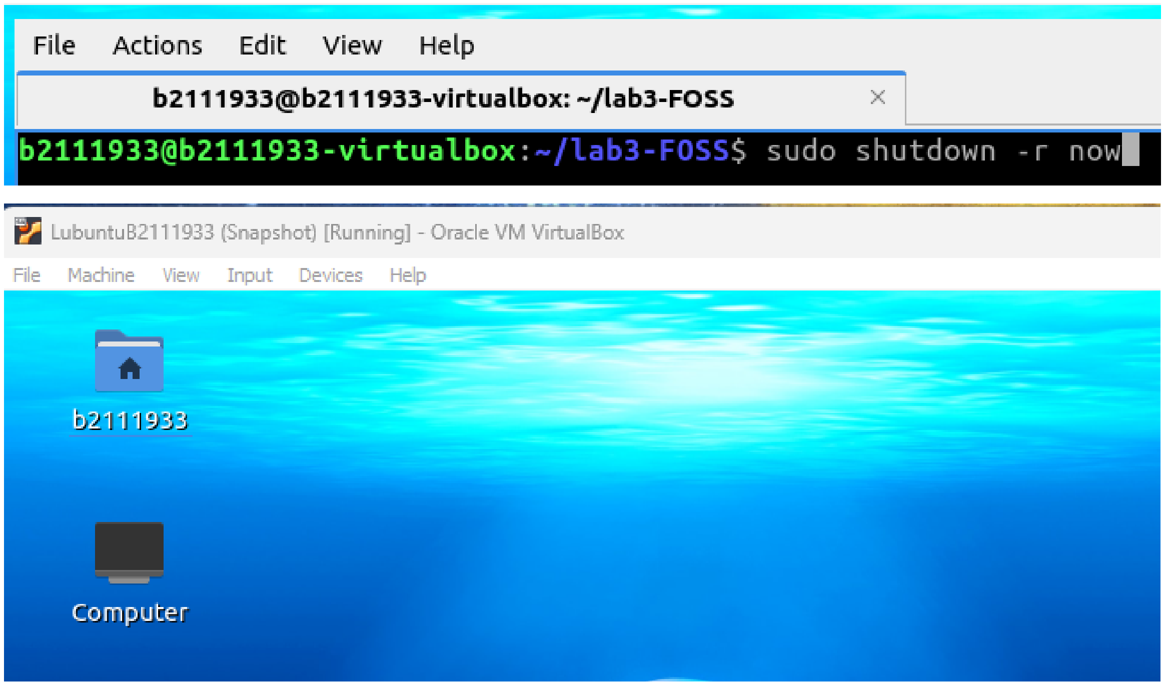

File Actions Edit View Help
b2111933@b2111933-virtualbox: ~/lab3-FOSS

```
Sourcing file `/etc/default/grub.d/lubuntu-grub-theme.cfg'
Generating grub configuration file ...
Found theme: /usr/share/grub/themes/lubuntu-grub-theme/theme.txt
Found linux image: /boot/vmlinuz-6.2.0-33-generic
Found initrd image: /boot/initrd.img-6.2.0-33-generic
Found linux image: /boot/vmlinuz-5.15.0-83-generic
Found initrd image: /boot/initrd.img-5.15.0-83-generic
Found linux image: /boot/vmlinuz-5.15.0-25-generic
Found initrd image: /boot/initrd.img-5.15.0-25-generic
Found memtest86+ image: /boot/memtest86+.elf
Found memtest86+ image: /boot/memtest86+.bin
Warning: os-prober will not be executed to detect other bootable parti
tions.
Systems on them will not be added to the GRUB boot configuration.
Check GRUB_DISABLE_OS_PROBER documentation entry.
done
b2111933@b2111933-virtualbox:~/lab3-FOSS$
```

Install version **6.2.0-33-generic**

- After a kernel upgrade, you must reboot the system. Then, if the device driver you need is in the latest kernel, your hardware will work as expected:

```
    sudo shutdown -r now
```

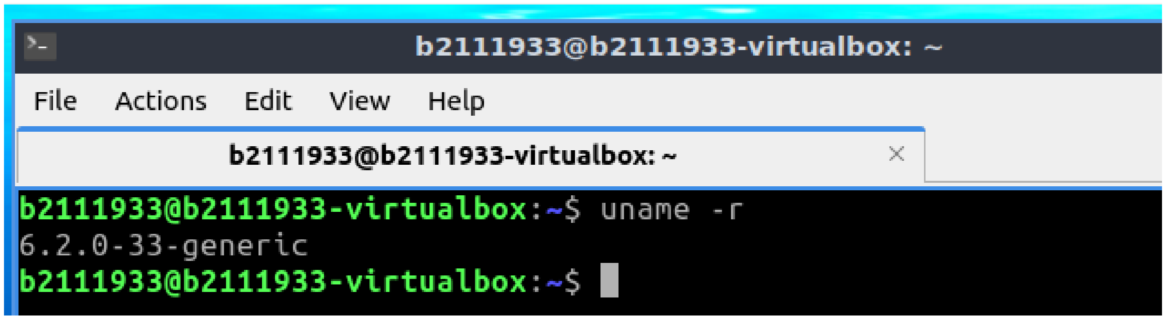

File Actions Edit View Help
b2111933@b2111933-virtualbox: ~/lab3-FOSS
LubuntuB2111933 (Snapshot) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
b2111933
Computer

```
b2111933@b2111933-virtualbox:~/lab3-FOSS$ sudo shutdown -r now
```

Shutdown & restart the VM

- Show your new current kernel version:

```
uname -r
```


b2111933@b2111933-virtualbox: ~
File Actions Edit View Help
b2111933@b2111933-virtualbox: ~

```
b2111933@b2111933-virtualbox:~$ uname -r
6.2.0-33-generic
b2111933@b2111933-virtualbox:~$
```

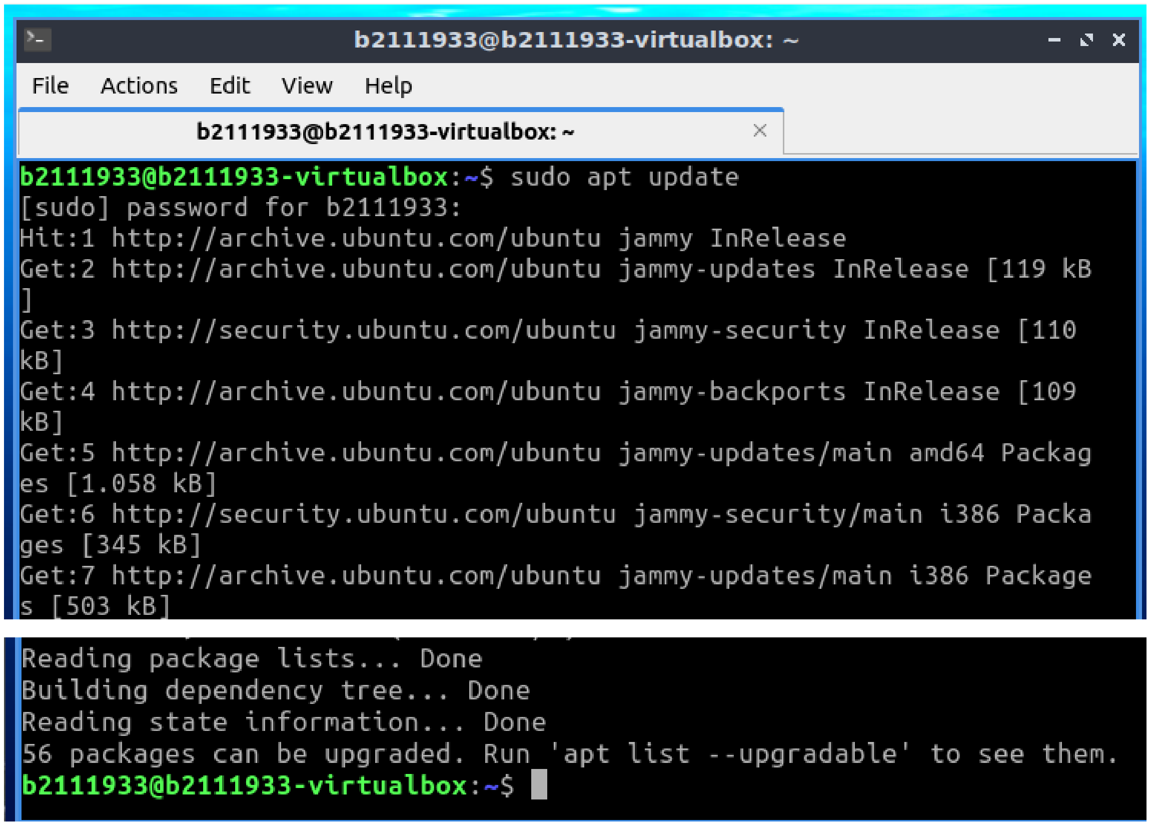
The current kernel version is **6.2.0-33-generic**

(take screenshots to show that you finish this exercise)

3. Build and install a new kernel version

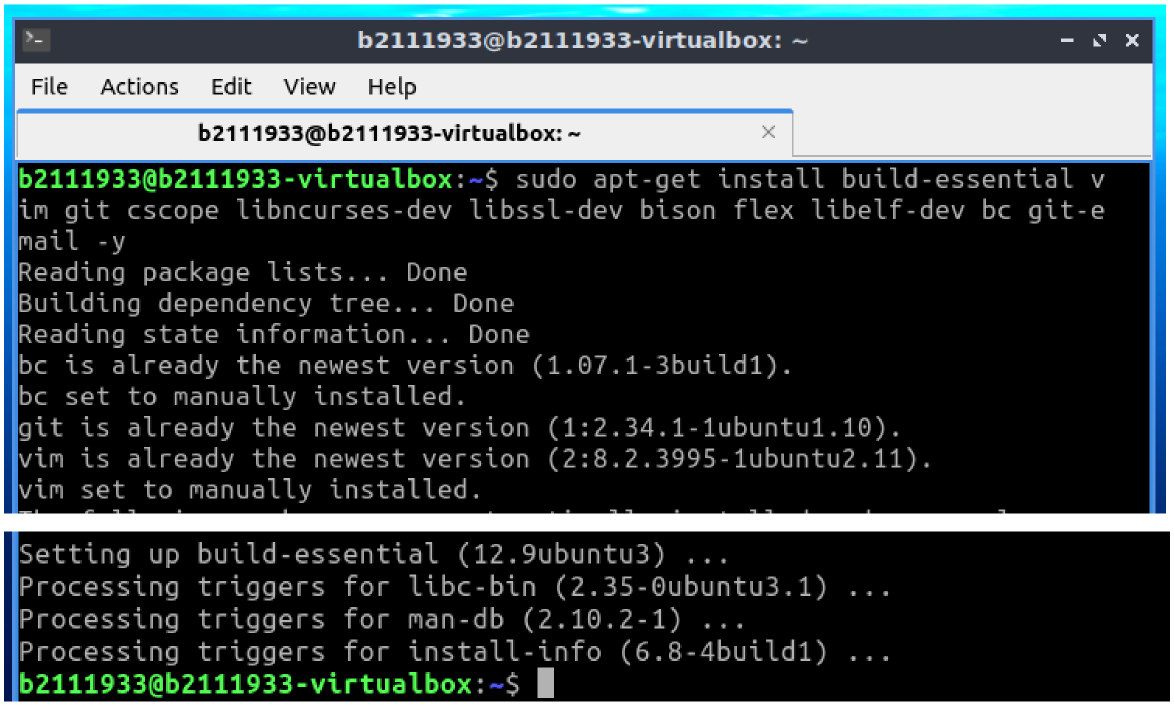
- Get your system ready

```
sudo apt update
```


b2111933@b2111933-virtualbox: ~
File Actions Edit View Help
b2111933@b2111933-virtualbox: ~

```
b2111933@b2111933-virtualbox:~$ sudo apt update
[sudo] password for b2111933:
Hit:1 http://archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB
]
Get:3 http://security.ubuntu.com/ubuntu jammy-security InRelease [110
kB]
Get:4 http://archive.ubuntu.com/ubuntu jammy-backports InRelease [109
kB]
Get:5 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packag
es [1.058 kB]
Get:6 http://security.ubuntu.com/ubuntu jammy-security/main i386 Packa
ges [345 kB]
Get:7 http://archive.ubuntu.com/ubuntu jammy-updates/main i386 Package
s [503 kB]
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
56 packages can be upgraded. Run 'apt list --upgradable' to see them.
b2111933@b2111933-virtualbox:~$
```

```
    sudo apt-get install build-essential vim git cscope
libncurses-dev libssl-dev bison flex libelf-dev bc
git-email -y
```


b2111933@b2111933-virtualbox: ~
File
Actions
Edit
View
Help
b2111933@b2111933-virtualbox: ~

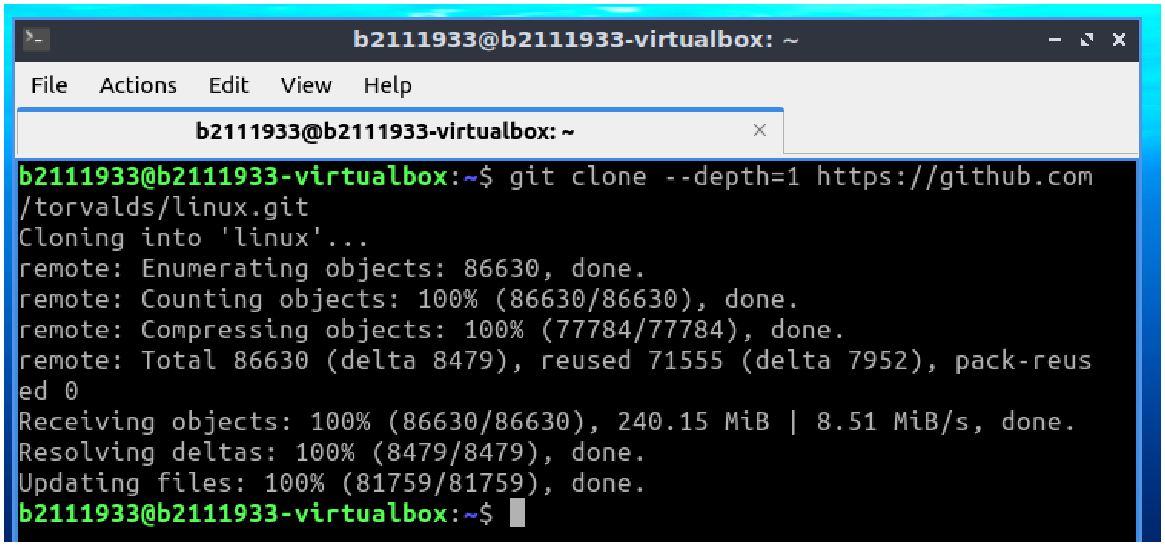
```
b2111933@b2111933-virtualbox:~$ sudo apt-get install build-essential v
im git cscope libncurses-dev libssl-dev bison flex libelf-dev bc git-e
mail -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
bc is already the newest version (1.07.1-3build1).
bc set to manually installed.
git is already the newest version (1:2.34.1-1ubuntu1.10).
vim is already the newest version (2:8.2.3995-1ubuntu2.11).
vim set to manually installed.
```

```
Setting up build-essential (12.9ubuntu3) ...
Processing triggers for libc-bin (2.35-0ubuntu3.1) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for install-info (6.8-4build1) ...
b2111933@b2111933-virtualbox:~$
```

Prepare the necessary things for the system

- Clone a mainline kernel source code to your computer:

```
    git clone --depth=1 \
    https://github.com/torvalds/linux.git
```

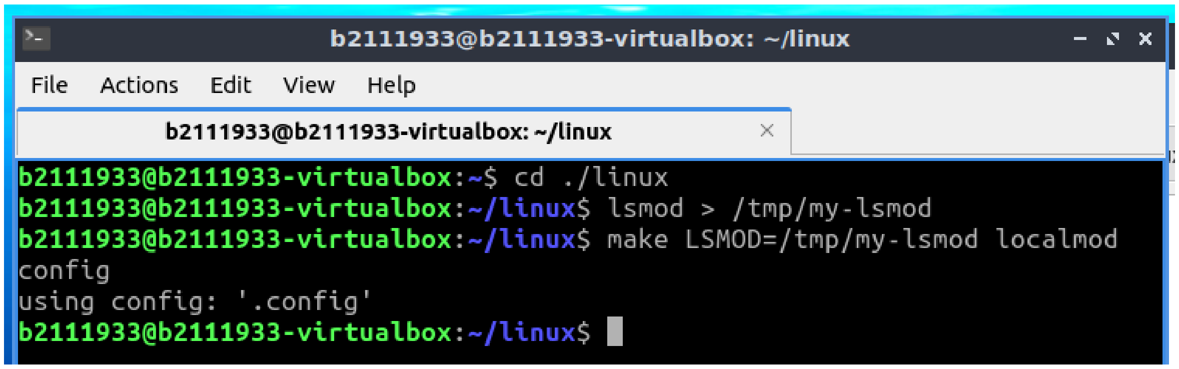

b2111933@b2111933-virtualbox: ~
File
Actions
Edit
View
Help
b2111933@b2111933-virtualbox: ~

```
b2111933@b2111933-virtualbox:~$ git clone --depth=1 https://github.com
/torvalds/linux.git
Cloning into 'linux'...
remote: Enumerating objects: 86630, done.
remote: Counting objects: 100% (86630/86630), done.
remote: Compressing objects: 100% (77784/77784), done.
remote: Total 86630 (delta 8479), reused 71555 (delta 7952), pack-reus
ed 0
Receiving objects: 100% (86630/86630), 240.15 MiB | 8.51 MiB/s, done.
Resolving deltas: 100% (8479/8479), done.
Updating files: 100% (81759/81759), done.
b2111933@b2111933-virtualbox:~$
```

Clone a mainline kernel source code to the VM

- To save time, just create a configuration file based on the list of modules currently loaded on your system (choose default values for other options).

```
lsmod > /tmp/my-lsmod
make LSMOD=/tmp/my-lsmod localmodconfig
```

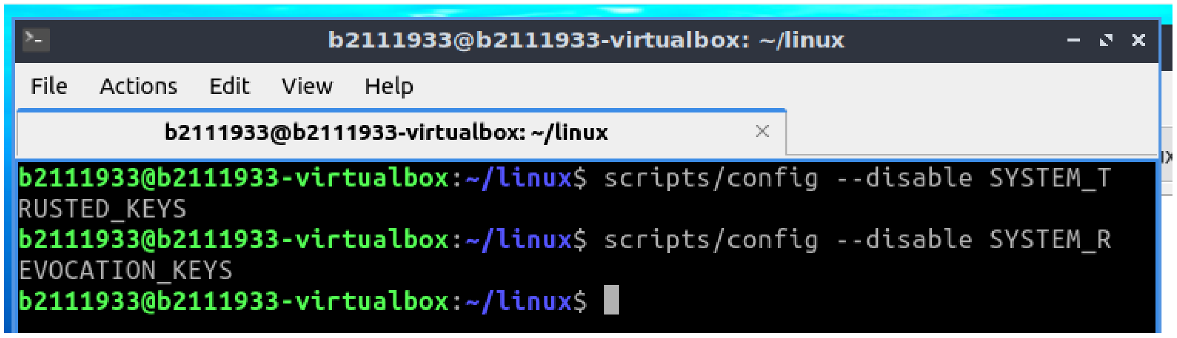

b2111933@b2111933-virtualbox: ~/linux
File
Actions
Edit
View
Help
b2111933@b2111933-virtualbox: ~/linux

```
b2111933@b2111933-virtualbox:~$ cd ./linux
b2111933@b2111933-virtualbox:~/linux$ lsmod > /tmp/my-lsmod
b2111933@b2111933-virtualbox:~/linux$ make LSMOD=/tmp/my-lsmod localmod
config
using config: '.config'
b2111933@b2111933-virtualbox:~/linux$
```

Create a configuration file

- Disable certificate stuff:

```
scripts/config --disable SYSTEM_TRUSTED_KEYS
scripts/config --disable SYSTEM_REVOCATION_KEYS
```

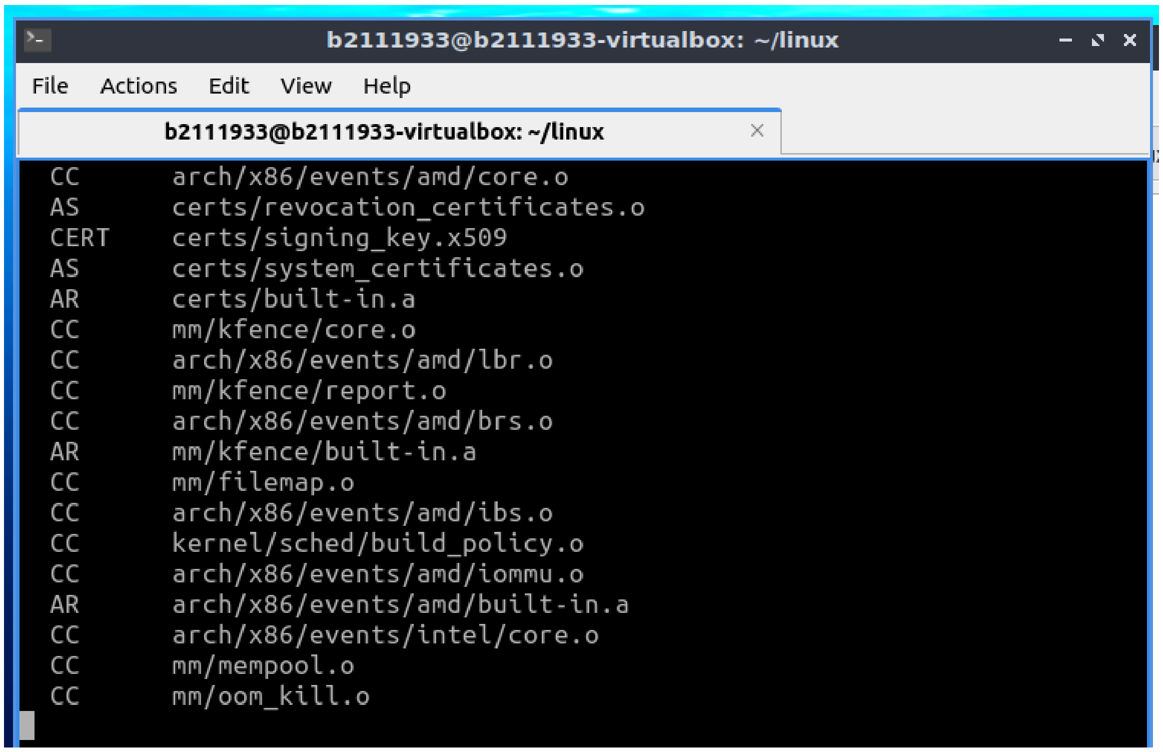

b2111933@b2111933-virtualbox: ~/linux
File
Actions
Edit
View
Help
b2111933@b2111933-virtualbox: ~/linux

```
b2111933@b2111933-virtualbox:~/linux$ scripts/config --disable SYSTEM_T
RUSTED_KEYS
b2111933@b2111933-virtualbox:~/linux$ scripts/config --disable SYSTEM_R
EVOCATION_KEYS
b2111933@b2111933-virtualbox:~/linux$
```

Disable certificate stuff

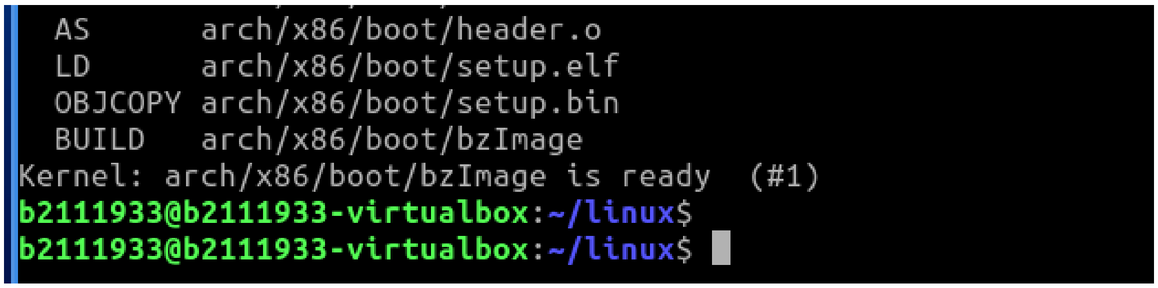
- Compile the kernel. The process takes about 1 hour, please be patient and enjoy a cup of coffee. It has been tested successfully on Lubuntu 20.04, if any errors occur, please try to fix them by yourself.

```
make -j3 all
```



b2111933@b2111933-virtualbox: ~/linux
File Actions Edit View Help
b2111933@b2111933-virtualbox: ~/linux

```
  CC      arch/x86/events/amd/core.o
  AS      certs/revocation_certificates.o
  CERT    certs/signing_key.x509
  AS      certs/system_certificates.o
  AR      certs/built-in.a
  CC      mm/kfence/core.o
  CC      arch/x86/events/amd/lbr.o
  CC      mm/kfence/report.o
  CC      arch/x86/events/amd/brs.o
  AR      mm/kfence/built-in.a
  CC      mm/filemap.o
  CC      arch/x86/events/amd/ibs.o
  CC      kernel/sched/build_policy.o
  CC      arch/x86/events/amd/iommu.o
  AR      arch/x86/events/amd/built-in.a
  CC      arch/x86/events/intel/core.o
  CC      mm/mempool.o
  CC      mm/oom_kill.o
```

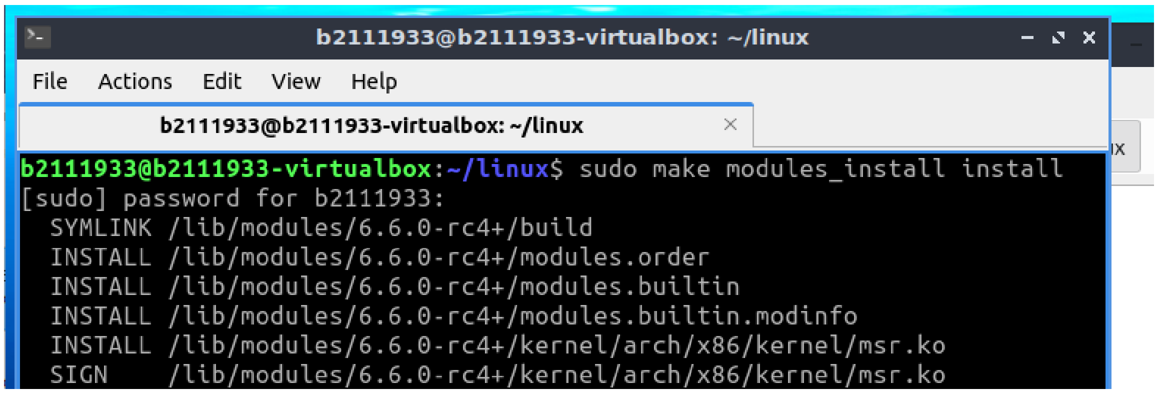


```
  AS      arch/x86/boot/header.o
  LD      arch/x86/boot/setup.elf
  OBJCOPY arch/x86/boot/setup.bin
  BUILD   arch/x86/boot/bzImage
Kernel: arch/x86/boot/bzImage is ready  (#1)
b2111933@b2111933-virtualbox:~/linux$
b2111933@b2111933-virtualbox:~/linux$
```

Build the new kernel

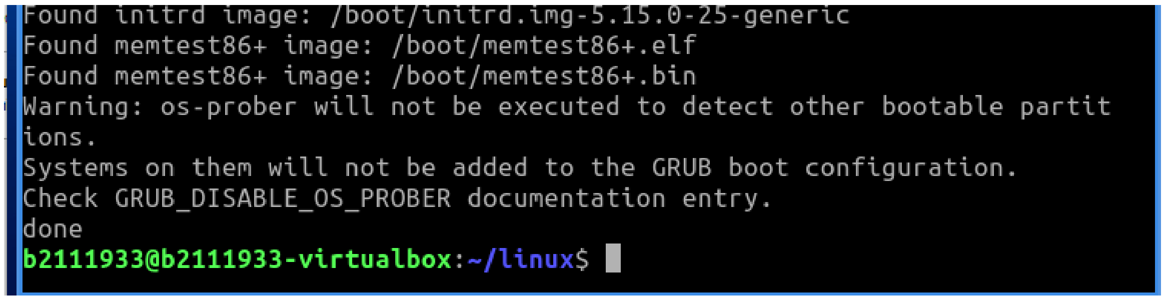
- Install the new kernel:

```
sudo make modules_install install
```



b2111933@b2111933-virtualbox: ~/linux
File Actions Edit View Help
b2111933@b2111933-virtualbox: ~/linux

```
b2111933@b2111933-virtualbox:~/linux$ sudo make modules_install install
[sudo] password for b2111933:
  SYMLINK /lib/modules/6.6.0-rc4+/build
  INSTALL /lib/modules/6.6.0-rc4+/modules.order
  INSTALL /lib/modules/6.6.0-rc4+/modules.builtin
  INSTALL /lib/modules/6.6.0-rc4+/modules.builtin.modinfo
  INSTALL /lib/modules/6.6.0-rc4+/kernel/arch/x86/kernel/msr.ko
  SIGN    /lib/modules/6.6.0-rc4+/kernel/arch/x86/kernel/msr.ko
```

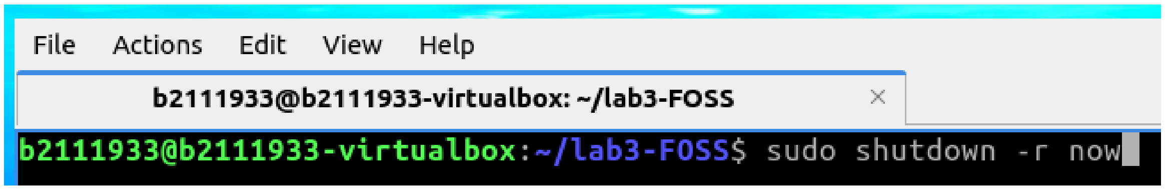


```
Found initrd image: /boot/initrd.img-5.15.0-25-generic
Found memtest86+ image: /boot/memtest86+.elf
Found memtest86+ image: /boot/memtest86+.bin
Warning: os-prober will not be executed to detect other bootable partit
ions.
Systems on them will not be added to the GRUB boot configuration.
Check GRUB_DISABLE_OS_PROBER documentation entry.
done
b2111933@b2111933-virtualbox:~/linux$
```

Install the new kernel

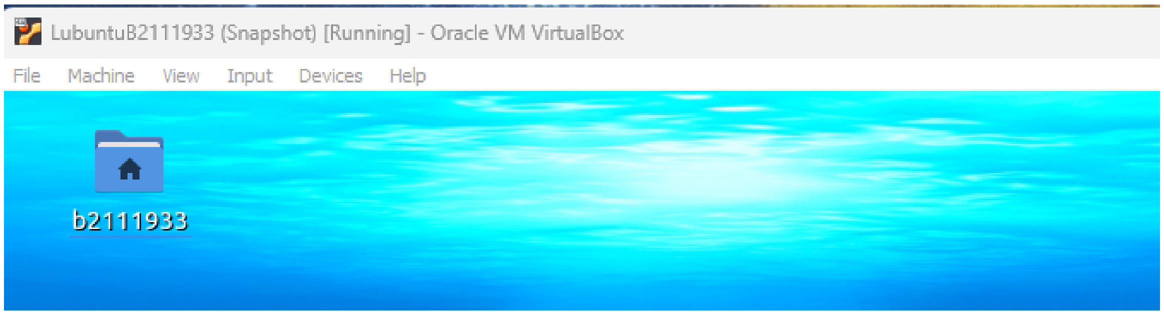
- Now it is time to reboot the system to boot the newly installed kernel:

```
sudo shutdown -r now
```



File Actions Edit View Help
b2111933@b2111933-virtualbox: ~/lab3-FOSS

```
b2111933@b2111933-virtualbox:~/lab3-FOSS$ sudo shutdown -r now
```

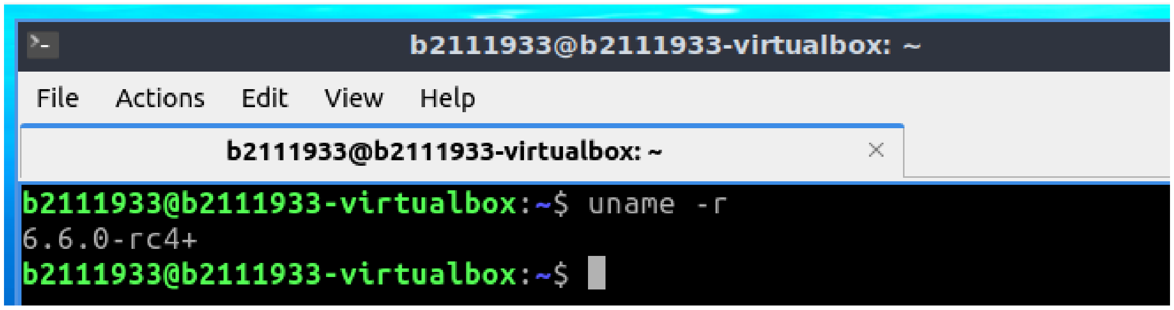


LubuntuB2111933 (Snapshot) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
b2111933

Shutdown & restart the VM

- Show your new current kernel version:

```
uname -r
```



b2111933@b2111933-virtualbox: ~
File Actions Edit View Help
b2111933@b2111933-virtualbox: ~

```
b2111933@b2111933-virtualbox:~$ uname -r
6.6.0-rc4+
b2111933@b2111933-virtualbox:~$
```

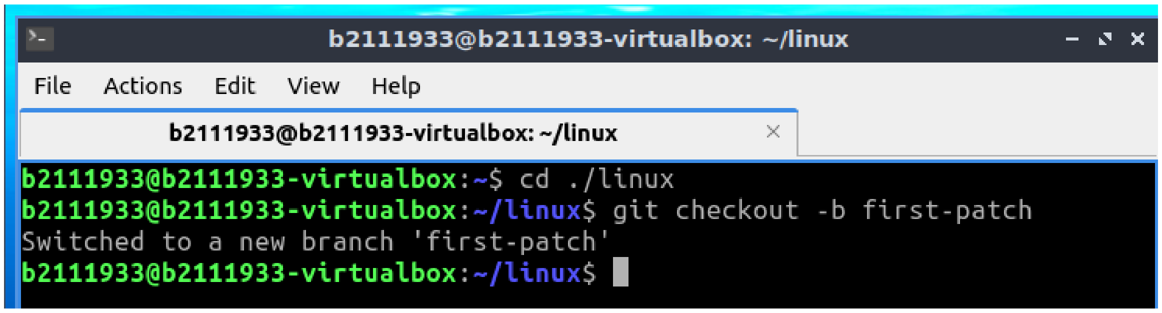
The new current kernel version is **6.6.0-rc4+**

(take screenshots to show that you finish this exercise)

4. Writing Your First Kernel Patch

- Creating a new branch in the linux_mainline repository (has been cloned in exercise 3)

```
git checkout -b first-patch
```



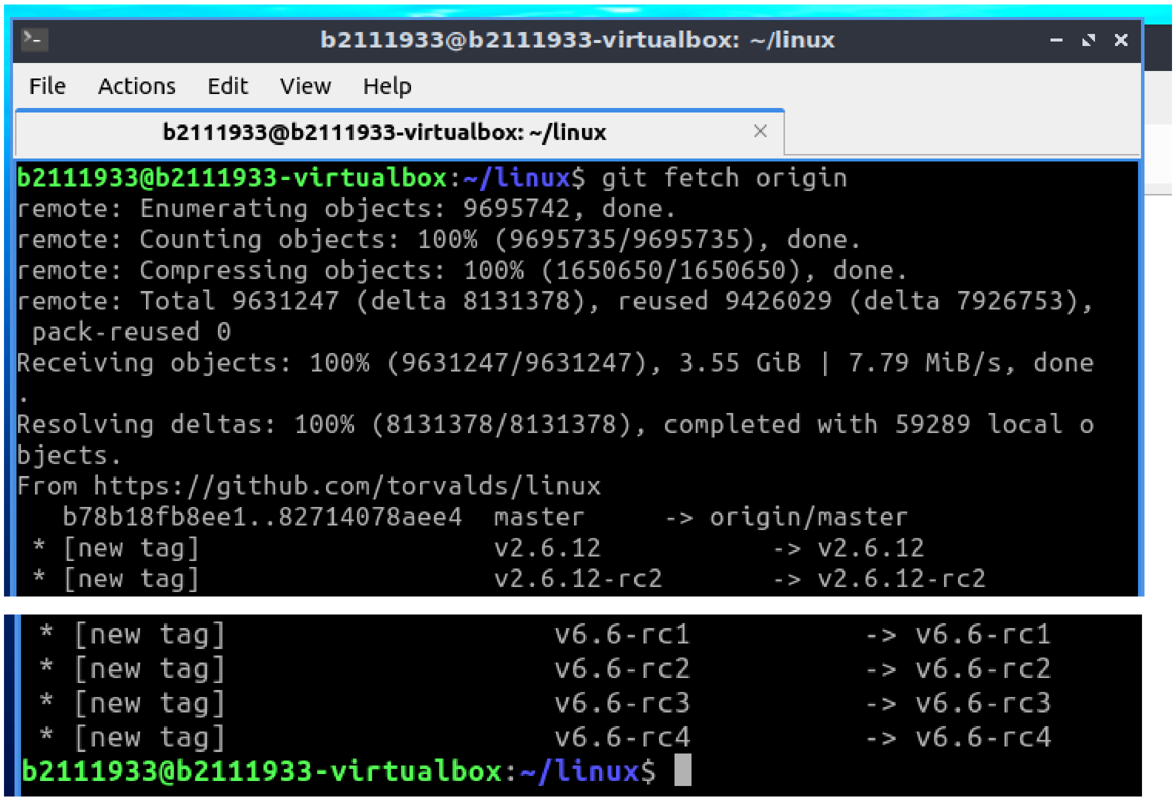
b2111933@b2111933-virtualbox: ~/linux
File Actions Edit View Help
b2111933@b2111933-virtualbox: ~/linux

```
b2111933@b2111933-virtualbox:~$ cd ./linux
b2111933@b2111933-virtualbox:~/linux$ git checkout -b first-patch
Switched to a new branch 'first-patch'
b2111933@b2111933-virtualbox:~/linux$
```

Create a new branch ‘**first-patch**’ in the **linux_mainline** repository

- Update the kernel

```
git fetch origin
```



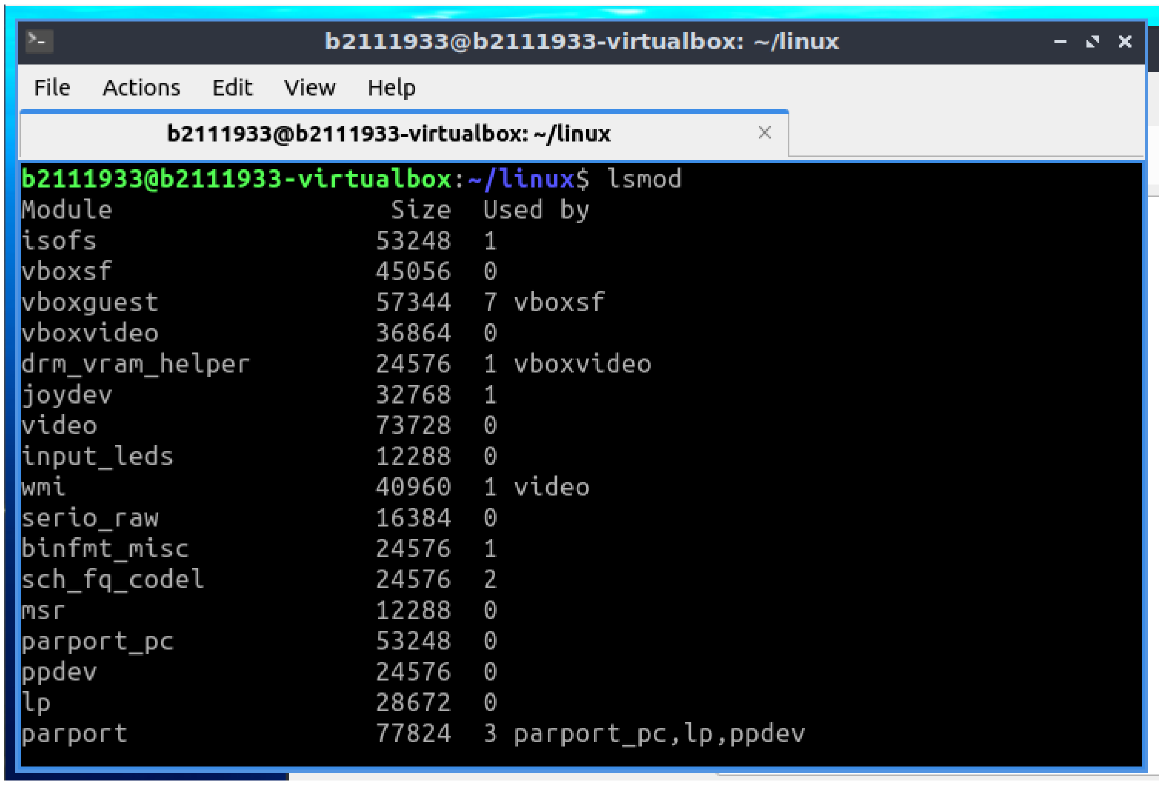
b2111933@b2111933-virtualbox: ~/linux
File Actions Edit View Help
b2111933@b2111933-virtualbox: ~/linux

```
b2111933@b2111933-virtualbox:~/linux$ git fetch origin
remote: Enumerating objects: 9695742, done.
remote: Counting objects: 100% (9695735/9695735), done.
remote: Compressing objects: 100% (1650650/1650650), done.
remote: Total 9631247 (delta 8131378), reused 9426029 (delta 7926753),
 pack-reused 0
Receiving objects: 100% (9631247/9631247), 3.55 GiB | 7.79 MiB/s, done
.
Resolving deltas: 100% (8131378/8131378), completed with 59289 local o
bjects.
From https://github.com/torvalds/linux
   b78b18fb8ee1..82714078aee4  master     -> origin/master
 * [new tag]                   v2.6.12          -> v2.6.12
 * [new tag]                   v2.6.12-rc2      -> v2.6.12-rc2
```

```
 * [new tag]               v6.6-rc1         -> v6.6-rc1
 * [new tag]               v6.6-rc2         -> v6.6-rc2
 * [new tag]               v6.6-rc3         -> v6.6-rc3
 * [new tag]               v6.6-rc4         -> v6.6-rc4
b2111933@b2111933-virtualbox:~/linux$
```

Update the kernel

- Run `lsmod` to see the modules loaded on your system, and pick a driver to change. One driver that's included in all VM images is the `e1000` driver, the Intel ethernet driver, or you can choose another driver depending on your working environment.

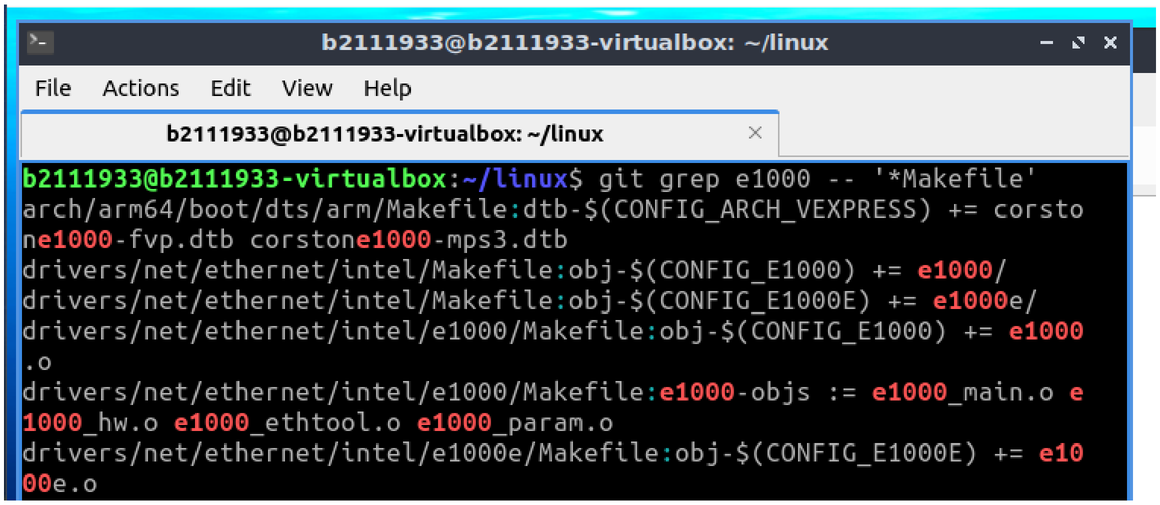

b2111933@b2111933-virtualbox: ~/linux
File
Actions
Edit
View
Help
b2111933@b2111933-virtualbox: ~/linux

```
b2111933@b2111933-virtualbox:~/linux$ lsmod
Module                  Size  Used by
isofs                  53248  1
vboxsf                 45056  0
vboxguest              57344  7 vboxsf
vboxvideo              36864  0
drm_vram_helper        24576  1 vboxvideo
joydev                 32768  1
video                  73728  0
input_leds             12288  0
wmi                    40960  1 video
serio_raw              16384  0
binfmt_misc            24576  1
sch_fq_codel           24576  2
msr                    12288  0
parport_pc             53248  0
ppdev                  24576  0
lp                     28672  0
parport                77824  3 parport_pc,lp,ppdev
```

List all modules

- Run `git grep` to look for `e1000` files

```
    git grep e1000 -- '*Makefile'
```

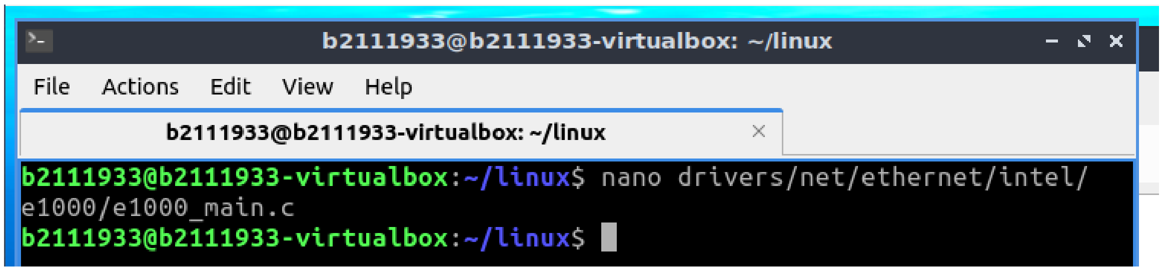

b2111933@b2111933-virtualbox: ~/linux
File
Actions
Edit
View
Help
b2111933@b2111933-virtualbox: ~/linux

```
b2111933@b2111933-virtualbox:~/linux$ git grep e1000 -- '*Makefile'
arch/arm64/boot/dts/arm/Makefile:dtb-$(CONFIG_ARCH_VEXPRESS) += corsto
ne1000-fvp.dtb corstone1000-mps3.dtb
drivers/net/ethernet/intel/Makefile:obj-$(CONFIG_E1000) += e1000/
drivers/net/ethernet/intel/Makefile:obj-$(CONFIG_E1000E) += e1000e/
drivers/net/ethernet/intel/e1000/Makefile:obj-$(CONFIG_E1000) += e1000
.o
drivers/net/ethernet/intel/e1000/Makefile:e1000-objs := e1000_main.o e
1000_hw.o e1000_ethtool.o e1000_param.o
drivers/net/ethernet/intel/e1000e/Makefile:obj-$(CONFIG_E1000E) += e10
00e.o
```

Look for **e1000** files

- Make a small change to the probe function of the e1000 driver

```
nano drivers/net/ethernet/intel/e1000/e1000_main.c
```

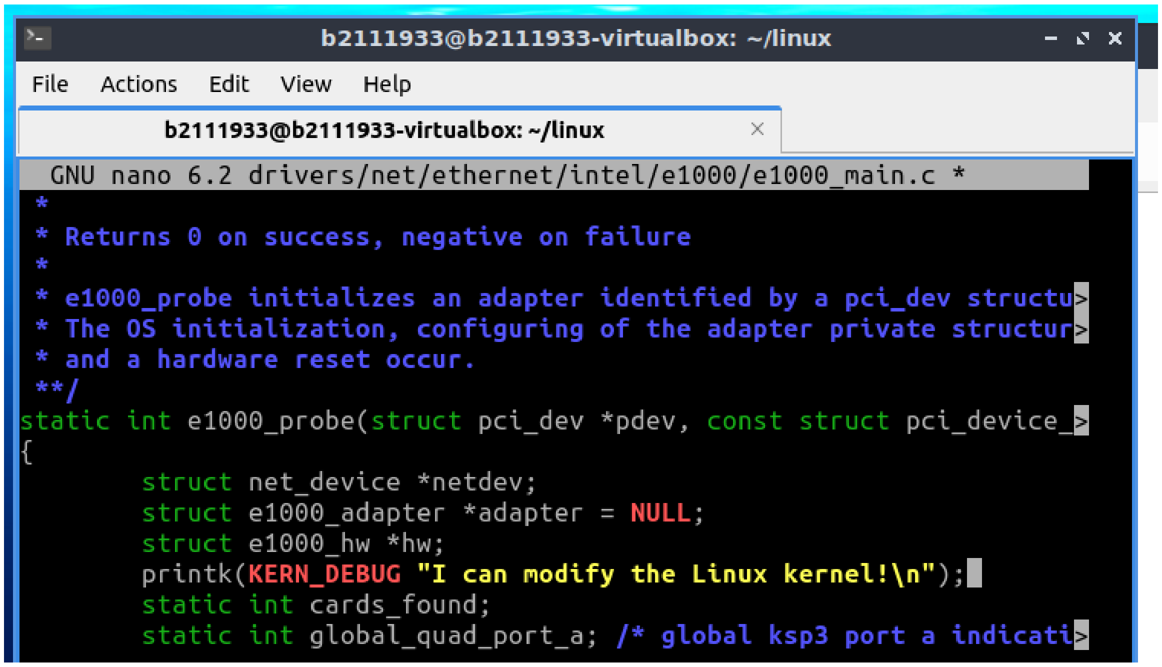


b2111933@b2111933-virtualbox: ~/linux
File Actions Edit View Help
b2111933@b2111933-virtualbox: ~/linux

```
b2111933@b2111933-virtualbox:~/linux$ nano drivers/net/ethernet/intel/
e1000/e1000_main.c
b2111933@b2111933-virtualbox:~/linux$
```

Open the file of driver with **nano**

```
    # Add a line of code as below
    static  int  e1000_probe(struct  pci_dev  *pdev,  const
struct pci_device_id *ent) {
    ...
    struct e1000_hw *hw;
    printk(KERN_DEBUG "I can modify the Linux kernel!\n");
    static int cards_found = 0;
    ...
```



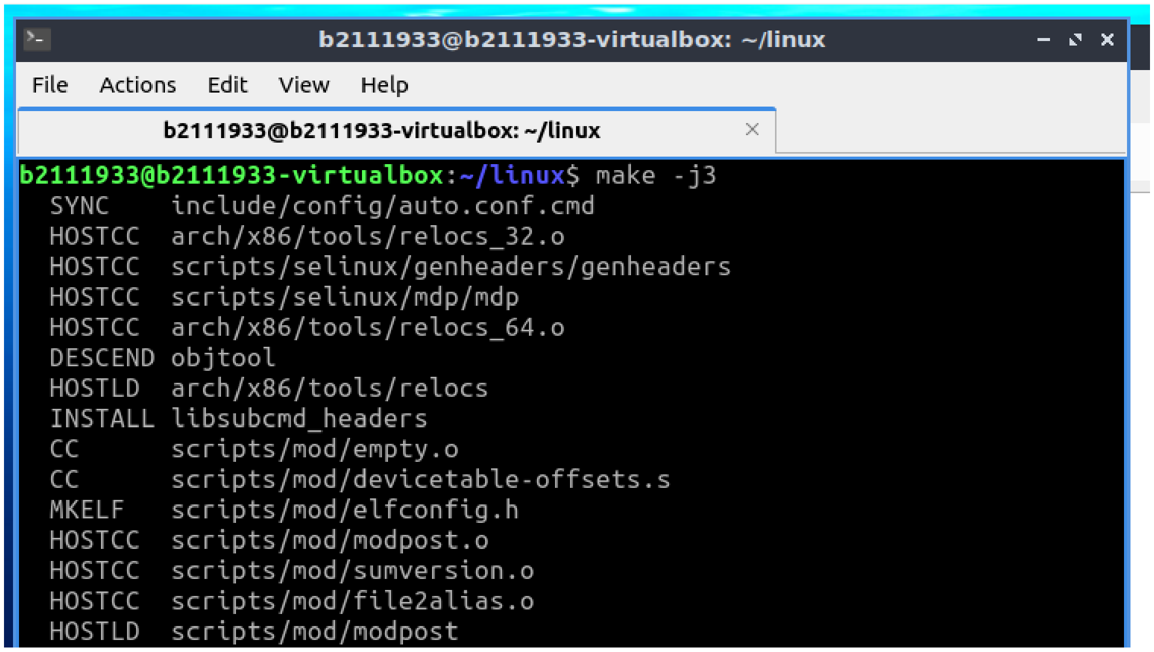
b2111933@b2111933-virtualbox: ~/linux
File Actions Edit View Help
b2111933@b2111933-virtualbox: ~/linux

```
 GNU nano 6.2 drivers/net/ethernet/intel/e1000/e1000_main.c *
 *
 * Returns 0 on success, negative on failure
 *
 * e1000_probe initializes an adapter identified by a pci_dev structu>
 * The OS initialization, configuring of the adapter private structur>
 * and a hardware reset occur.
 **/
static int e1000_probe(struct pci_dev *pdev, const struct pci_device_>
{
        struct net_device *netdev;
        struct e1000_adapter *adapter = NULL;
        struct e1000_hw *hw;
        printk(KERN_DEBUG "I can modify the Linux kernel!\n");
        static int cards_found;
        static int global_quad_port_a; /* global ksp3 port a indicati>
```

Add a line of code

- Compile and install your changes:

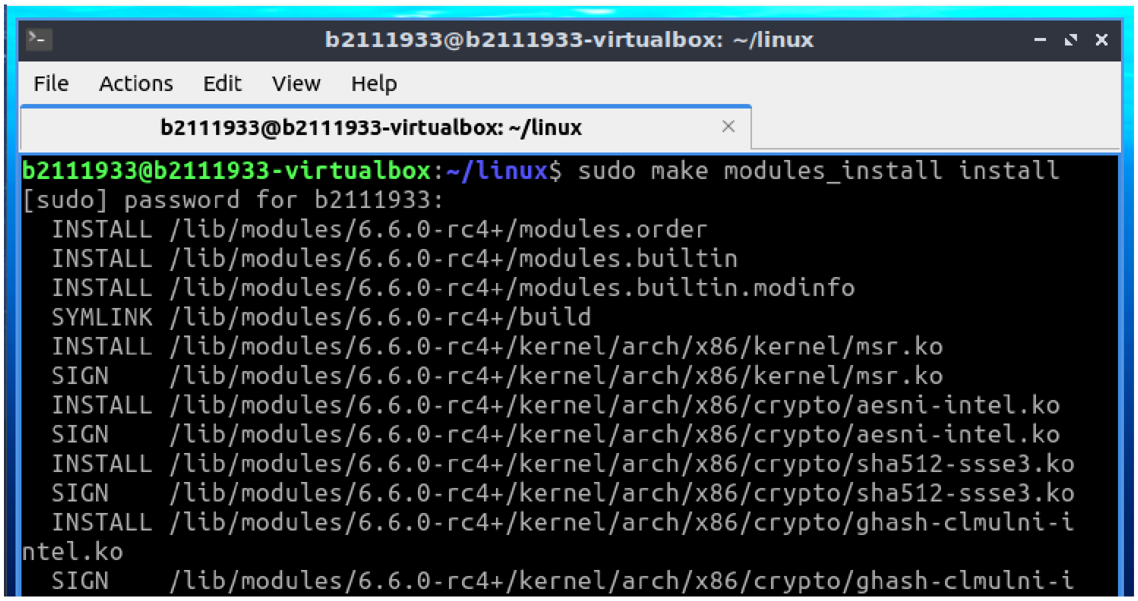
```
make -j3
```


b2111933@b2111933-virtualbox: ~/linux
File Actions Edit View Help
b2111933@b2111933-virtualbox: ~/linux

```
b2111933@b2111933-virtualbox:~/linux$ make -j3
  SYNC    include/config/auto.conf.cmd
  HOSTCC  arch/x86/tools/relocs_32.o
  HOSTCC  scripts/selinux/genheaders/genheaders
  HOSTCC  scripts/selinux/mdp/mdp
  HOSTCC  arch/x86/tools/relocs_64.o
  DESCEND objtool
  HOSTLD  arch/x86/tools/relocs
  INSTALL libsubcmd_headers
  CC      scripts/mod/empty.o
  CC      scripts/mod/devicetable-offsets.s
  MKELF   scripts/mod/elfconfig.h
  HOSTCC  scripts/mod/modpost.o
  HOSTCC  scripts/mod/sumversion.o
  HOSTCC  scripts/mod/file2alias.o
  HOSTLD  scripts/mod/modpost
```

Compile the changes

```
sudo make modules_install install
```

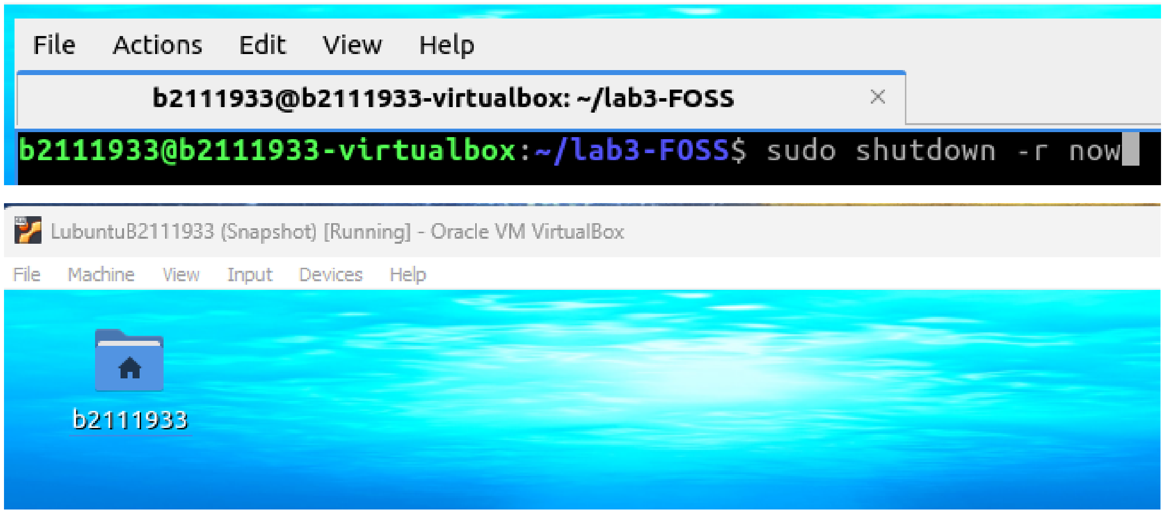

b2111933@b2111933-virtualbox: ~/linux
File Actions Edit View Help
b2111933@b2111933-virtualbox: ~/linux

```
b2111933@b2111933-virtualbox:~/linux$ sudo make modules_install install
[sudo] password for b2111933:
  INSTALL /lib/modules/6.6.0-rc4+/modules.order
  INSTALL /lib/modules/6.6.0-rc4+/modules.builtin
  INSTALL /lib/modules/6.6.0-rc4+/modules.builtin.modinfo
  SYMLINK /lib/modules/6.6.0-rc4+/build
  INSTALL /lib/modules/6.6.0-rc4+/kernel/arch/x86/kernel/msr.ko
  SIGN    /lib/modules/6.6.0-rc4+/kernel/arch/x86/kernel/msr.ko
  INSTALL /lib/modules/6.6.0-rc4+/kernel/arch/x86/crypto/aesni-intel.ko
  SIGN    /lib/modules/6.6.0-rc4+/kernel/arch/x86/crypto/aesni-intel.ko
  INSTALL /lib/modules/6.6.0-rc4+/kernel/arch/x86/crypto/sha512-ssse3.ko
  SIGN    /lib/modules/6.6.0-rc4+/kernel/arch/x86/crypto/sha512-ssse3.ko
  INSTALL /lib/modules/6.6.0-rc4+/kernel/arch/x86/crypto/ghash-clmulni-i
ntel.ko
  SIGN    /lib/modules/6.6.0-rc4+/kernel/arch/x86/crypto/ghash-clmulni-i
```

Install the changes

- Reboot the system:

```
sudo shutdown -r now
```

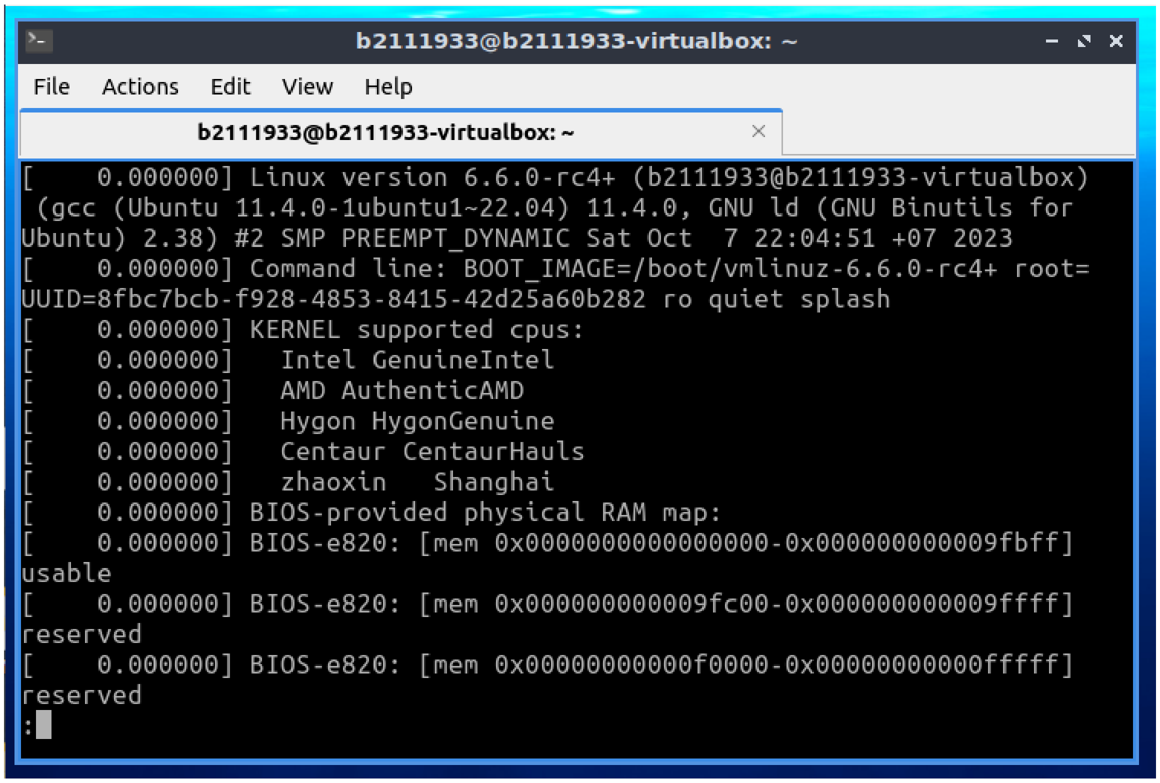

File Actions Edit View Help
b2111933@b2111933-virtualbox: ~/lab3-FOSS
LubuntuB2111933 (Snapshot) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
b2111933

```
b2111933@b2111933-virtualbox:~/lab3-FOSS$ sudo shutdown -r now
```

Reboot the VM

- Show kernel buffer log:

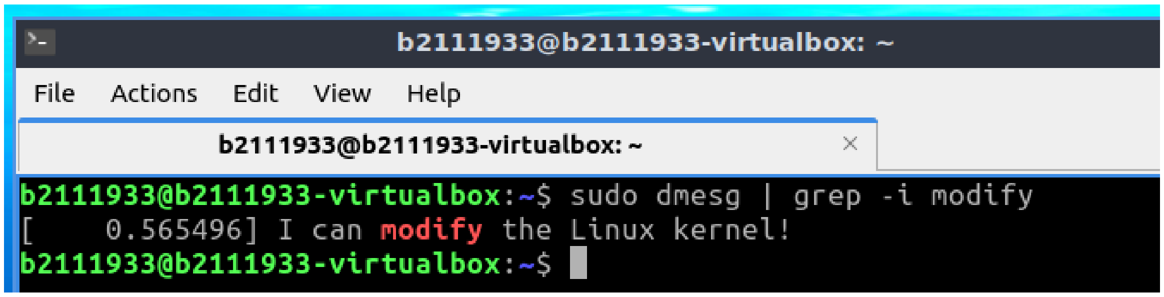
```
dmesg | less
```


b2111933@b2111933-virtualbox: ~
File Actions Edit View Help
b2111933@b2111933-virtualbox: ~

```
[    0.000000] Linux version 6.6.0-rc4+ (b2111933@b2111933-virtualbox)
 (gcc (Ubuntu 11.4.0-1ubuntu1~22.04) 11.4.0, GNU ld (GNU Binutils for
Ubuntu) 2.38) #2 SMP PREEMPT_DYNAMIC Sat Oct  7 22:04:51 +07 2023
[    0.000000] Command line: BOOT_IMAGE=/boot/vmlinuz-6.6.0-rc4+ root=
UUID=8fbc7bcb-f928-4853-8415-42d25a60b282 ro quiet splash
[    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] BIOS-provided physical RAM map:
[    0.000000] BIOS-e820: [mem 0x0000000000000000-0x000000000009fbff]
usable
[    0.000000] BIOS-e820: [mem 0x000000000009fc00-0x000000000009ffff]
reserved
[    0.000000] BIOS-e820: [mem 0x00000000000f0000-0x00000000000fffff]
reserved
:
```

Show kernel buffer log

```
    # Search for your printk in the log file by typing "/I
can modify"
```

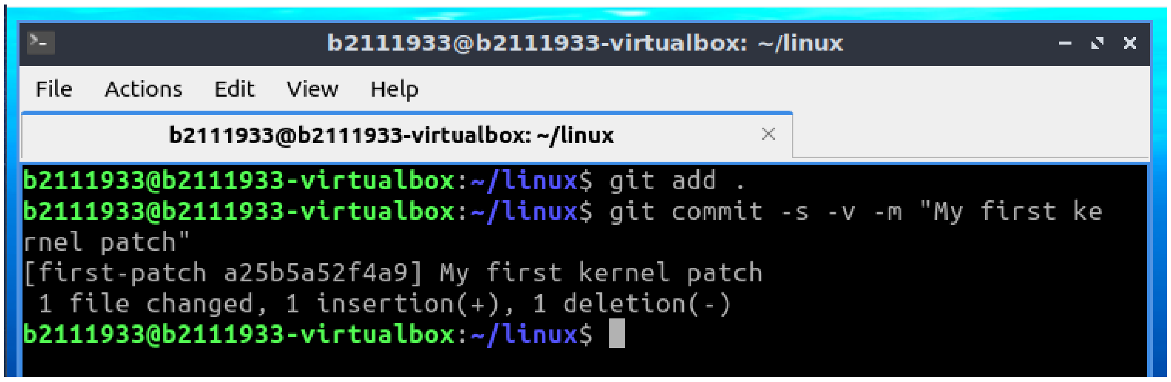


```
b2111933@b2111933-virtualbox:~$ sudo dmesg | grep -i modify
[    0.565496] I can modify the Linux kernel!
b2111933@b2111933-virtualbox:~$
```

Search for **printk** in the log

- Committing changes, and view your commit

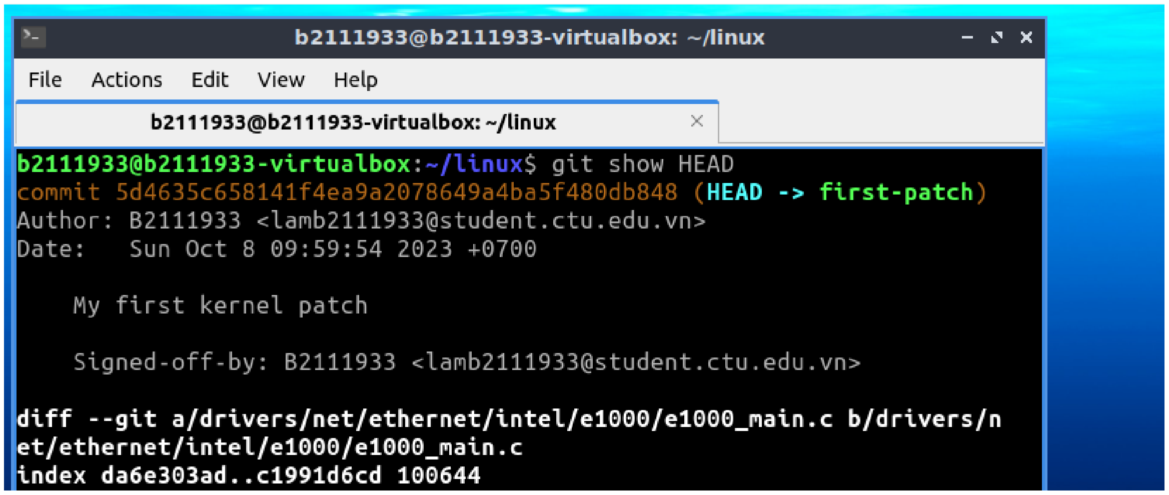
```
    git add .
    git commit -s -v -m "My first kernel patch"
```



```
b2111933@b2111933-virtualbox:~/linux$ git add .
b2111933@b2111933-virtualbox:~/linux$ git commit -s -v -m "My first ke
rnel patch"
[first-patch a25b5a52f4a9] My first kernel patch
 1 file changed, 1 insertion(+), 1 deletion(-)
b2111933@b2111933-virtualbox:~/linux$
```

Commit changes

```
    git show HEAD
```

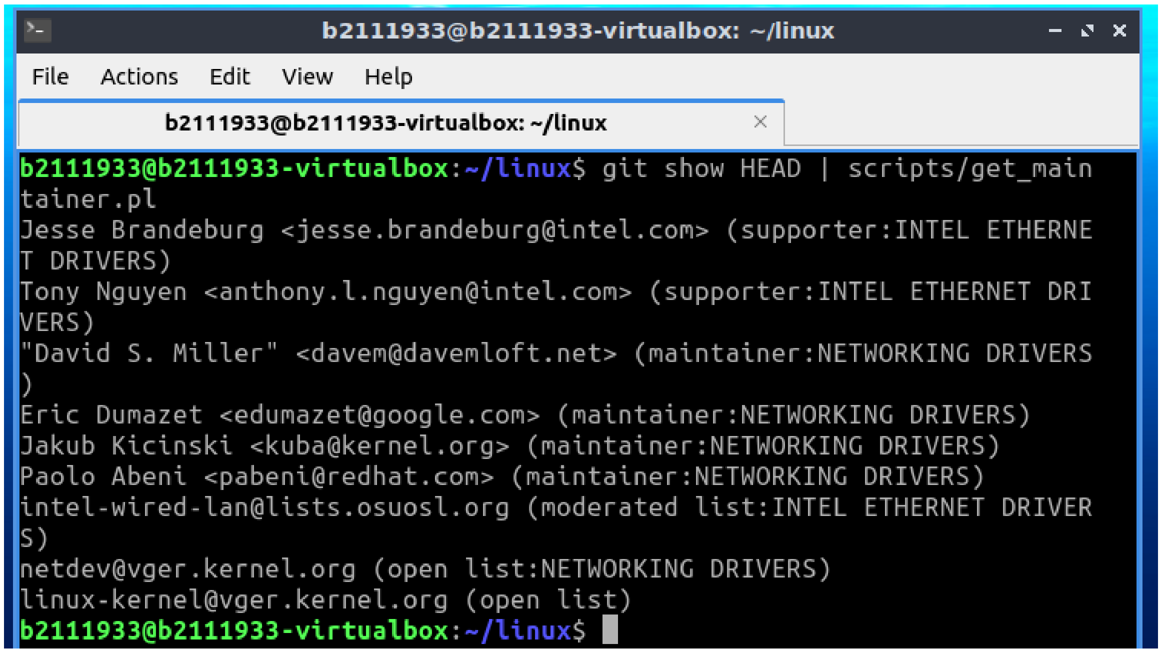


```
b2111933@b2111933-virtualbox:~/linux$ git show HEAD
commit 5d4635c658141f4ea9a2078649a4ba5f480db848 (HEAD -> first-patch)
Author: B2111933 <lamb2111933@student.ctu.edu.vn>
Date:   Sun Oct 8 09:59:54 2023 +0700

    My first kernel patch

    Signed-off-by: B2111933 <lamb2111933@student.ctu.edu.vn>

diff --git a/drivers/net/ethernet/intel/e1000/e1000_main.c b/drivers/n
et/ethernet/intel/e1000/e1000_main.c
index da6e303ad..c1991d6cd 100644
```

View the commit, the commit id is **5d4635c658141f4ea9a2078649a4ba5f480db848**

- Find whom to send the patch to

```
git show HEAD | scripts/get_maintainer.pl
```

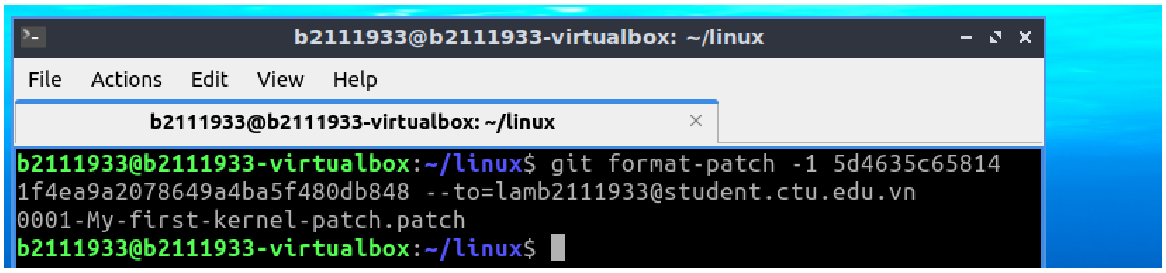

b2111933@b2111933-virtualbox: ~/linux
File Actions Edit View Help
b2111933@b2111933-virtualbox: ~/linux

```
b2111933@b2111933-virtualbox:~/linux$ git show HEAD | scripts/get_main
tainer.pl
Jesse Brandeburg <jesse.brandeburg@intel.com> (supporter:INTEL ETHERNE
T DRIVERS)
Tony Nguyen <anthony.l.nguyen@intel.com> (supporter:INTEL ETHERNET DRI
VERS)
"David S. Miller" <davem@davemloft.net> (maintainer:NETWORKING DRIVERS
)
Eric Dumazet <edumazet@google.com> (maintainer:NETWORKING DRIVERS)
Jakub Kicinski <kuba@kernel.org> (maintainer:NETWORKING DRIVERS)
Paolo Abeni <pabeni@redhat.com> (maintainer:NETWORKING DRIVERS)
intel-wired-lan@lists.osuosl.org (moderated list:INTEL ETHERNET DRIVER
S)
netdev@vger.kernel.org (open list:NETWORKING DRIVERS)
linux-kernel@vger.kernel.org (open list)
b2111933@b2111933-virtualbox:~/linux$
```

Find whom to send the patch to

- Create a patch

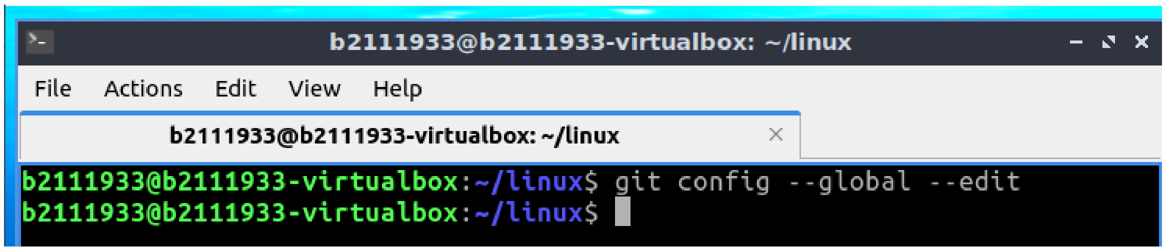
```
git format-patch -1 <commit ID> --to=<your email>
Note: Please do not send your patch to a maintainer,
send it to yourself instead.
```


b2111933@b2111933-virtualbox: ~/linux
File Actions Edit View Help
b2111933@b2111933-virtualbox: ~/linux

```
b2111933@b2111933-virtualbox:~/linux$ git format-patch -1 5d4635c65814
1f4ea9a2078649a4ba5f480db848 --to=lamb2111933@student.ctu.edu.vn
0001-My-first-kernel-patch.patch
b2111933@b2111933-virtualbox:~/linux$
```

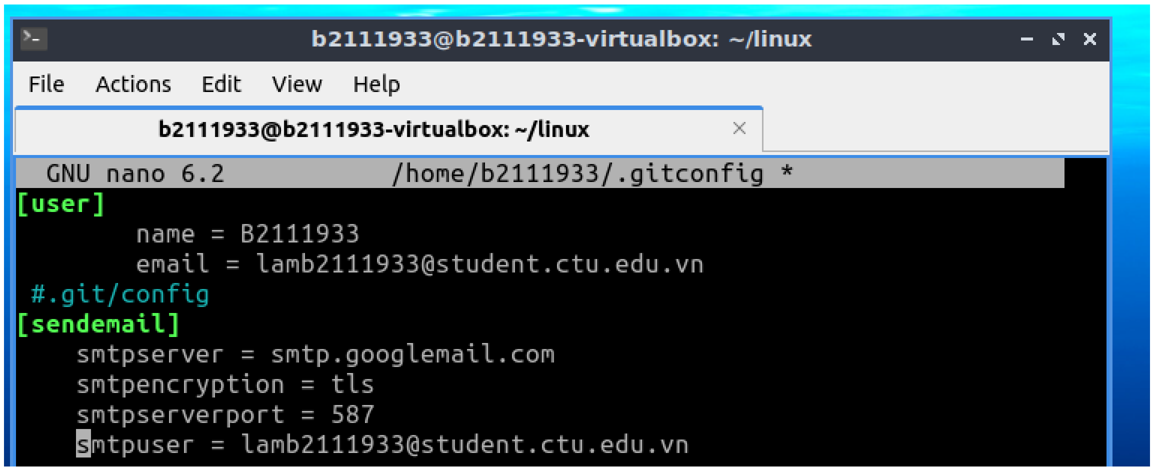
Create a patch

- Modify `./git/config` file to configure send-email


b2111933@b2111933-virtualbox: ~/linux
File Actions Edit View Help
b2111933@b2111933-virtualbox: ~/linux

```
b2111933@b2111933-virtualbox:~/linux$ git config --global --edit
b2111933@b2111933-virtualbox:~/linux$
```

```
#.git/config
[sendemail]
     smtpserver = smtp.googlemail.com
     smtpencryption = tls
     smtpserverport = 587
    smtpuser = your gmail address (CTU student email is
OK
```

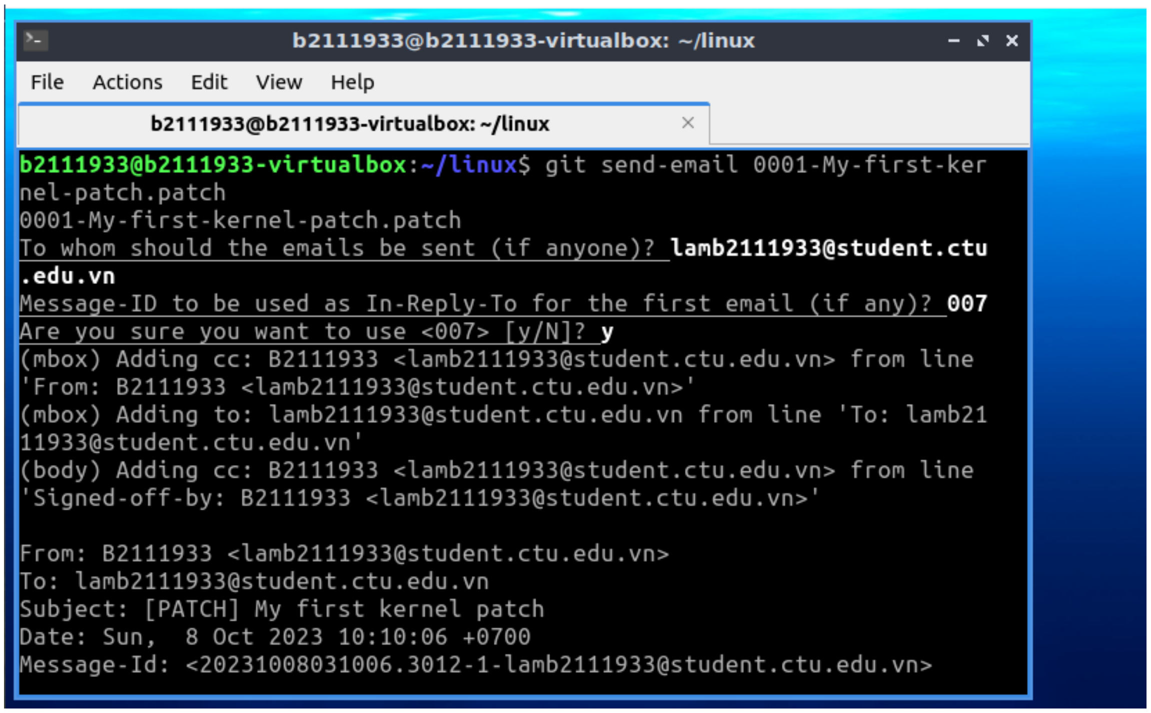


```
GNU nano 6.2              /home/b2111933/.gitconfig *
[user]
        name = B2111933
        email = lamb2111933@student.ctu.edu.vn
 #.git/config
[sendemail]
    smtpserver = smtp.googlemail.com
    smtpencryption = tls
    smtpserverport = 587
    smtpuser = lamb2111933@student.ctu.edu.vn
```

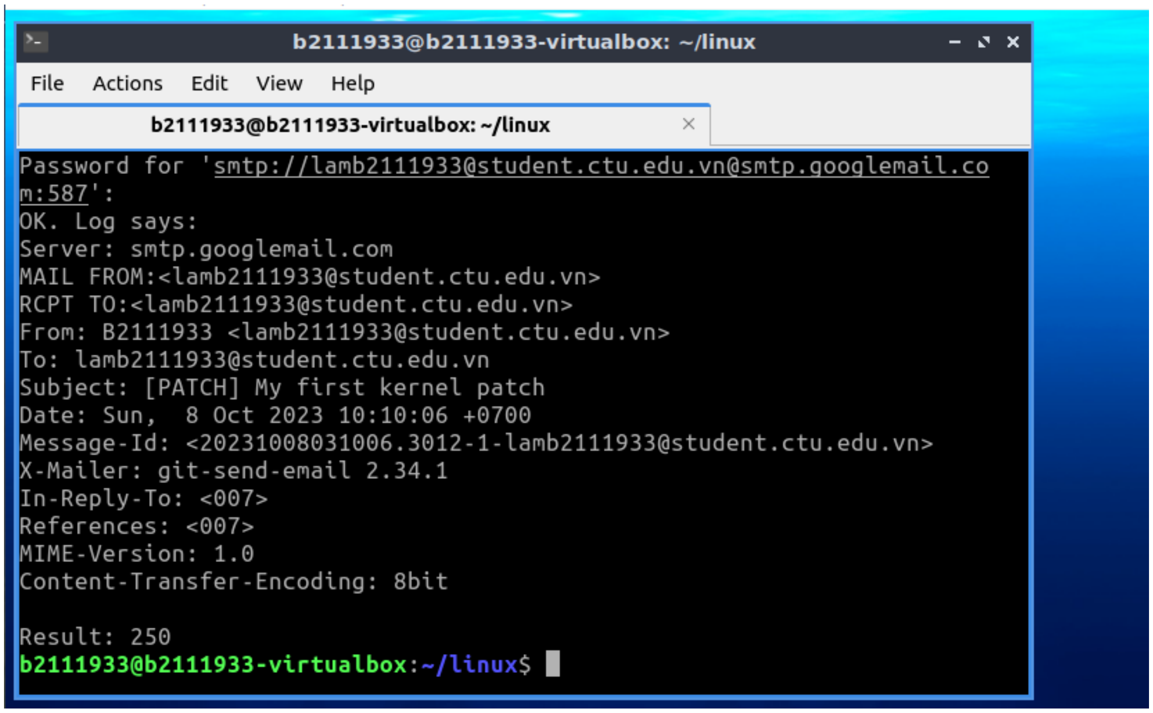
Modify **./git/config** file to configure send-email

- Send the patch

```
git send-email <patch_file>
```

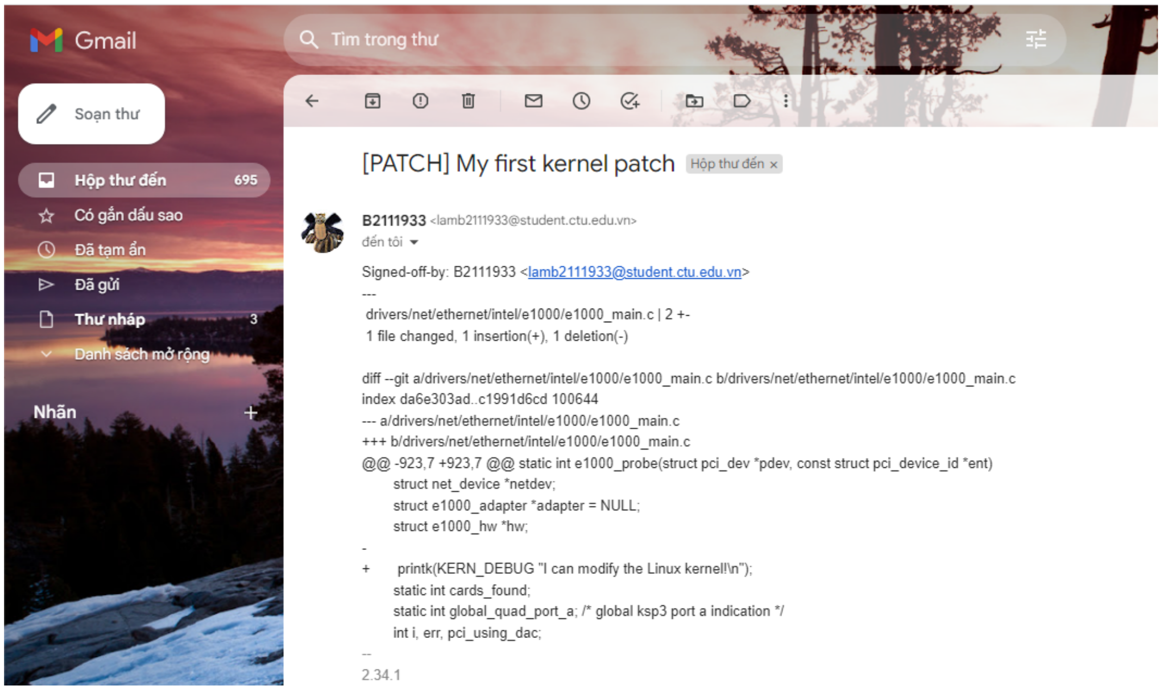


```
b2111933@b2111933-virtualbox:~/linux$ git send-email 0001-My-first-ker
nel-patch.patch
0001-My-first-kernel-patch.patch
To whom should the emails be sent (if anyone)? lamb2111933@student.ctu
.edu.vn
Message-ID to be used as In-Reply-To for the first email (if any)? 007
Are you sure you want to use <007> [y/N]? y
(mbox) Adding cc: B2111933 <lamb2111933@student.ctu.edu.vn> from line
'From: B2111933 <lamb2111933@student.ctu.edu.vn>'
(mbox) Adding to: lamb2111933@student.ctu.edu.vn from line 'To: lamb21
11933@student.ctu.edu.vn'
(body) Adding cc: B2111933 <lamb2111933@student.ctu.edu.vn> from line
'Signed-off-by: B2111933 <lamb2111933@student.ctu.edu.vn>'

From: B2111933 <lamb2111933@student.ctu.edu.vn>
To: lamb2111933@student.ctu.edu.vn
Subject: [PATCH] My first kernel patch
Date: Sun,  8 Oct 2023 10:10:06 +0700
Message-Id: <20231008031006.3012-1-lamb2111933@student.ctu.edu.vn>
```


b2111933@b2111933-virtualbox: ~/linux
File Actions Edit View Help
b2111933@b2111933-virtualbox: ~/linux

```
Password for 'smtp://lamb2111933@student.ctu.edu.vn@smtp.googlemail.co
m:587':
OK. Log says:
Server: smtp.googlemail.com
MAIL FROM:<lamb2111933@student.ctu.edu.vn>
RCPT TO:<lamb2111933@student.ctu.edu.vn>
From: B2111933 <lamb2111933@student.ctu.edu.vn>
To: lamb2111933@student.ctu.edu.vn
Subject: [PATCH] My first kernel patch
Date: Sun,  8 Oct 2023 10:10:06 +0700
Message-Id: <20231008031006.3012-1-lamb2111933@student.ctu.edu.vn>
X-Mailer: git-send-email 2.34.1
In-Reply-To: <007>
References: <007>
MIME-Version: 1.0
Content-Transfer-Encoding: 8bit

Result: 250
b2111933@b2111933-virtualbox:~/linux$
```

Send the patch successfully



```
Signed-off-by: B2111933 <lamb2111933@student.ctu.edu.vn>
---
 drivers/net/ethernet/intel/e1000/e1000_main.c | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)

diff --git a/drivers/net/ethernet/intel/e1000/e1000_main.c b/drivers/net/ethernet/intel/e1000/e1000_main.c
index da6e303ad..c1991d6cd 100644
--- a/drivers/net/ethernet/intel/e1000/e1000_main.c
+++ b/drivers/net/ethernet/intel/e1000/e1000_main.c
@@ -923,7 +923,7 @@ static int e1000_probe(struct pci_dev *pdev, const struct pci_device_id *ent)
        struct net_device *netdev;
        struct e1000_adapter *adapter = NULL;
        struct e1000_hw *hw;
-
+       printk(KERN_DEBUG "I can modify the Linux kernel!\n");
        static int cards_found;
        static int global_quad_port_a; /* global ksp3 port a indication */
        int i, err, pci_using_dac;
--
2.34.1
```

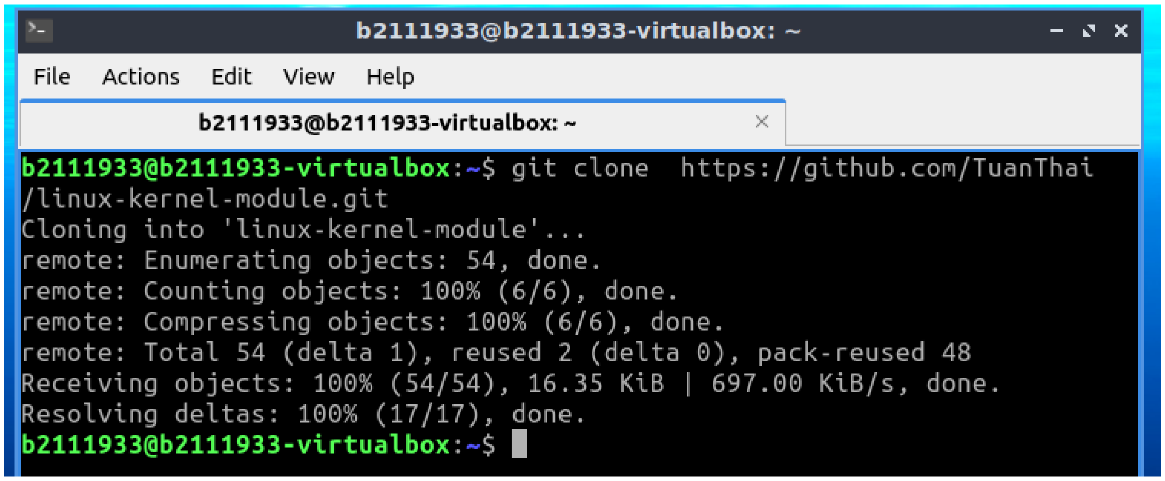
Check my gmail

(take screenshots to show that you finish this exercise)

5. Writing a simple Linux kernel module: Greeter sample

This module simply takes a name as a parameter, and writes a greeting to the kernel log (/var/log/kern.log):

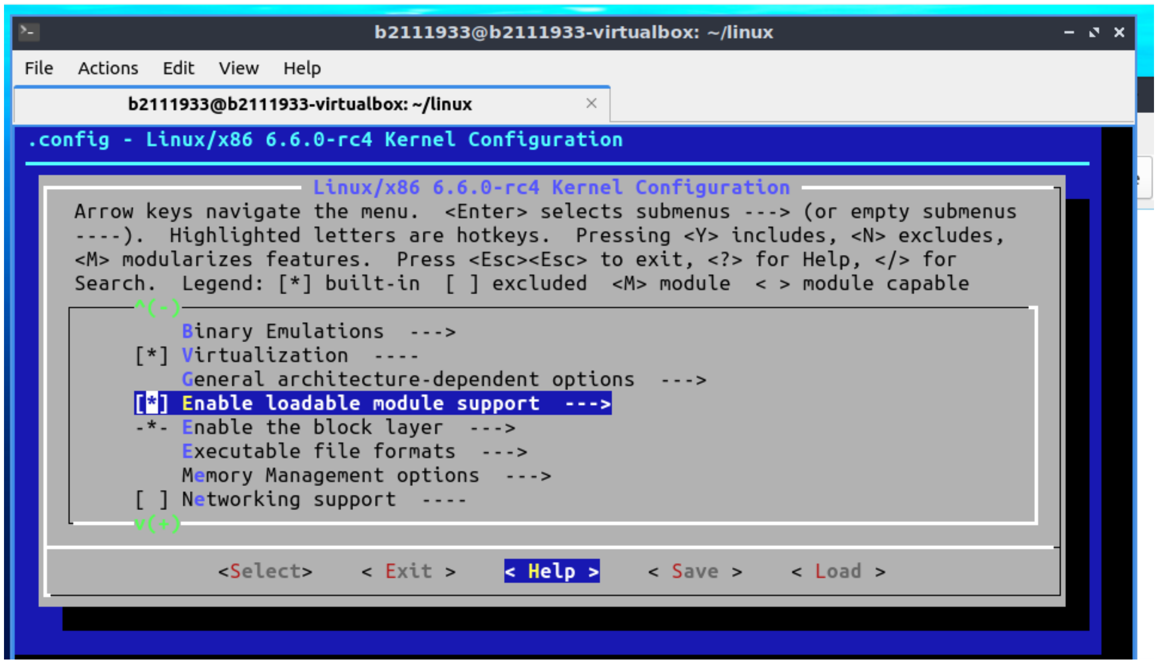
- Clone this repository to your computer:
https://github.com/TuanThai/linux-kernel-module.git



b2111933@b2111933-virtualbox: ~
File Actions Edit View Help
b2111933@b2111933-virtualbox: ~

```
b2111933@b2111933-virtualbox:~$ git clone  https://github.com/TuanThai
/linux-kernel-module.git
Cloning into 'linux-kernel-module'...
remote: Enumerating objects: 54, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 54 (delta 1), reused 2 (delta 0), pack-reused 48
Receiving objects: 100% (54/54), 16.35 KiB | 697.00 KiB/s, done.
Resolving deltas: 100% (17/17), done.
b2111933@b2111933-virtualbox:~$
```

Clone https://github.com/TuanThai/linux-kernel-module.git repository to the VM

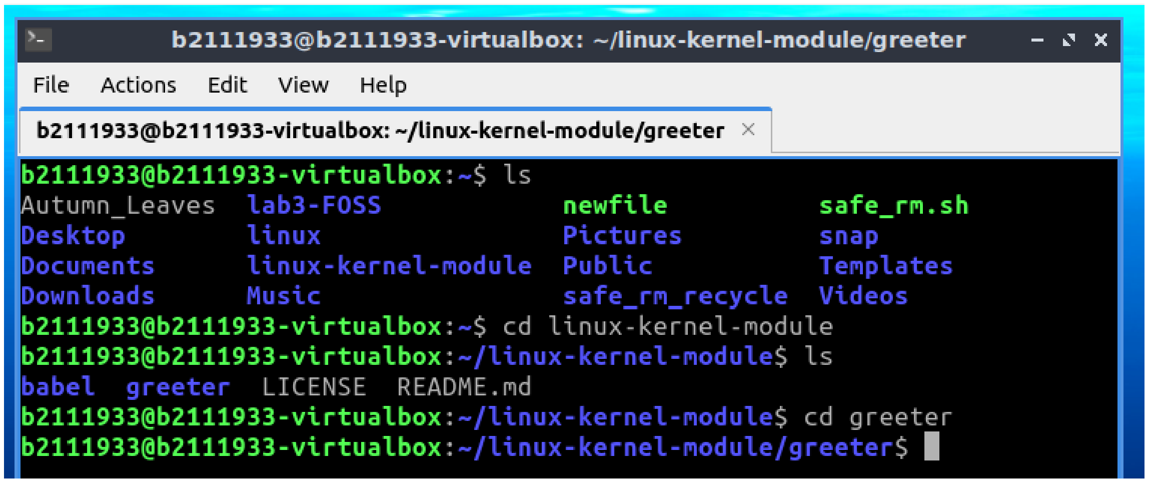


b2111933@b2111933-virtualbox: ~/linux
File Actions Edit View Help
b2111933@b2111933-virtualbox: ~/linux

```
.config - Linux/x86 6.6.0-rc4 Kernel Configuration
                Linux/x86 6.6.0-rc4 Kernel Configuration
  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty submenus
  ----).  Highlighted letters are hotkeys.  Pressing <Y> includes, <N> excludes,
  <M> modularizes features.  Press <Esc><Esc> to exit, <?> for Help, </> for
  Search.  Legend: [*] built-in  [ ] excluded  <M> module  < > module capable
        ^(-)
             Binary Emulations  --->
        [*] Virtualization  ----
             General architecture-dependent options  --->
        [*] Enable loadable module support  --->
        -*- Enable the block layer  --->
             Executable file formats  --->
             Memory Management options  --->
        [ ] Networking support  ----
        v(+)

            <Select>    < Exit >    < Help >    < Save >    < Load >
```

Enable loadable module support in the kernel folder first

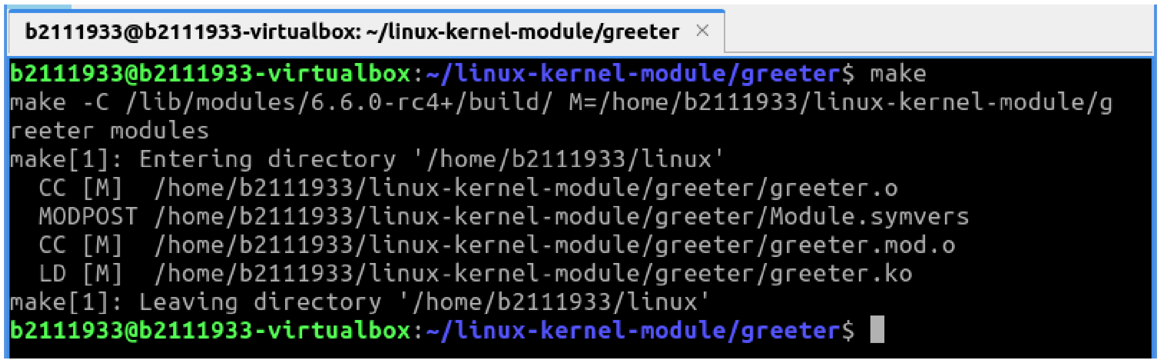
- Move into `greeter/` directory.



```
b2111933@b2111933-virtualbox:~$ ls
Autumn_Leaves  lab3-FOSS            newfile          safe_rm.sh
Desktop        linux                Pictures         snap
Documents      linux-kernel-module  Public           Templates
Downloads      Music                safe_rm_recycle  Videos
b2111933@b2111933-virtualbox:~$ cd linux-kernel-module
b2111933@b2111933-virtualbox:~/linux-kernel-module$ ls
babel  greeter  LICENSE  README.md
b2111933@b2111933-virtualbox:~/linux-kernel-module$ cd greeter
b2111933@b2111933-virtualbox:~/linux-kernel-module/greeter$
```

Move into **greeter/** directory

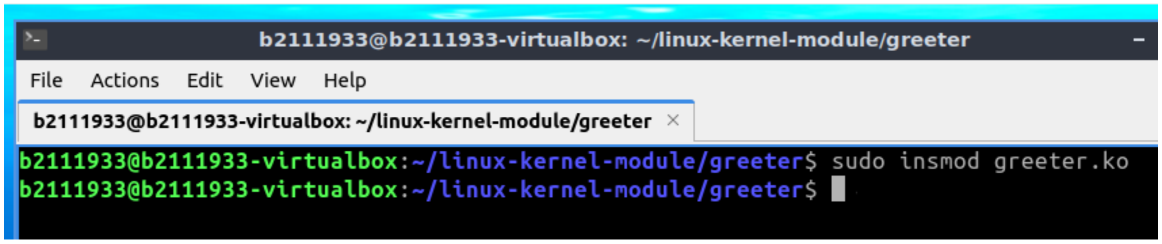
- Build the module using `make` command. The module is compiled to `greeter.ko`



```
b2111933@b2111933-virtualbox:~/linux-kernel-module/greeter$ make
make -C /lib/modules/6.6.0-rc4+/build/ M=/home/b2111933/linux-kernel-module/g
reeter modules
make[1]: Entering directory '/home/b2111933/linux'
  CC [M]  /home/b2111933/linux-kernel-module/greeter/greeter.o
  MODPOST /home/b2111933/linux-kernel-module/greeter/Module.symvers
  CC [M]  /home/b2111933/linux-kernel-module/greeter/greeter.mod.o
  LD [M]  /home/b2111933/linux-kernel-module/greeter/greeter.ko
make[1]: Leaving directory '/home/b2111933/linux'
b2111933@b2111933-virtualbox:~/linux-kernel-module/greeter$
```

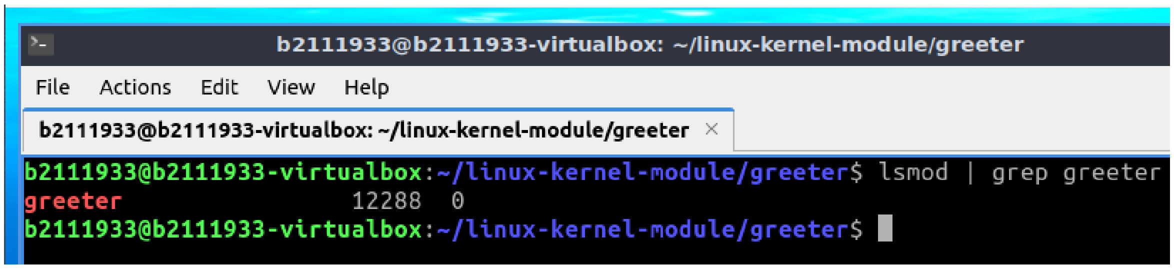
Build the module using **make** command. The module is compiled to **greeter.ko**

- Install the module using `insmod greeter.ko` command, then show that the module has been installed using `lsmod | grep greeter` command



```
b2111933@b2111933-virtualbox:~/linux-kernel-module/greeter$ sudo insmod greeter.ko
b2111933@b2111933-virtualbox:~/linux-kernel-module/greeter$
```

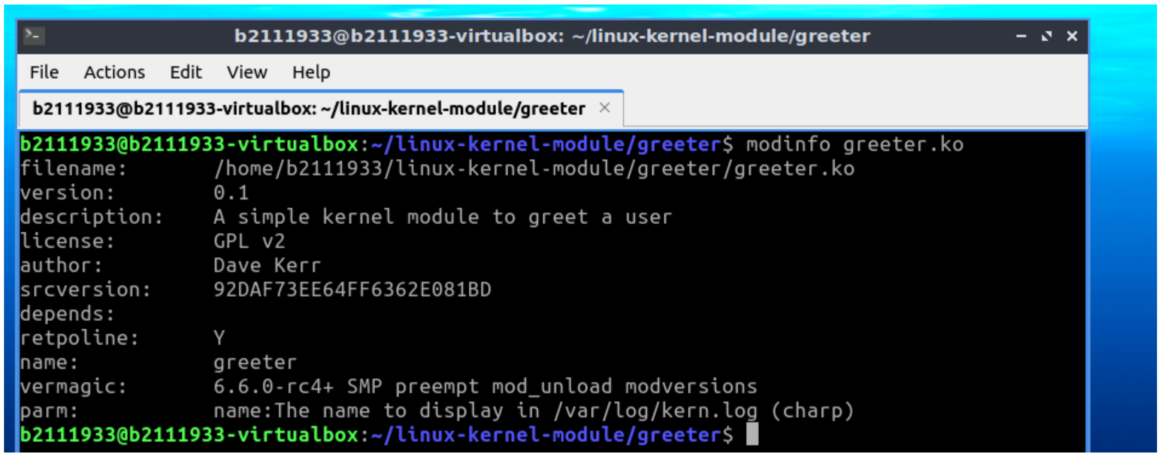
Install the module using **insmod greeter.ko** command


b2111933@b2111933-virtualbox: ~/linux-kernel-module/greeter
File Actions Edit View Help

```
b2111933@b2111933-virtualbox:~/linux-kernel-module/greeter$ lsmod | grep greeter
greeter                 12288  0
b2111933@b2111933-virtualbox:~/linux-kernel-module/greeter$ 
```

Show that the module has been installed using **lsmod | grep greeter** command

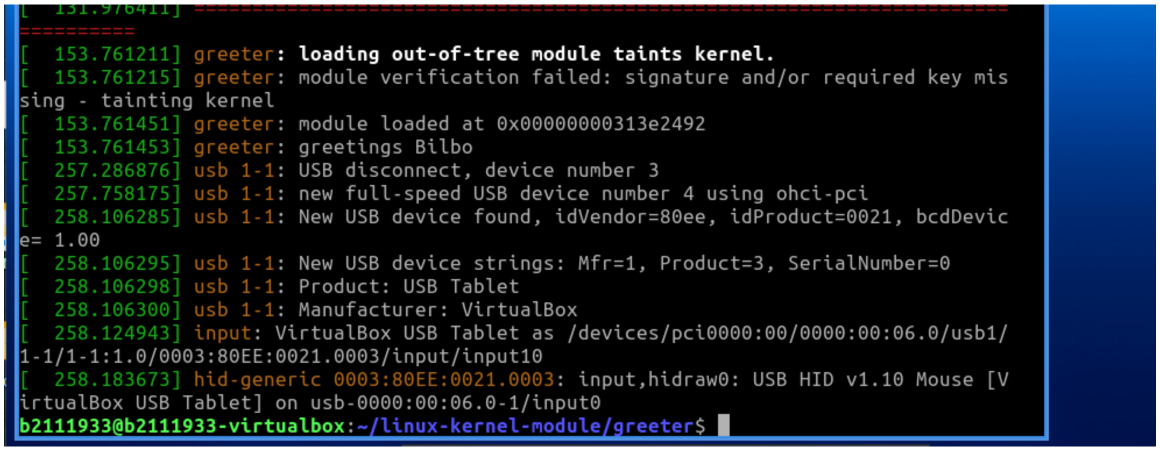
- Show the information of the module using `modinfo greeter.ko`


b2111933@b2111933-virtualbox: ~/linux-kernel-module/greeter
File Actions Edit View Help

```
b2111933@b2111933-virtualbox:~/linux-kernel-module/greeter$ modinfo greeter.ko
filename:       /home/b2111933/linux-kernel-module/greeter/greeter.ko
version:        0.1
description:    A simple kernel module to greet a user
license:        GPL v2
author:         Dave Kerr
srcversion:     92DAF73EE64FF6362E081BD
depends:        
retpoline:      Y
name:           greeter
vermagic:       6.6.0-rc4+ SMP preempt mod_unload modversions 
parm:           name:The name to display in /var/log/kern.log (charp)
b2111933@b2111933-virtualbox:~/linux-kernel-module/greeter$ 
```

Show the information of the module using **modinfo greeter.ko**

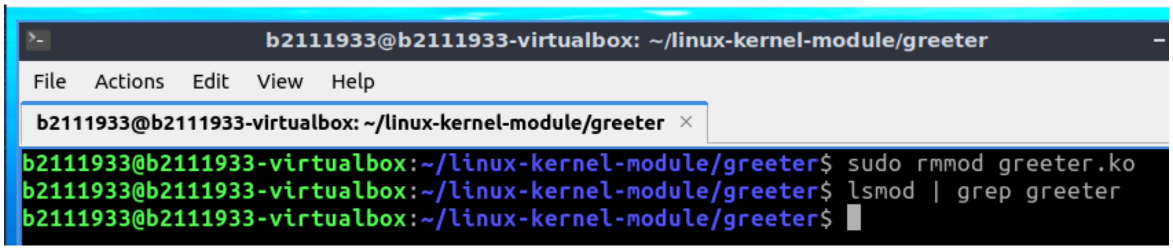
- Show kernel log with `dmesg`



```
[  131.976411] ====================================================================
==========
[  153.761211] greeter: loading out-of-tree module taints kernel.
[  153.761215] greeter: module verification failed: signature and/or required key mis
sing - tainting kernel
[  153.761451] greeter: module loaded at 0x00000000313e2492
[  153.761453] greeter: greetings Bilbo
[  257.286876] usb 1-1: USB disconnect, device number 3
[  257.758175] usb 1-1: new full-speed USB device number 4 using ohci-pci
[  258.106285] usb 1-1: New USB device found, idVendor=80ee, idProduct=0021, bcdDevic
e= 1.00
[  258.106295] usb 1-1: New USB device strings: Mfr=1, Product=3, SerialNumber=0
[  258.106298] usb 1-1: Product: USB Tablet
[  258.106300] usb 1-1: Manufacturer: VirtualBox
[  258.124943] input: VirtualBox USB Tablet as /devices/pci0000:00/0000:00:06.0/usb1/
1-1/1-1:1.0/0003:80EE:0021.0003/input/input10
[  258.183673] hid-generic 0003:80EE:0021.0003: input,hidraw0: USB HID v1.10 Mouse [V
irtualBox USB Tablet] on usb-0000:00:06.0-1/input0
b2111933@b2111933-virtualbox:~/linux-kernel-module/greeter$ 
```

Show kernel log with **dmesg**

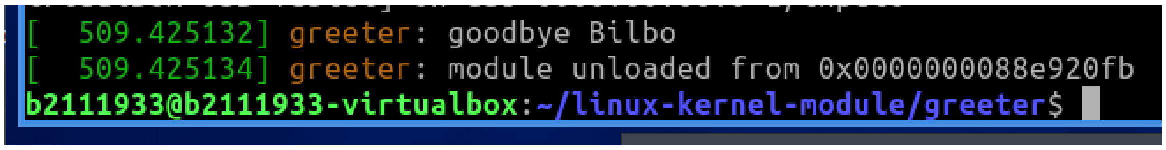
- Remove the module using `rmmod greeter.ko` command, then show that the module has been removed using `lsmod | grep greeter` command.


b2111933@b2111933-virtualbox: ~/linux-kernel-module/greeter
File Actions Edit View Help
b2111933@b2111933-virtualbox: ~/linux-kernel-module/greeter

```
b2111933@b2111933-virtualbox:~/linux-kernel-module/greeter$ sudo rmmod greeter.ko
b2111933@b2111933-virtualbox:~/linux-kernel-module/greeter$ lsmod | grep greeter
b2111933@b2111933-virtualbox:~/linux-kernel-module/greeter$ 
```

Remove the module and check

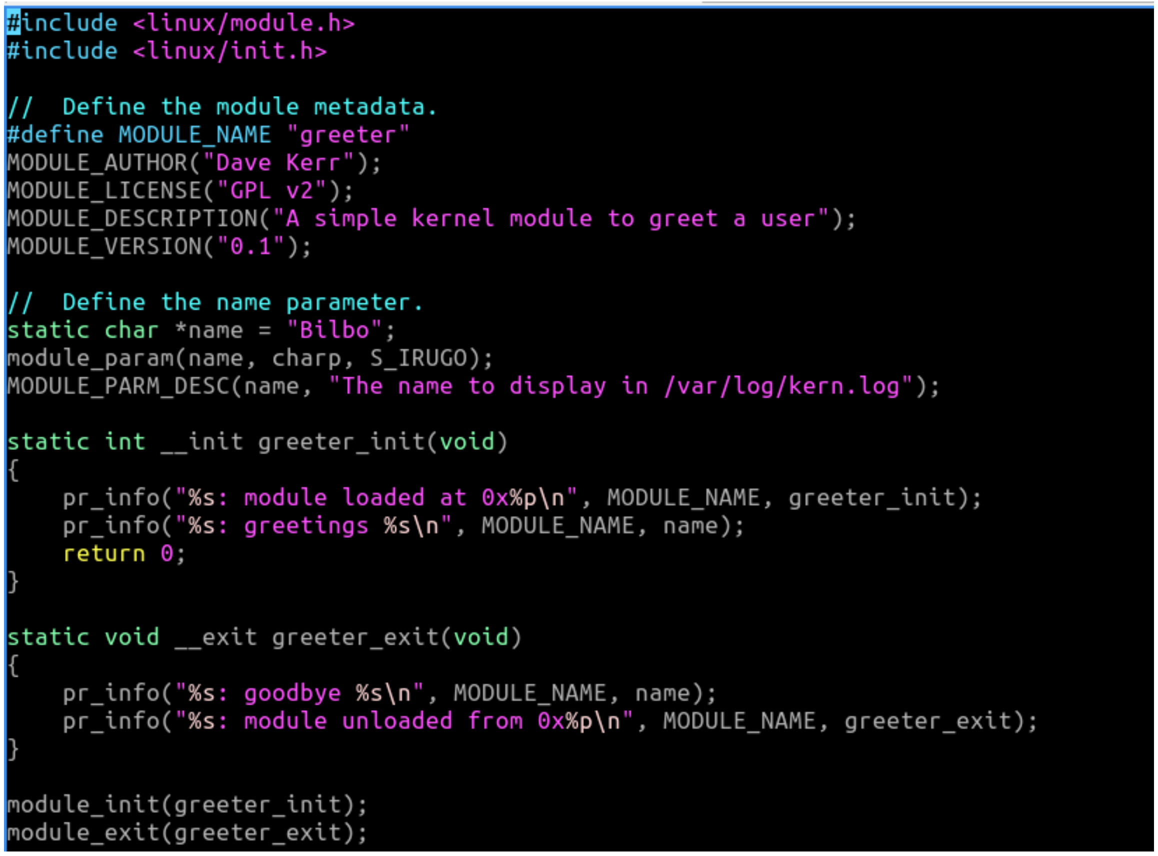
- Show kernel log with `dmesg`



```
[  509.425132] greeter: goodbye Bilbo
[  509.425134] greeter: module unloaded from 0x0000000088e920fb
b2111933@b2111933-virtualbox:~/linux-kernel-module/greeter$ 
```

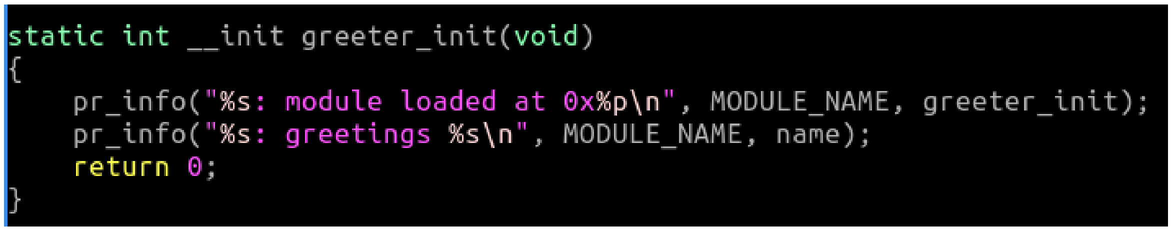
This is what we got

- Move to `greeter.c` file, then briefly explain below functions:

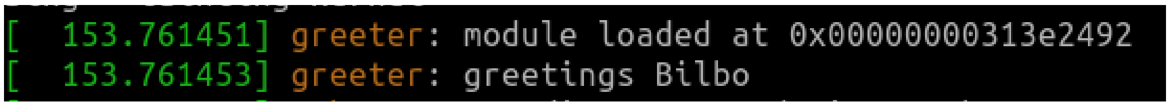


```
#include <linux/module.h>
#include <linux/init.h>

//  Define the module metadata.
#define MODULE_NAME "greeter"
MODULE_AUTHOR("Dave Kerr");
MODULE_LICENSE("GPL v2");
MODULE_DESCRIPTION("A simple kernel module to greet a user");
MODULE_VERSION("0.1");

//  Define the name parameter.
static char *name = "Bilbo";
module_param(name, charp, S_IRUGO);
MODULE_PARM_DESC(name, "The name to display in /var/log/kern.log");

static int __init greeter_init(void)
{
    pr_info("%s: module loaded at 0x%p\n", MODULE_NAME, greeter_init);
    pr_info("%s: greetings %s\n", MODULE_NAME, name);
    return 0;
}

static void __exit greeter_exit(void)
{
    pr_info("%s: goodbye %s\n", MODULE_NAME, name);
    pr_info("%s: module unloaded from 0x%p\n", MODULE_NAME, greeter_exit);
}

module_init(greeter_init);
module_exit(greeter_exit);
```

`greeter_init`



```
static int __init greeter_init(void)
{
    pr_info("%s: module loaded at 0x%p\n", MODULE_NAME, greeter_init);
    pr_info("%s: greetings %s\n", MODULE_NAME, name);
    return 0;
}
```

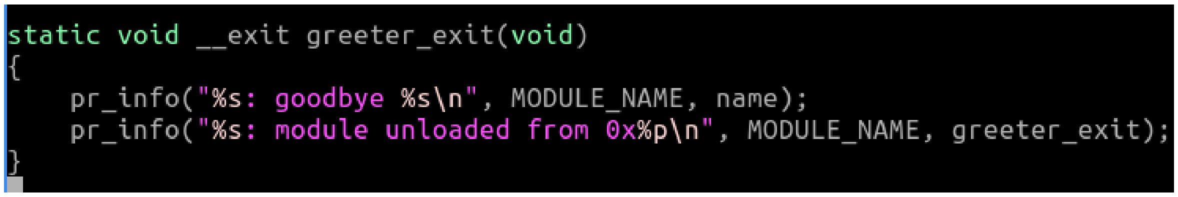


```
[  153.761451] greeter: module loaded at 0x00000000313e2492
[  153.761453] greeter: greetings Bilbo
```

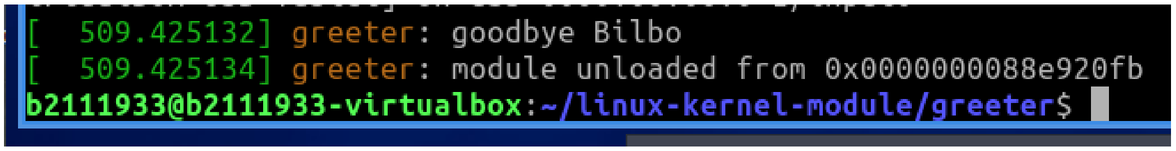
+ Output “<MODULE_NAME>: module loaded at 0x<greeter_init>”
    - The module name is: **greeter**
    - The address of greeter_init is: **00000000313e2492**

+ Output “<MODULE_NAME>: greetings <name>”
    - The module name is: **greeter**
    - We can define the name in the code above (default: **Bilbo**)

`greeter_exit`



```
static void __exit greeter_exit(void)
{
    pr_info("%s: goodbye %s\n", MODULE_NAME, name);
    pr_info("%s: module unloaded from 0x%p\n", MODULE_NAME, greeter_exit);
}
```

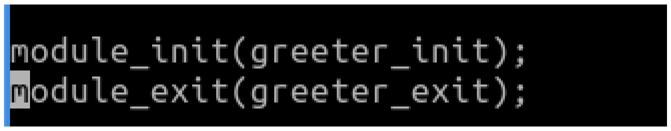


```
[  509.425132] greeter: goodbye Bilbo
[  509.425134] greeter: module unloaded from 0x0000000088e920fb
b2111933@b2111933-virtualbox:~/linux-kernel-module/greeter$
```

+ Output “<MODULE_NAME>: goodbye <name>”
    - The module name is: **greeter**
    - We can define the name in the code above (default: **Bilbo**)

+ Output “<MODULE_NAME>: module unloaded from 0x<greeter_exit>”
    - The module name is: **greeter**
    - The address of greeter_exit is: **0000000088e920fb**

```
module_init(greeter_init)
module_exit(greeter_exit)
```



```
module_init(greeter_init);
module_exit(greeter_exit);
```

- At the moment we install the module using **insmod greeter.ko** command, the kernel module calls the **greeter_init** command

- At the moment we remove the module using **rmmod greeter.ko** command, the kernel module calls the **greeter_exit** command

(take screenshots to show that you finish this exercise)

---END---