

## LAB 4

### LINUX KERNEL DEVELOPMENT



Fullname: Lam The Vinh

Student ID: B2206022

- 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
```

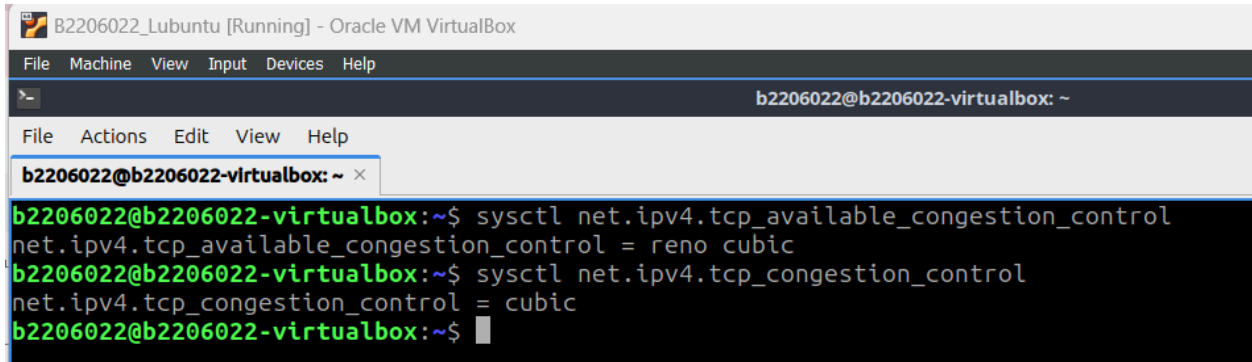
```
B2206022_Lubuntu [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
b2206022@b2206022-virtualbox: ~
File Actions Edit View Help
b2206022@b2206022-virtualbox: ~ x
b2206022@b2206022-virtualbox:~$ sysctl -a
abi.vsyscall32 = 1
debug.exception-trace = 1
debug.kprobes-optimization = 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
dev.cdrom.info = Can read MCN:        1
dev.cdrom.info = Reports media changed: 1
```

- List all available TCP congestion control algorithms:

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

- Show which TCP congestion control algorithm is using:

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



The screenshot shows a terminal window titled "B2206022\_Lubuntu [Running] - Oracle VM VirtualBox". The terminal prompt is "b2206022@b2206022-virtualbox: ~". The user enters the command "sysctl net.ipv4.tcp\_available\_congestion\_control", and the output is "net.ipv4.tcp\_available\_congestion\_control = reno cubic". Then, the user enters "sysctl net.ipv4.tcp\_congestion\_control", and the output is "net.ipv4.tcp\_congestion\_control = cubic".

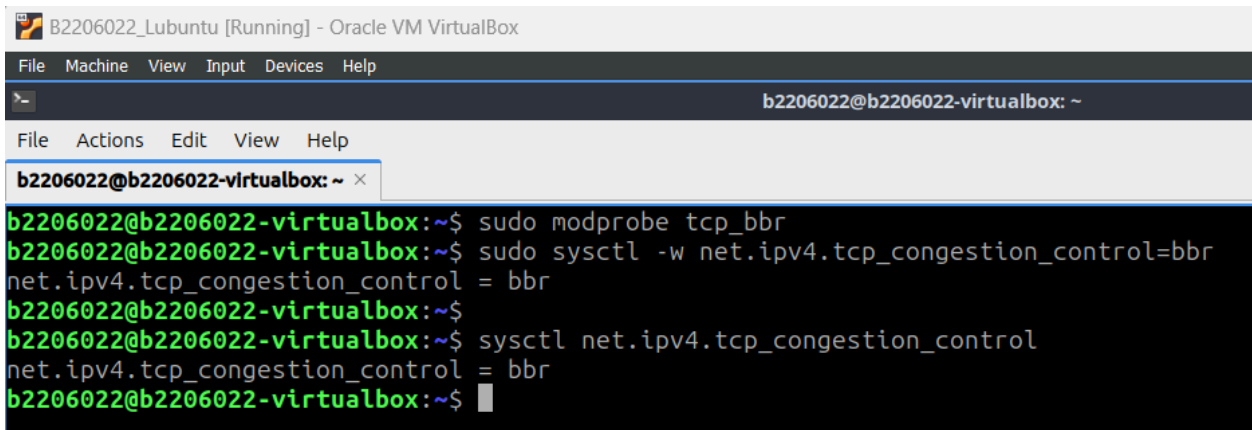
- Install bbr TCP congestion control algorithm module:

```
sudo modprobe tcp_bbr
```

- Switch to the bbr TCP congestion control algorithm:

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

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



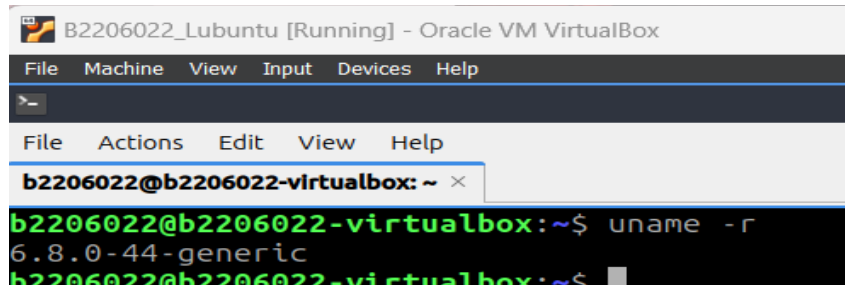
The screenshot shows a terminal window titled "B2206022\_Lubuntu [Running] - Oracle VM VirtualBox". The terminal prompt is "b2206022@b2206022-virtualbox: ~". The user enters "sudo modprobe tcp\_bbr". Then, the user enters "sudo sysctl -w net.ipv4.tcp\_congestion\_control=bbr", and the output is "net.ipv4.tcp\_congestion\_control = bbr". Finally, the user enters "sysctl net.ipv4.tcp\_congestion\_control", and the output is "net.ipv4.tcp\_congestion\_control = bbr".

(take screenshots to show that you finish this exercise)

## 2. Install new kernel version

- Show your current kernel version:

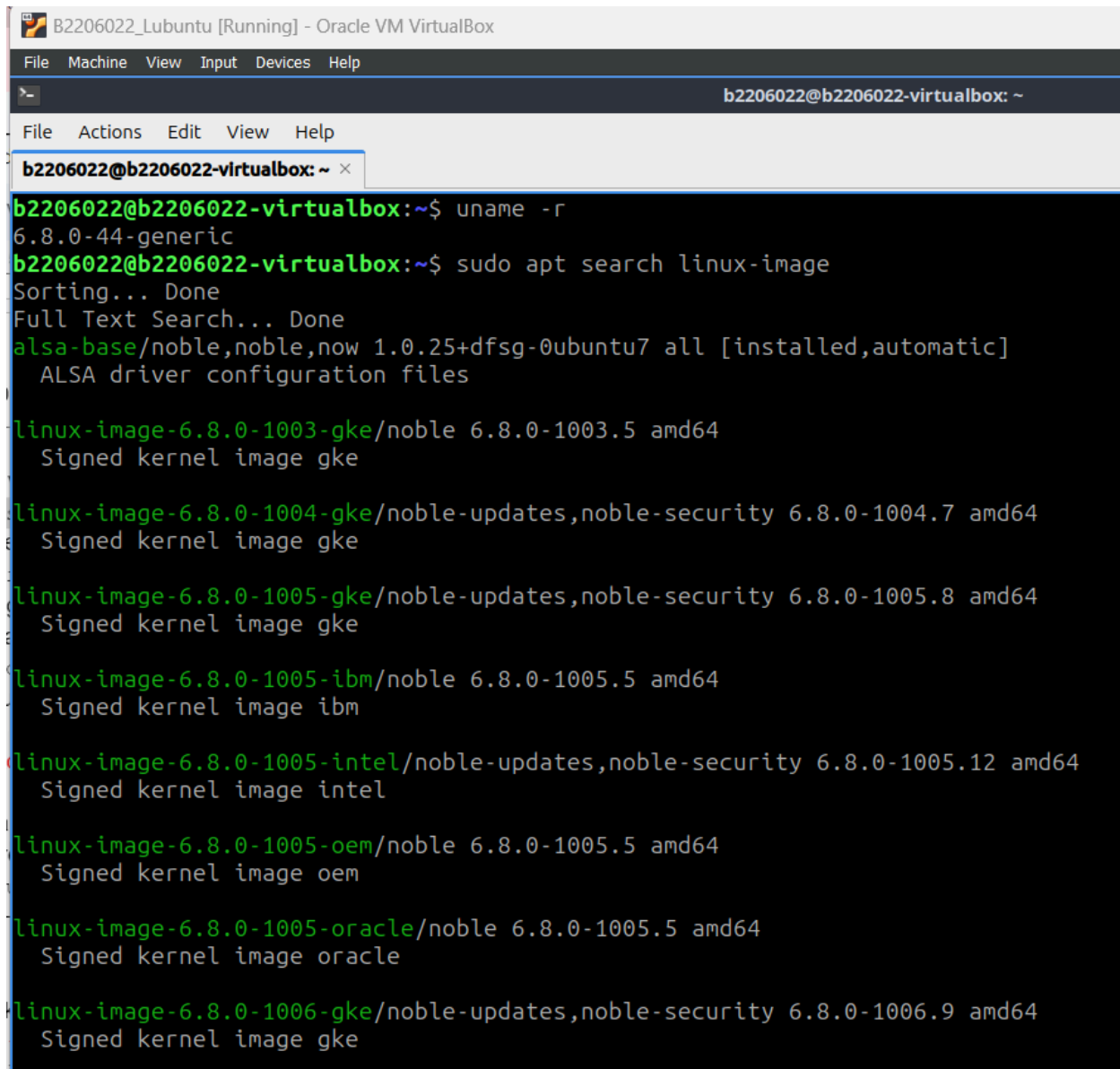
```
uname -r
```



```
B2206022_Lubuntu [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
b2206022@b2206022-virtualbox: ~ x
b2206022@b2206022-virtualbox:~$ uname -r
6.8.0-44-generic
b2206022@b2206022-virtualbox:~$
```

- Search for newer versions:

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



```
B2206022_Lubuntu [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
b2206022@b2206022-virtualbox: ~
File Actions Edit View Help
b2206022@b2206022-virtualbox: ~ x
b2206022@b2206022-virtualbox:~$ uname -r
6.8.0-44-generic
b2206022@b2206022-virtualbox:~$ sudo apt search linux-image
Sorting... Done
Full Text Search... Done
alsa-base/noble,noble,now 1.0.25+dfsg-0ubuntu7 all [installed,automatic]
  ALSA driver configuration files

linux-image-6.8.0-1003-gke/noble 6.8.0-1003.5 amd64
  Signed kernel image gke

linux-image-6.8.0-1004-gke/noble-updates,noble-security 6.8.0-1004.7 amd64
  Signed kernel image gke

linux-image-6.8.0-1005-gke/noble-updates,noble-security 6.8.0-1005.8 amd64
  Signed kernel image gke

linux-image-6.8.0-1005-ibm/noble 6.8.0-1005.5 amd64
  Signed kernel image ibm

linux-image-6.8.0-1005-intel/noble-updates,noble-security 6.8.0-1005.12 amd64
  Signed kernel image intel

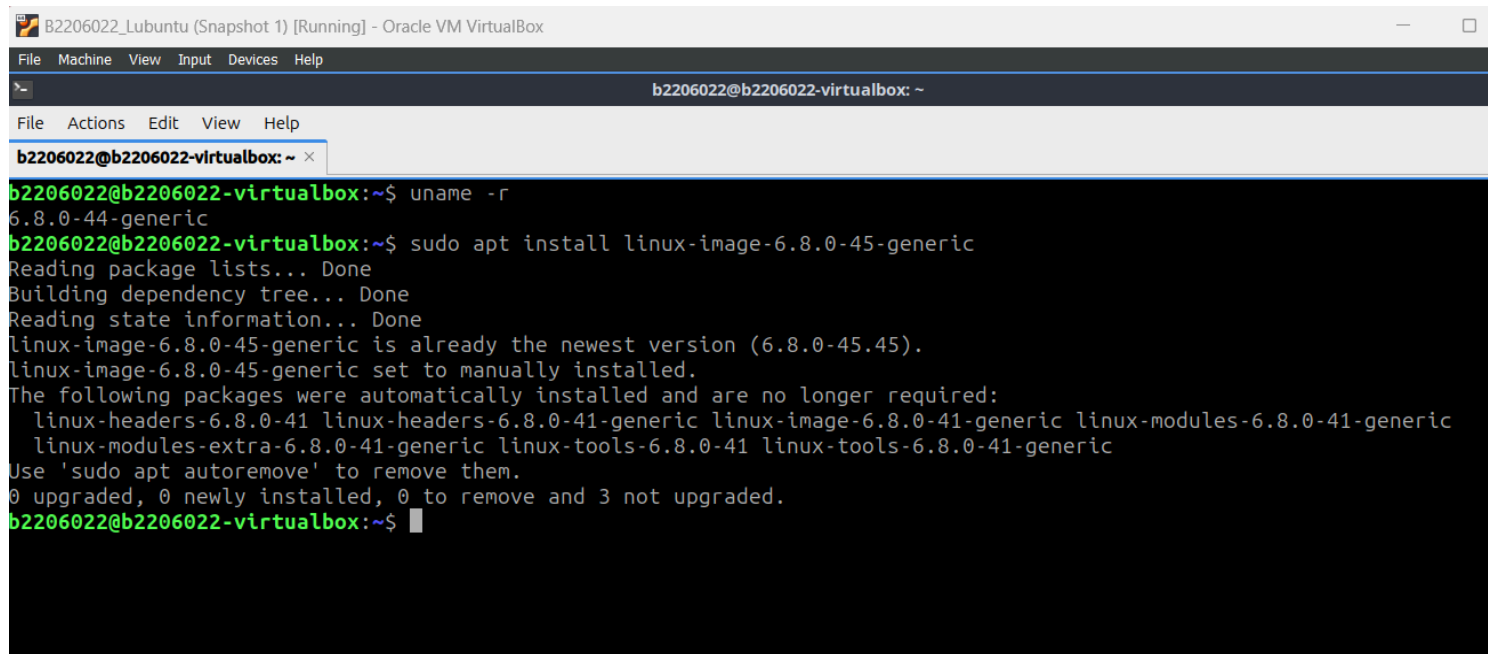
linux-image-6.8.0-1005-oem/noble 6.8.0-1005.5 amd64
  Signed kernel image oem

linux-image-6.8.0-1005-oracle/noble 6.8.0-1005.5 amd64
  Signed kernel image oracle

linux-image-6.8.0-1006-gke/noble-updates,noble-security 6.8.0-1006.9 amd64
  Signed kernel image gke
```

- Install the latest version you find:

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



The screenshot shows a terminal window titled "B2206022\_Lubuntu (Snapshot 1) [Running] - Oracle VM VirtualBox". The terminal output is as follows:

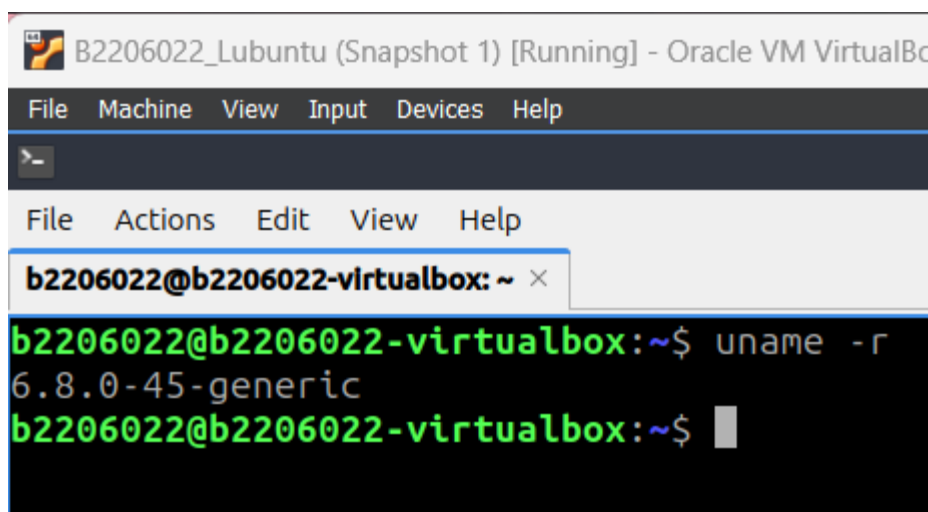
```
b2206022@b2206022-virtualbox: ~$ uname -r
6.8.0-44-generic
b2206022@b2206022-virtualbox:~$ sudo apt install linux-image-6.8.0-45-generic
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
linux-image-6.8.0-45-generic is already the newest version (6.8.0-45.45).
linux-image-6.8.0-45-generic set to manually installed.
The following packages were automatically installed and are no longer required:
  linux-headers-6.8.0-41 linux-headers-6.8.0-41-generic linux-image-6.8.0-41-generic linux-modules-6.8.0-41-generic
  linux-modules-extra-6.8.0-41-generic linux-tools-6.8.0-41 linux-tools-6.8.0-41-generic
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 3 not upgraded.
b2206022@b2206022-virtualbox:~$
```

- 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
```

- Show your new current kernel version:

```
uname -r
```



The screenshot shows a terminal window titled "B2206022\_Lubuntu (Snapshot 1) [Running] - Oracle VM VirtualBox". The terminal output is as follows:

```
b2206022@b2206022-virtualbox: ~$ uname -r
6.8.0-45-generic
b2206022@b2206022-virtualbox:~$
```

(take screenshots to show that you finish this exercise)

### 3. Build and install a new kernel version

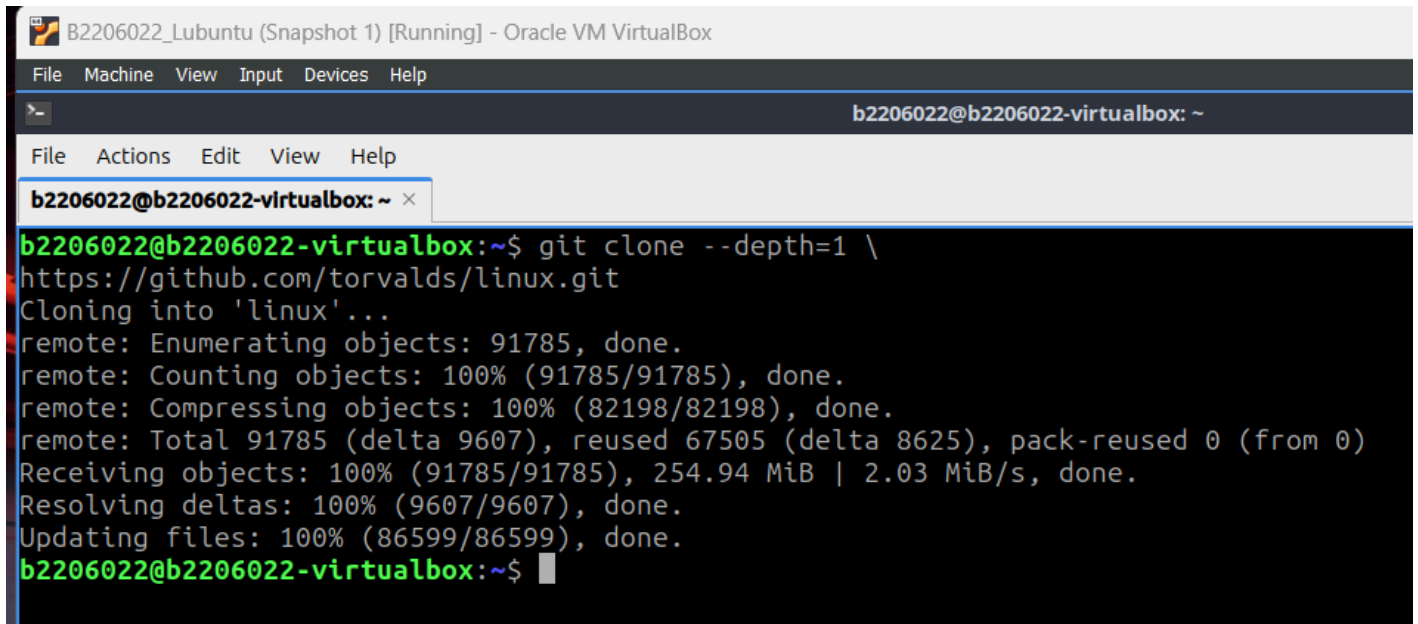
- Get your system ready

```
sudo apt update
```

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

- Clone a mainline kernel source code to your computer:

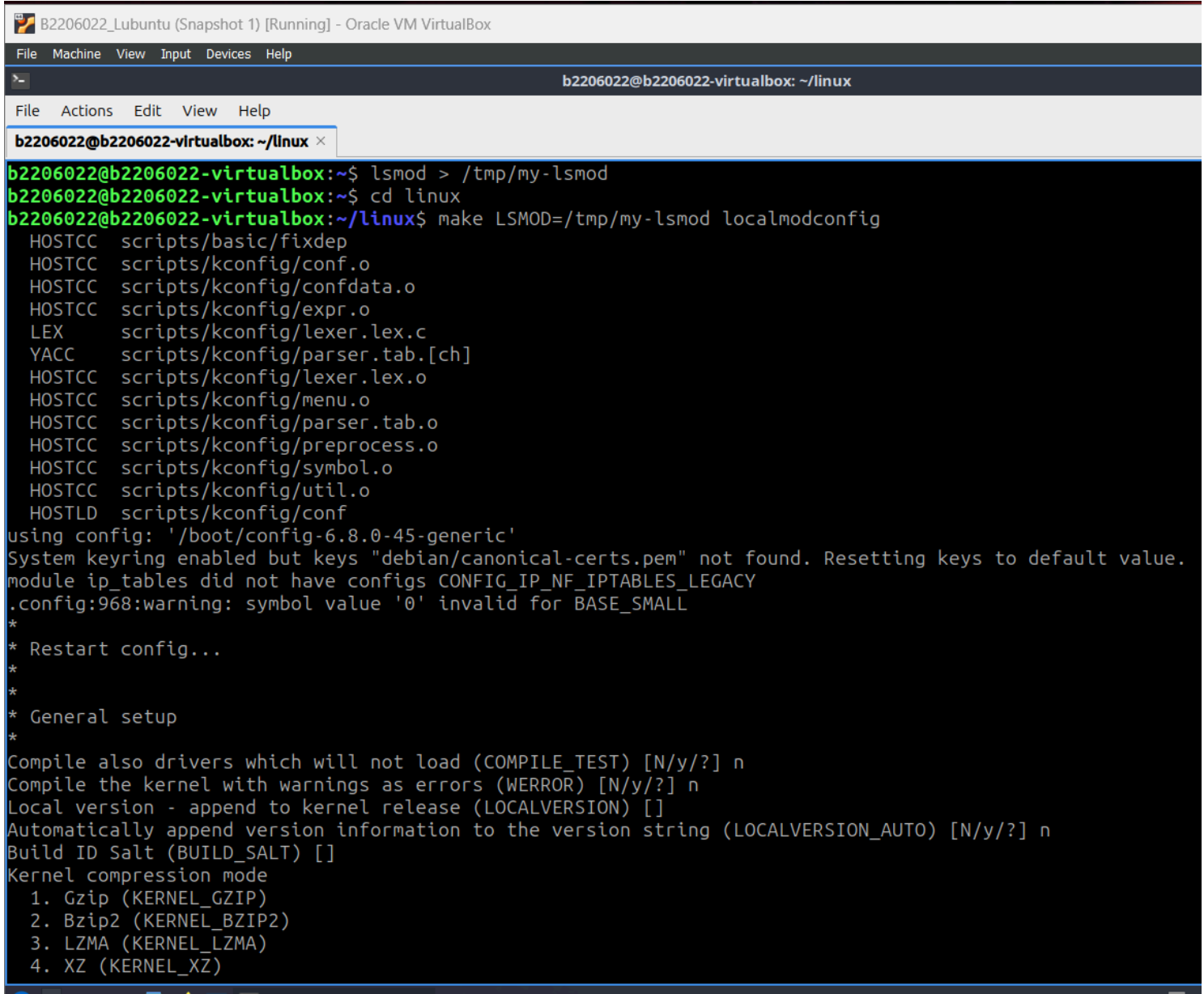
```
git clone --depth=1 \  
https://github.com/torvalds/linux.git
```



The screenshot shows a terminal window titled "B2206022\_Lubuntu (Snapshot 1) [Running] - Oracle VM VirtualBox". The terminal has a menu bar with "File", "Machine", "View", "Input", "Devices", and "Help". Below the menu bar, the prompt is "b2206022@b2206022-virtualbox: ~". The terminal content shows the execution of the command `git clone --depth=1 https://github.com/torvalds/linux.git`. The output indicates that the repository is being cloned into a directory named 'linux'. The progress shows 91785 objects enumerated, 100% counted, 100% compressed, and 254.94 MiB received at 2.03 MiB/s. The process concludes with "Updating files: 100% (86599/86599), done."

- 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
```

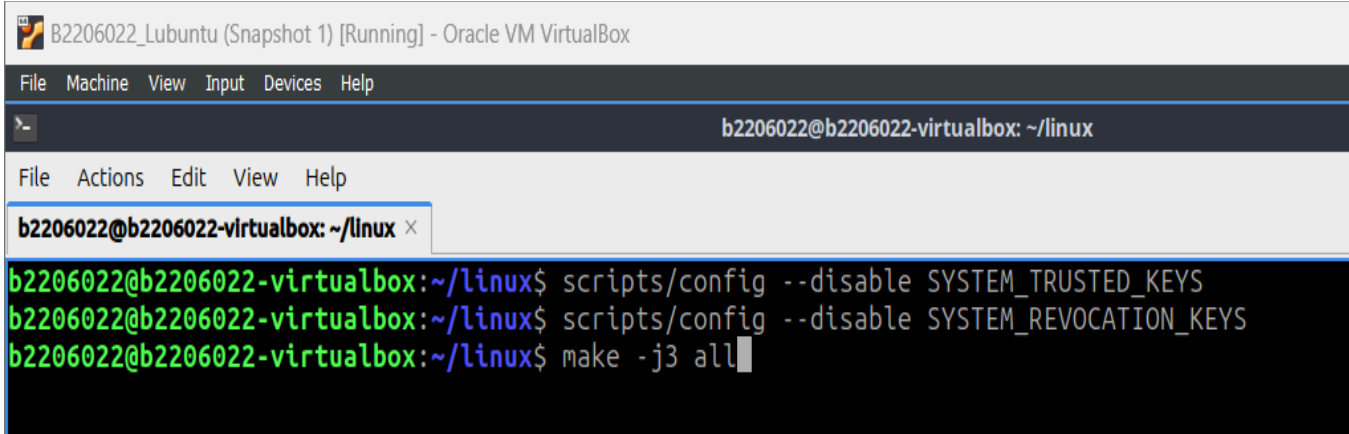


```
B2206022_Lubuntu (Snapshot 1) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
b2206022@b2206022-virtualbox: ~/linux
File Actions Edit View Help
b2206022@b2206022-virtualbox: ~/linux x
b2206022@b2206022-virtualbox:~$ lsmod > /tmp/my-lsmod
b2206022@b2206022-virtualbox:~$ cd linux
b2206022@b2206022-virtualbox:~/linux$ make LSMOD=/tmp/my-lsmod localmodconfig
HOSTCC scripts/basic/fixdep
HOSTCC scripts/kconfig/conf.o
HOSTCC scripts/kconfig/confdata.o
HOSTCC scripts/kconfig/expr.o
LEX scripts/kconfig/lexer.lex.c
YACC scripts/kconfig/parser.tab.[ch]
HOSTCC scripts/kconfig/lexer.lex.o
HOSTCC scripts/kconfig/menu.o
HOSTCC scripts/kconfig/parser.tab.o
HOSTCC scripts/kconfig/preprocess.o
HOSTCC scripts/kconfig/symbol.o
HOSTCC scripts/kconfig/util.o
HOSTLD scripts/kconfig/conf
using config: '/boot/config-6.8.0-45-generic'
System keyring enabled but keys "debian/canonical-certs.pem" not found. Resetting keys to default value.
module ip_tables did not have configs CONFIG_IP_NF_IPTABLES_LEGACY
.config:968:warning: symbol value '0' invalid for BASE_SMALL
*
* Restart config...
*
*
* General setup
*
Compile also drivers which will not load (COMPILE_TEST) [N/y/?] n
Compile the kernel with warnings as errors (WERROR) [N/y/?] n
Local version - append to kernel release (LOCALVERSION) []
Automatically append version information to the version string (LOCALVERSION_AUTO) [N/y/?] n
Build ID Salt (BUILD_SALT) []
Kernel compression mode
 1. Gzip (KERNEL_GZIP)
 2. Bzip2 (KERNEL_BZIP2)
 3. LZMA (KERNEL_LZMA)
 4. XZ (KERNEL_XZ)
```

- Disable certificate stuff:

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

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



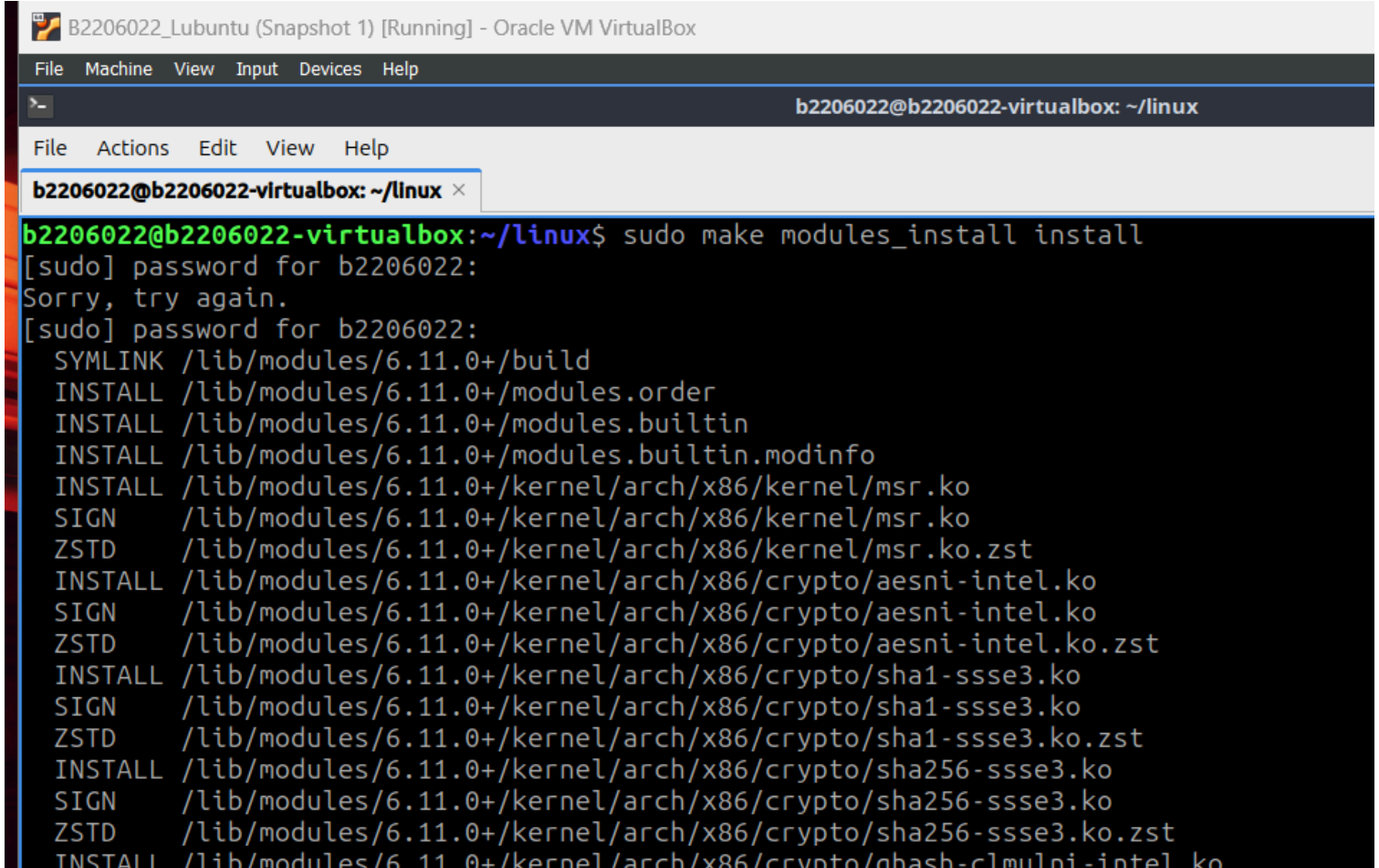
```
B2206022_Lubuntu (Snapshot 1) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
b2206022@b2206022-virtualbox: ~/linux
File Actions Edit View Help
b2206022@b2206022-virtualbox: ~/linux x
b2206022@b2206022-virtualbox:~/linux$ scripts/config --disable SYSTEM_TRUSTED_KEYS
b2206022@b2206022-virtualbox:~/linux$ scripts/config --disable SYSTEM_REVOCATION_KEYS
b2206022@b2206022-virtualbox:~/linux$ make -j3 all
```

- 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
```

- Install the new kernel:

```
sudo make modules_install install
```



The screenshot shows a terminal window titled "B2206022\_Lubuntu (Snapshot 1) [Running] - Oracle VM VirtualBox". The terminal prompt is "b2206022@b2206022-virtualbox: ~/linux". The user has entered the command "sudo make modules\_install install". The terminal output shows the installation of various kernel modules, including "msr.ko", "aesni-intel.ko", "sha1-ssse3.ko", and "sha256-ssse3.ko". The output is as follows:

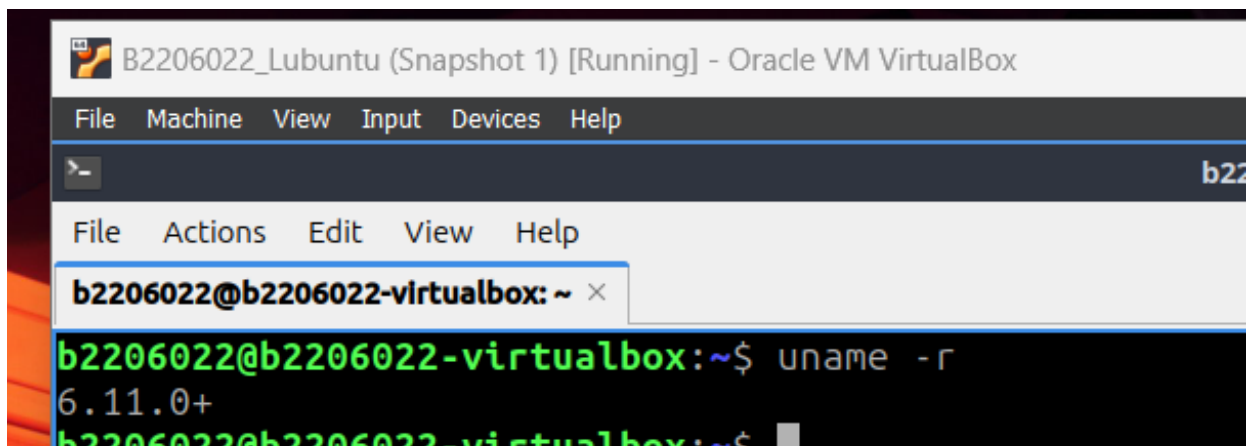
```
b2206022@b2206022-virtualbox:~/linux$ sudo make modules_install install
[sudo] password for b2206022:
Sorry, try again.
[sudo] password for b2206022:
SYMLINK /lib/modules/6.11.0+/build
INSTALL /lib/modules/6.11.0+/modules.order
INSTALL /lib/modules/6.11.0+/modules.builtin
INSTALL /lib/modules/6.11.0+/modules.builtin.modinfo
INSTALL /lib/modules/6.11.0+/kernel/arch/x86/kernel/msr.ko
SIGN /lib/modules/6.11.0+/kernel/arch/x86/kernel/msr.ko
ZSTD /lib/modules/6.11.0+/kernel/arch/x86/kernel/msr.ko.zst
INSTALL /lib/modules/6.11.0+/kernel/arch/x86/crypto/aesni-intel.ko
SIGN /lib/modules/6.11.0+/kernel/arch/x86/crypto/aesni-intel.ko
ZSTD /lib/modules/6.11.0+/kernel/arch/x86/crypto/aesni-intel.ko.zst
INSTALL /lib/modules/6.11.0+/kernel/arch/x86/crypto/sha1-ssse3.ko
SIGN /lib/modules/6.11.0+/kernel/arch/x86/crypto/sha1-ssse3.ko
ZSTD /lib/modules/6.11.0+/kernel/arch/x86/crypto/sha1-ssse3.ko.zst
INSTALL /lib/modules/6.11.0+/kernel/arch/x86/crypto/sha256-ssse3.ko
SIGN /lib/modules/6.11.0+/kernel/arch/x86/crypto/sha256-ssse3.ko
ZSTD /lib/modules/6.11.0+/kernel/arch/x86/crypto/sha256-ssse3.ko.zst
INSTALL /lib/modules/6.11.0+/kernel/arch/x86/crypto/ghash-clmulni-intel.ko
```

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

```
sudo shutdown -r now
```

- Show your new current kernel version:

```
uname -r
```



The screenshot shows a terminal window titled "B2206022\_Lubuntu (Snapshot 1) [Running] - Oracle VM VirtualBox". The terminal prompt is "b2206022@b2206022-virtualbox: ~". The user has entered the command "uname -r". The terminal output is "6.11.0+", indicating the new kernel version.

```
b2206022@b2206022-virtualbox:~$ uname -r
6.11.0+
```

(take screenshots to show that you finish this exercise)



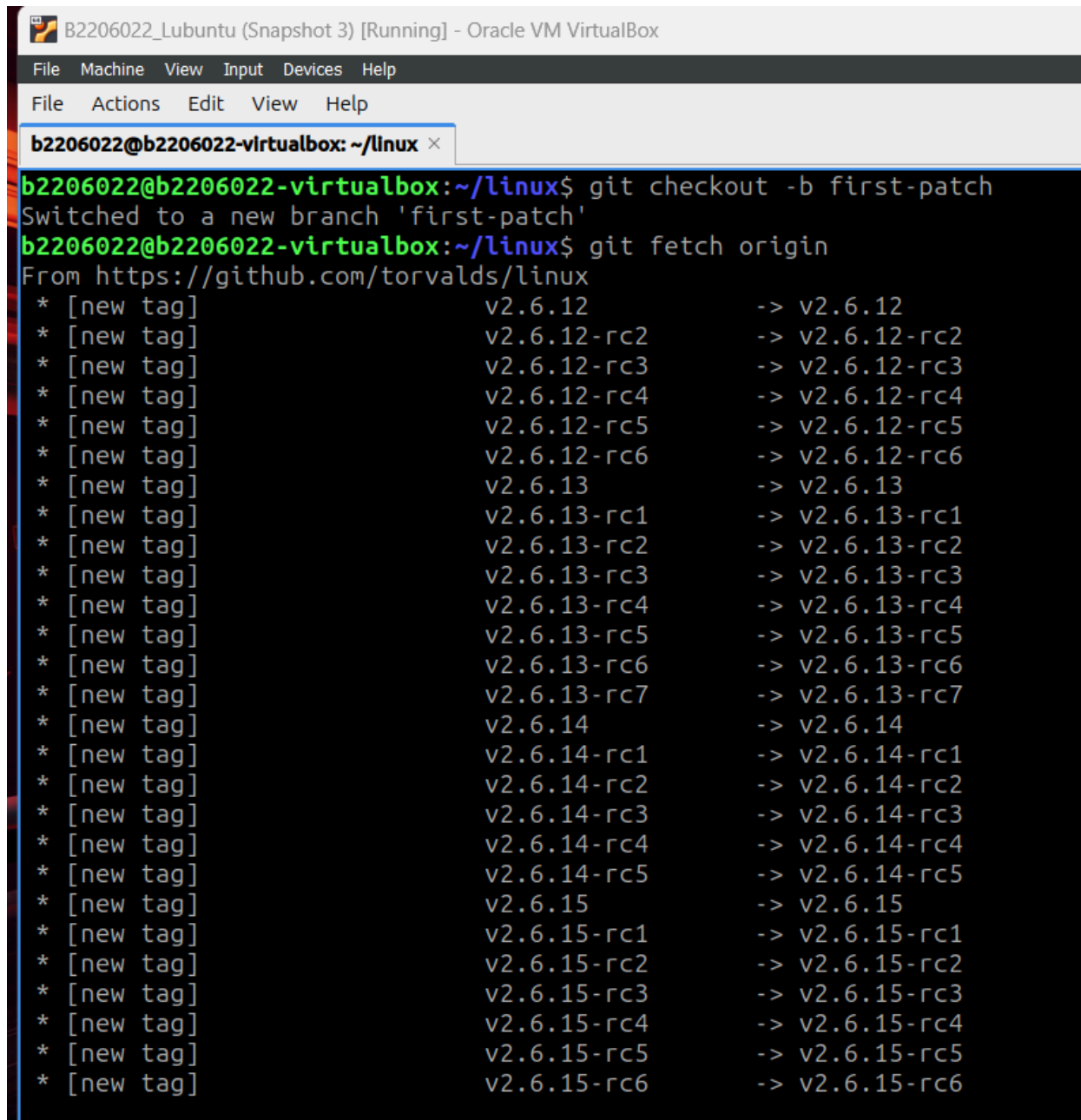
#### 4. Writing Your First Kernel Patch

- Creating a new branch in the linux\_mainline repository (has been cloned in exercise 3)

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

- Update the kernel

```
git fetch origin
```



The screenshot shows a terminal window titled "B2206022\_Lubuntu (Snapshot 3) [Running] - Oracle VM VirtualBox". The terminal has a menu bar with "File", "Machine", "View", "Input", "Devices", and "Help". Below the menu bar is another menu bar with "File", "Actions", "Edit", "View", and "Help". The terminal prompt is "b2206022@b2206022-virtualbox: ~/linux". The user enters the command "git checkout -b first-patch", and the output is "Switched to a new branch 'first-patch'". The user then enters the command "git fetch origin", and the output is "From https://github.com/torvalds/linux". This is followed by a list of new tags and their corresponding versions, such as "v2.6.12", "v2.6.12-rc2", "v2.6.12-rc3", "v2.6.12-rc4", "v2.6.12-rc5", "v2.6.12-rc6", "v2.6.13", "v2.6.13-rc1", "v2.6.13-rc2", "v2.6.13-rc3", "v2.6.13-rc4", "v2.6.13-rc5", "v2.6.13-rc6", "v2.6.13-rc7", "v2.6.14", "v2.6.14-rc1", "v2.6.14-rc2", "v2.6.14-rc3", "v2.6.14-rc4", "v2.6.14-rc5", "v2.6.15", "v2.6.15-rc1", "v2.6.15-rc2", "v2.6.15-rc3", "v2.6.15-rc4", "v2.6.15-rc5", and "v2.6.15-rc6".

```
b2206022@b2206022-virtualbox: ~/linux$ git checkout -b first-patch
Switched to a new branch 'first-patch'
b2206022@b2206022-virtualbox: ~/linux$ git fetch origin
From https://github.com/torvalds/linux
* [new tag]                v2.6.12                -> v2.6.12
* [new tag]                v2.6.12-rc2             -> v2.6.12-rc2
* [new tag]                v2.6.12-rc3             -> v2.6.12-rc3
* [new tag]                v2.6.12-rc4             -> v2.6.12-rc4
* [new tag]                v2.6.12-rc5             -> v2.6.12-rc5
* [new tag]                v2.6.12-rc6             -> v2.6.12-rc6
* [new tag]                v2.6.13                -> v2.6.13
* [new tag]                v2.6.13-rc1             -> v2.6.13-rc1
* [new tag]                v2.6.13-rc2             -> v2.6.13-rc2
* [new tag]                v2.6.13-rc3             -> v2.6.13-rc3
* [new tag]                v2.6.13-rc4             -> v2.6.13-rc4
* [new tag]                v2.6.13-rc5             -> v2.6.13-rc5
* [new tag]                v2.6.13-rc6             -> v2.6.13-rc6
* [new tag]                v2.6.13-rc7             -> v2.6.13-rc7
* [new tag]                v2.6.14                -> v2.6.14
* [new tag]                v2.6.14-rc1             -> v2.6.14-rc1
* [new tag]                v2.6.14-rc2             -> v2.6.14-rc2
* [new tag]                v2.6.14-rc3             -> v2.6.14-rc3
* [new tag]                v2.6.14-rc4             -> v2.6.14-rc4
* [new tag]                v2.6.14-rc5             -> v2.6.14-rc5
* [new tag]                v2.6.15                -> v2.6.15
* [new tag]                v2.6.15-rc1             -> v2.6.15-rc1
* [new tag]                v2.6.15-rc2             -> v2.6.15-rc2
* [new tag]                v2.6.15-rc3             -> v2.6.15-rc3
* [new tag]                v2.6.15-rc4             -> v2.6.15-rc4
* [new tag]                v2.6.15-rc5             -> v2.6.15-rc5
* [new tag]                v2.6.15-rc6             -> v2.6.15-rc6
```

- 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.

- Run `git grep` to look for e1000 files

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

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

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

```
# 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;
```

```
...
```

- Compile and install your changes:

```
make -j3
```

```
sudo make modules_install install
```

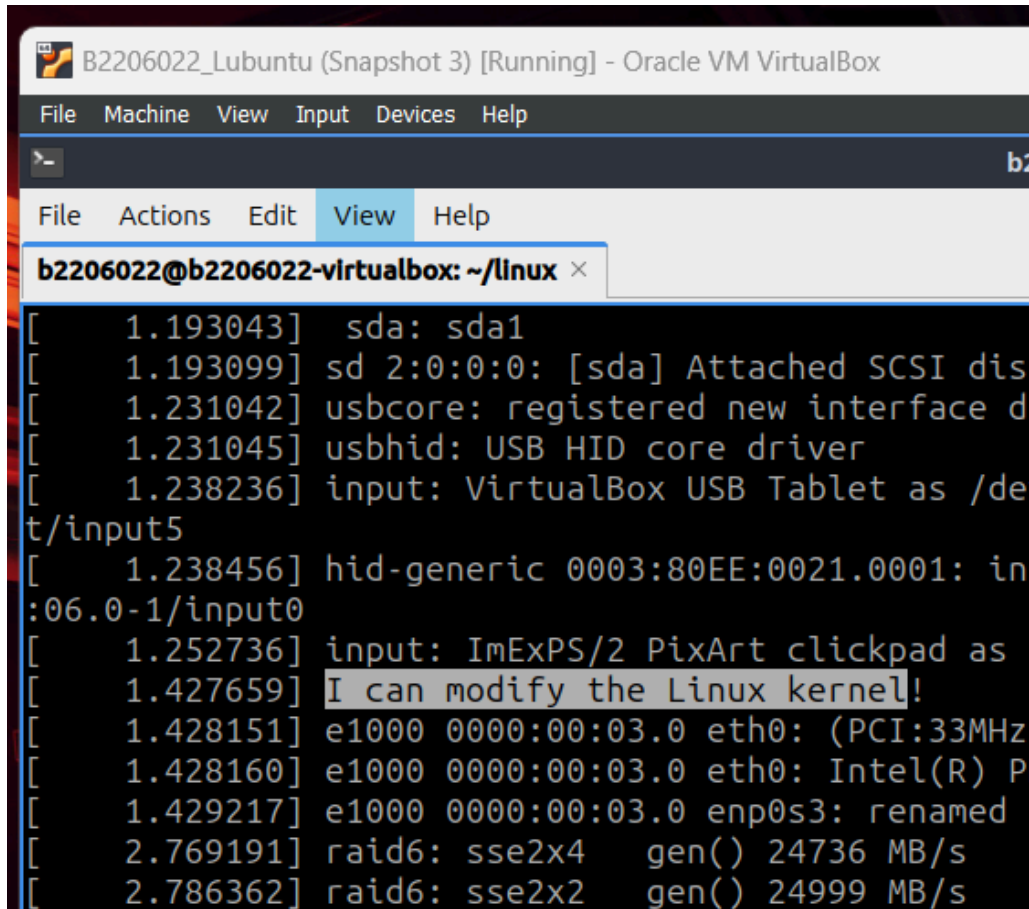
- Reboot the system:

```
sudo shutdown -r now
```

- Show kernel buffer log:

```
dmesg | less
```

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



The screenshot shows a VirtualBox window titled "B2206022\_Lubuntu (Snapshot 3) [Running] - Oracle VM VirtualBox". Inside the window is a terminal window titled "b2206022@b2206022-virtualbox: ~/linux". The terminal output shows the following lines:

```
[ 1.193043] sda: sda1
[ 1.193099] sd 2:0:0:0: [sda] Attached SCSI disk
[ 1.231042] usbcore: registered new interface driver
[ 1.231045] usbhid: USB HID core driver
[ 1.238236] input: VirtualBox USB Tablet as /dev/input5
[ 1.238456] hid-generic 0003:80EE:0021.0001: input06.0-1/input0
[ 1.252736] input: ImExPS/2 PixArt clickpad as /dev/input1
[ 1.427659] I can modify the Linux kernel!
[ 1.428151] e1000 0000:00:03.0 eth0: (PCI:33MHz) 00:00:00:00:00:00
[ 1.428160] e1000 0000:00:03.0 eth0: Intel(R) PRO/1000 MT Desktop
[ 1.429217] e1000 0000:00:03.0 enp0s3: renamed from eth0
[ 2.769191] raid6: sse2x4   gen() 24736 MB/s
[ 2.786362] raid6: sse2x2   gen() 24999 MB/s
```

- Committing changes, and view your commit

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

```
B2206022_Lubuntu (Snapshot 3) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
b2206022@b2206022-virtualbox: ~/linux
File Actions Edit View Help
b2206022@b2206022-virtualbox: ~/linux x
b2206022@b2206022-virtualbox:~/linux$ git add .
b2206022@b2206022-virtualbox:~/linux$ git commit -s -v -m "My first kernel patch"
[first-patch 36b88919c096] My first kernel patch
1 file changed, 1 insertion(+), 1 deletion(-)
b2206022@b2206022-virtualbox:~/linux$ git show HEAD
commit 36b88919c096cea7d1e6a8f8526d917355a01bdc (HEAD -> first-patch)
Author: Vinh_Lam_the <vinhb2206022@student.ctu.edu.vn>
Date: Fri Sep 27 18:57:46 2024 +0700

    My first kernel patch

    Signed-off-by: Vinh_Lam_the <vinhb2206022@student.ctu.edu.vn>

diff --git a/drivers/net/ethernet/intel/e1000/e1000_main.c b/drivers/net/ethernet/intel/e1000/e1000_main.c
index ab7ae418d294..bc5d4c3a52a6 100644
--- a/drivers/net/ethernet/intel/e1000/e1000_main.c
+++ b/drivers/net/ethernet/intel/e1000/e1000_main.c
@@ -301,7 +301,7 @@ static void e1000_update_mng_vlan(struct e1000_adapter *adapter)
     struct net_device *netdev = adapter->netdev;
     u16 vid = hw->mng_cookie.vlan_id;
     u16 old_vid = adapter->mng_vlan_id;

+    printk(KERN_DEBUG "I can modify the Linux kernel!\n");
     if (!e1000_vlan_used(adapter))
         return;
```


- Find whom to send the patch to

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

```
B2206022_Lubuntu (Snapshot 3) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
b2206022@b2206022-virtualbox: ~/linux
File Actions Edit View Help
b2206022@b2206022-virtualbox: ~/linux x
b2206022@b2206022-virtualbox:~/linux$ git show HEAD | scripts/get_maintainer.pl
Tony Nguyen <anthony.l.nguyen@intel.com> (supporter:INTEL ETHERNET DRIVERS)
Przemek Kitszel <przemyslaw.kitszel@intel.com> (supporter:INTEL ETHERNET DRIVERS)
"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 DRIVERS)
netdev@vger.kernel.org (open list:NETWORKING DRIVERS)
linux-kernel@vger.kernel.org (open list)
b2206022@b2206022-virtualbox:~/linux$
```

- 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.



The screenshot shows a terminal window titled "B2206022\_Lubuntu (Snapshot 3) [Running] - Oracle VM VirtualBox". The terminal has a menu bar with "File", "Machine", "View", "Input", "Devices", and "Help". Below the menu bar, the terminal address bar shows "b2206022@b2206022-virtualbox: ~/linux". The terminal window has a menu bar with "File", "Actions", "Edit", "View", and "Help". The terminal content shows the command `git format-patch -1 36b88919c096cea7d1e6a8f8526d917355a01bdc --to=lamthevinh71vt@gmail.com` being executed, which results in the file `0001-My-first-kernel-patch.patch` being created. The prompt then returns to `b2206022@b2206022-virtualbox:~/linux$`.

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

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

- Send the patch

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

## [PATCH] My first kernel patch Hộp thư đến x



**Vinh\_Lam\_the** <vinhb2206022@student.ctu.edu.vn>

đến lamthevinh71vt, tôi ▾

Signed-off-by: Vinh\_Lam\_the <[vinhb2206022@student.ctu.edu.vn](mailto:vinhb2206022@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 ab7ae418d294..bc5d4c3a52a6 100644

--- a/drivers/net/ethernet/intel/e1000/e1000\_main.c

+++ b/drivers/net/ethernet/intel/e1000/e1000\_main.c

@@ -301,7 +301,7 @@ static void e1000\_update\_mng\_vlan(struct e1000\_adapter \*adapter)

struct net\_device \*netdev = adapter->netdev;

u16 vid = hw->mng\_cookie.vlan\_id;

u16 old\_vid = adapter->mng\_vlan\_id;

-

+ printk(KERN\_DEBUG "I can modify the Linux kernel!\n");

if (!e1000\_vlan\_used(adapter))

return;

--

2.43.0

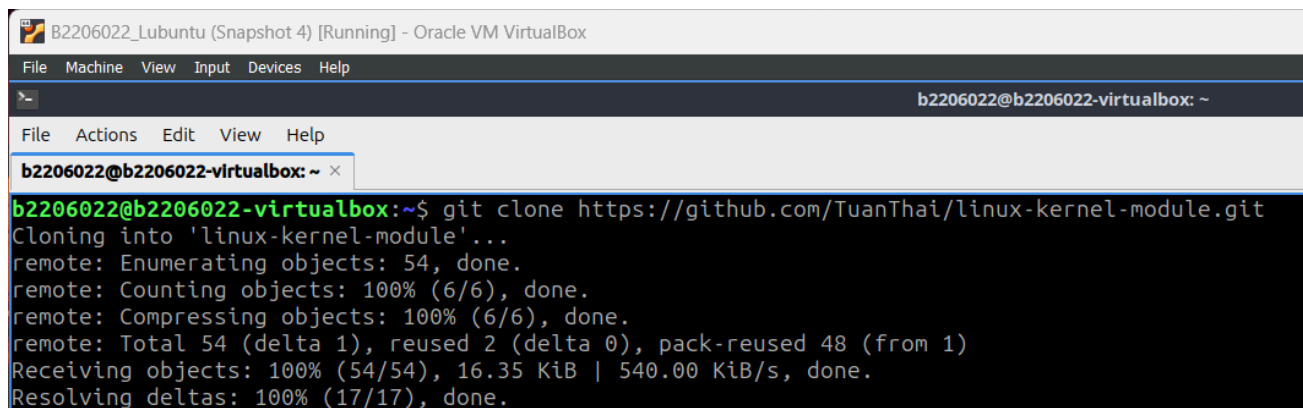
(take screenshots to show that you finish this exercise)

### 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):

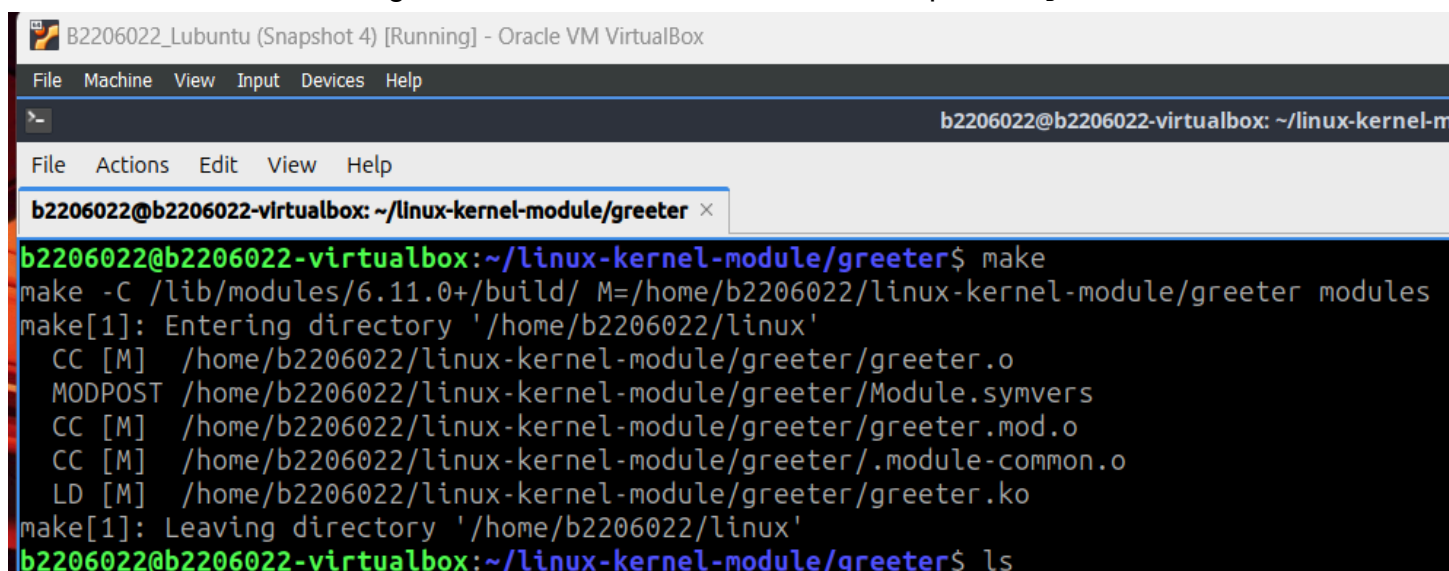
- Clone this repository to your computer:

<https://github.com/TuanThai/linux-kernel-module.git>



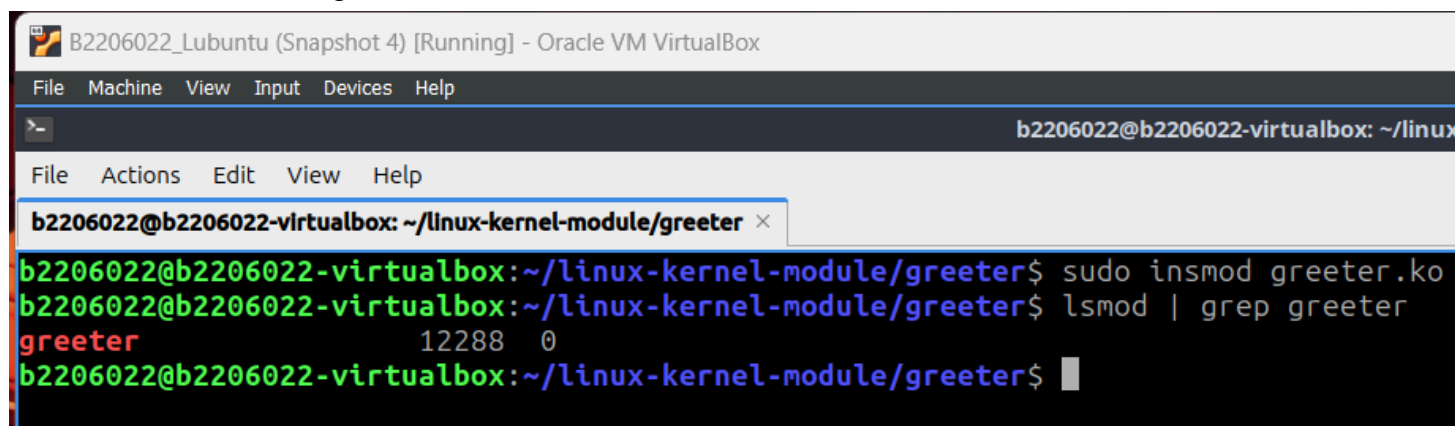
```
B2206022_Lubuntu (Snapshot 4) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
b2206022@b2206022-virtualbox: ~
File Actions Edit View Help
b2206022@b2206022-virtualbox: ~ x
b2206022@b2206022-virtualbox:~$ git clone https://github.com/TuanThai/linux-kernel-module.git
Cloning into '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 (from 1)
Receiving objects: 100% (54/54), 16.35 KiB | 540.00 KiB/s, done.
Resolving deltas: 100% (17/17), done.
```

- Move into greeter/ directory.ls
- Build the module using make command. The module is compiled to greeter.ko



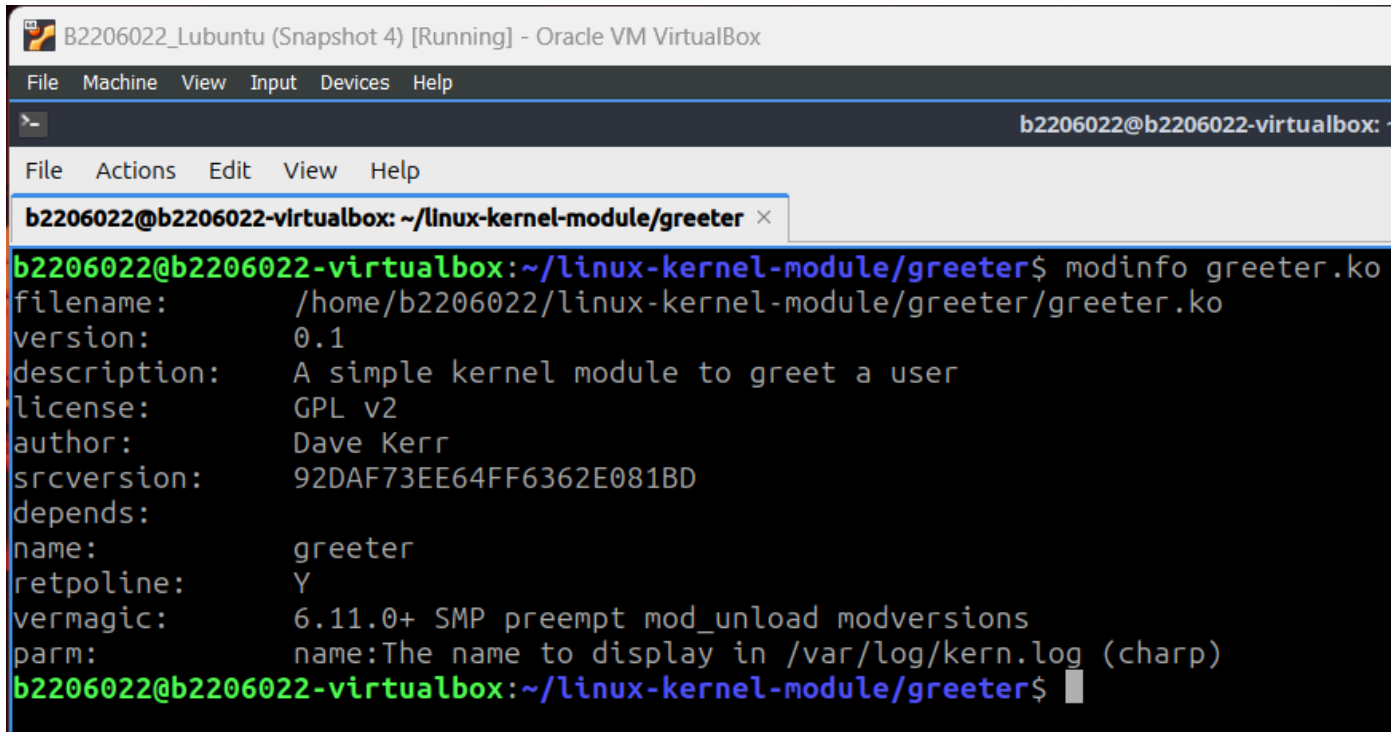
```
B2206022_Lubuntu (Snapshot 4) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
b2206022@b2206022-virtualbox: ~/linux-kernel-m
File Actions Edit View Help
b2206022@b2206022-virtualbox: ~/linux-kernel-module/greeter x
b2206022@b2206022-virtualbox:~/linux-kernel-module/greeter$ make
make -C /lib/modules/6.11.0+/build/ M=/home/b2206022/linux-kernel-module/greeter modules
make[1]: Entering directory '/home/b2206022/linux'
CC [M] /home/b2206022/linux-kernel-module/greeter/greeter.o
MODPOST /home/b2206022/linux-kernel-module/greeter/Module.symvers
CC [M] /home/b2206022/linux-kernel-module/greeter/greeter.mod.o
CC [M] /home/b2206022/linux-kernel-module/greeter/.module-common.o
LD [M] /home/b2206022/linux-kernel-module/greeter/greeter.ko
make[1]: Leaving directory '/home/b2206022/linux'
b2206022@b2206022-virtualbox:~/linux-kernel-module/greeter$ ls
```

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



```
B2206022_Lubuntu (Snapshot 4) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
b2206022@b2206022-virtualbox: ~/linux
File Actions Edit View Help
b2206022@b2206022-virtualbox: ~/linux-kernel-module/greeter x
b2206022@b2206022-virtualbox:~/linux-kernel-module/greeter$ sudo insmod greeter.ko
b2206022@b2206022-virtualbox:~/linux-kernel-module/greeter$ lsmod | grep greeter
greeter                12288  0
b2206022@b2206022-virtualbox:~/linux-kernel-module/greeter$
```

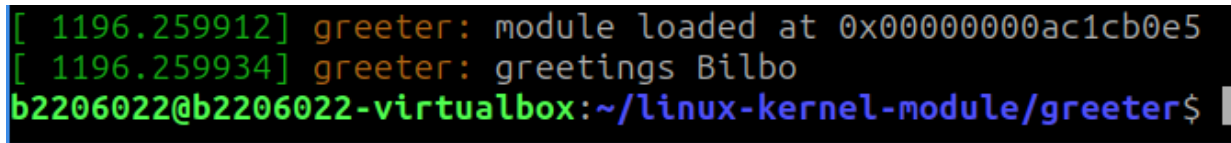
- Show the information of the module using `modinfo greeter.ko`



The screenshot shows a terminal window titled "B2206022\_Lubuntu (Snapshot 4) [Running] - Oracle VM VirtualBox". The terminal has a menu bar with "File", "Machine", "View", "Input", "Devices", and "Help". Below the menu bar, the prompt is "b2206022@b2206022-virtualbox: ~". A tab is open with the title "b2206022@b2206022-virtualbox: ~/linux-kernel-module/greeter". The terminal shows the command `modinfo greeter.ko` and its output:

```
b2206022@b2206022-virtualbox:~/linux-kernel-module/greeter$ modinfo greeter.ko
filename:      /home/b2206022/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:
name:          greeter
retpoline:     Y
vermagic:      6.11.0+ SMP preempt mod_unload modversions
parm:          name:The name to display in /var/log/kern.log (charp)
b2206022@b2206022-virtualbox:~/linux-kernel-module/greeter$
```

- Show kernel log with `dmesg`

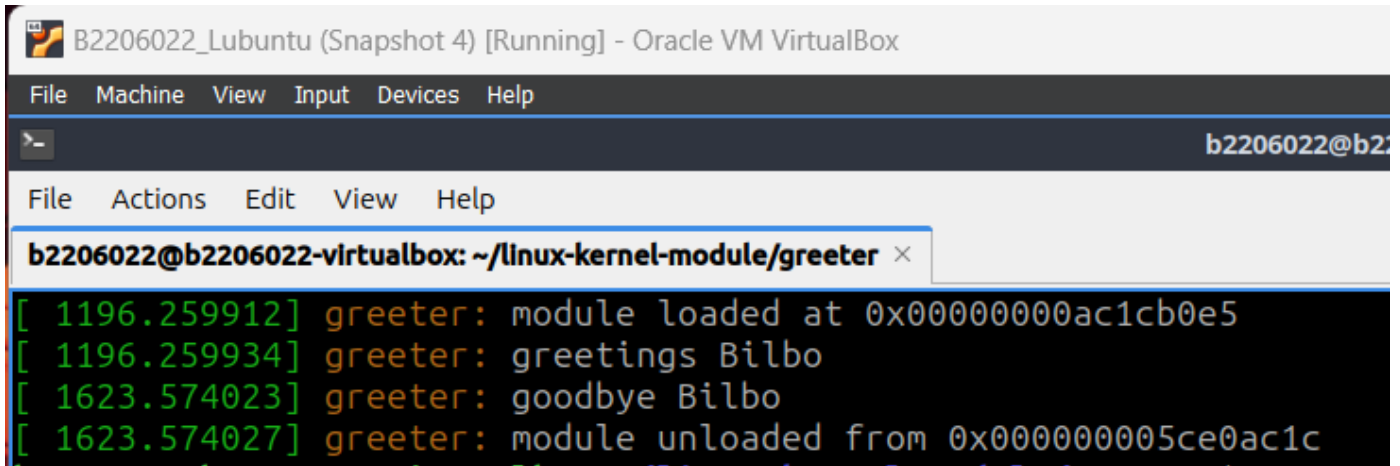


The screenshot shows a terminal window with the prompt "b2206022@b2206022-virtualbox: ~/linux-kernel-module/greeter". The terminal shows the output of the `dmesg` command:

```
[ 1196.259912] greeter: module loaded at 0x00000000ac1cb0e5
[ 1196.259934] greeter: greetings Bilbo
b2206022@b2206022-virtualbox:~/linux-kernel-module/greeter$
```

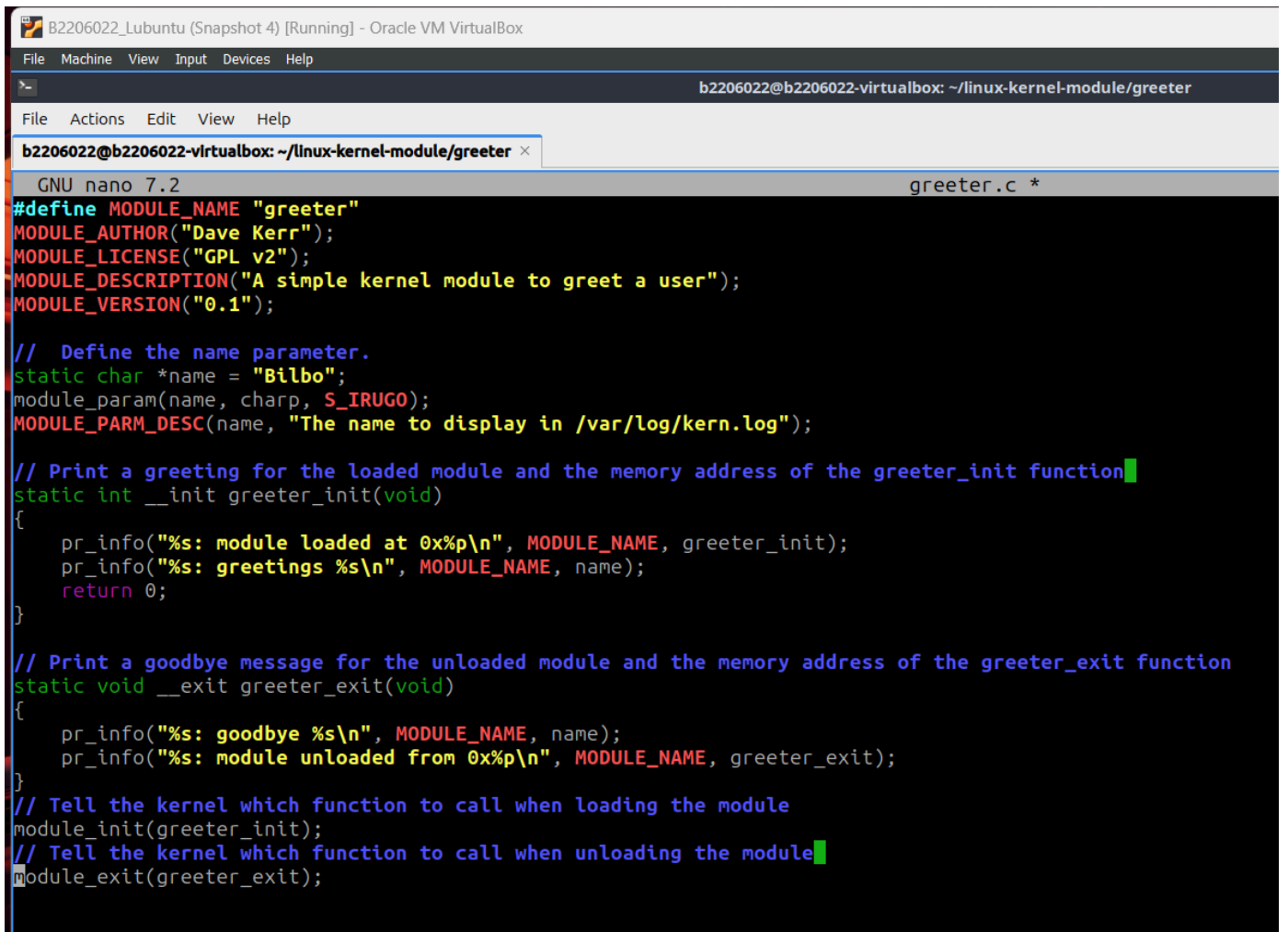
- Remove the module using `rmmod greeter.ko` command, then show that the module has been removed using `lsmod | grep greeter` command.
- Show kernel log with `dmesg`





```
B2206022_Lubuntu (Snapshot 4) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
b2206022@b2206022-virtualbox: ~/linux-kernel-module/greeter x
File Actions Edit View Help
b2206022@b2206022-virtualbox: ~/linux-kernel-module/greeter x
[ 1196.259912] greeter: module loaded at 0x00000000ac1cb0e5
[ 1196.259934] greeter: greetings Bilbo
[ 1623.574023] greeter: goodbye Bilbo
[ 1623.574027] greeter: module unloaded from 0x000000005ce0ac1c
```

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



```
B2206022_Lubuntu (Snapshot 4) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
b2206022@b2206022-virtualbox: ~/linux-kernel-module/greeter
File Actions Edit View Help
b2206022@b2206022-virtualbox: ~/linux-kernel-module/greeter x
GNU nano 7.2 greeter.c *
#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");

// Print a greeting for the loaded module and the memory address of the greeter_init function
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;
}

// Print a goodbye message for the unloaded module and the memory address of the greeter_exit function
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);
}

// Tell the kernel which function to call when loading the module
module_init(greeter_init);
// Tell the kernel which function to call when unloading the module
module_exit(greeter_exit);
```

`greeter_init:`

- **Purpose:** This is the initialization function that is called when a module is loaded into the kernel.
- **pr\_info("%s: module loaded at 0x%p\n", MODULE\_NAME, greeter\_init);** -> Prints information of the loaded module including name, memory address of the **greeter\_init** function.
- **pr\_info("%s: greetings %s\n", MODULE\_NAME, name);**-> Prints a greeting, using the name provided by the user via the name parameter (default is "Bilbo").

`greeter_exit:`

- **Purpose:** This is the cleanup function that is called when the module is removed from the kernel.
- **pr\_info("%s: goodbye %s\n", MODULE\_NAME, name);** -> Prints a goodbye message, using the name provided by the user via the name parameter (default is "Bilbo").
- Prints information of the removed module including the module name, the name parameter (default is "Bilbo").
- **pr\_info("%s: module unloaded from 0x%p\n", MODULE\_NAME, greeter\_exit);**-> Prints a message when the module is unloaded and shows the memory address of the **greeter\_exit** function.

`module_init(greeter_init)` -> tells the kernel which function to call when loading the module.

`module_exit(greeter_exit)` -> tells the kernel which function to call when unloading the module.

(take screenshots to show that you finish this exercise)

---END---