LAB 4

LINUX KERNEL DEVELOPMENT

Fullname: Tran Dang Khoa

Student ID: B2014926

- 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
```

```
khoab2014926@khoab2014926-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
dev.cdrom.info = Can play audio:                1
dev.cdrom.info = Can write CD-R:                0
dev.cdrom.info = Can write CD-RW:       0
```

- List all available TCP congestion control algorithms:

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

```
khoab2014926@khoab2014926-virtualbox:~$ sysctl net.ipv4.tcp_available_congest
ion_control
net.ipv4.tcp_available_congestion_control = reno cubic
khoab2014926@khoab2014926-virtualbox:~$
```

- Show which TCP congestion control algorithm is using:

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

```
khoab2014926@khoab2014926-virtualbox:~$ sysctl net.ipv4.tcp_congestion_contro
l
net.ipv4.tcp_congestion_control = cubic
khoab2014926@khoab2014926-virtualbox:~$
```

- Install `bbr` TCP congestion control algorithm module:

```
sudo modprobe tcp_bbr
```

```
khoab2014926@khoab2014926-virtualbox:~$ sudo modprobe tcp_bbr
[sudo] password for khoab2014926:
khoab2014926@khoab2014926-virtualbox:~$
```

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

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

```
khoab2014926@khoab2014926-virtualbox:~$ sudo sysctl -w net.ipv4.tcp_congestio
n_control=bbr
net.ipv4.tcp_congestion_control = bbr
khoab2014926@khoab2014926-virtualbox:~$
```

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

```
khoab2014926@khoab2014926-virtualbox:~$ sysctl net.ipv4.tcp_congestion_contro
l
net.ipv4.tcp_congestion_control = bbr
khoab2014926@khoab2014926-virtualbox:~$
```

(take screenshots to show that you finish this exercise)

2. Install new kernel version

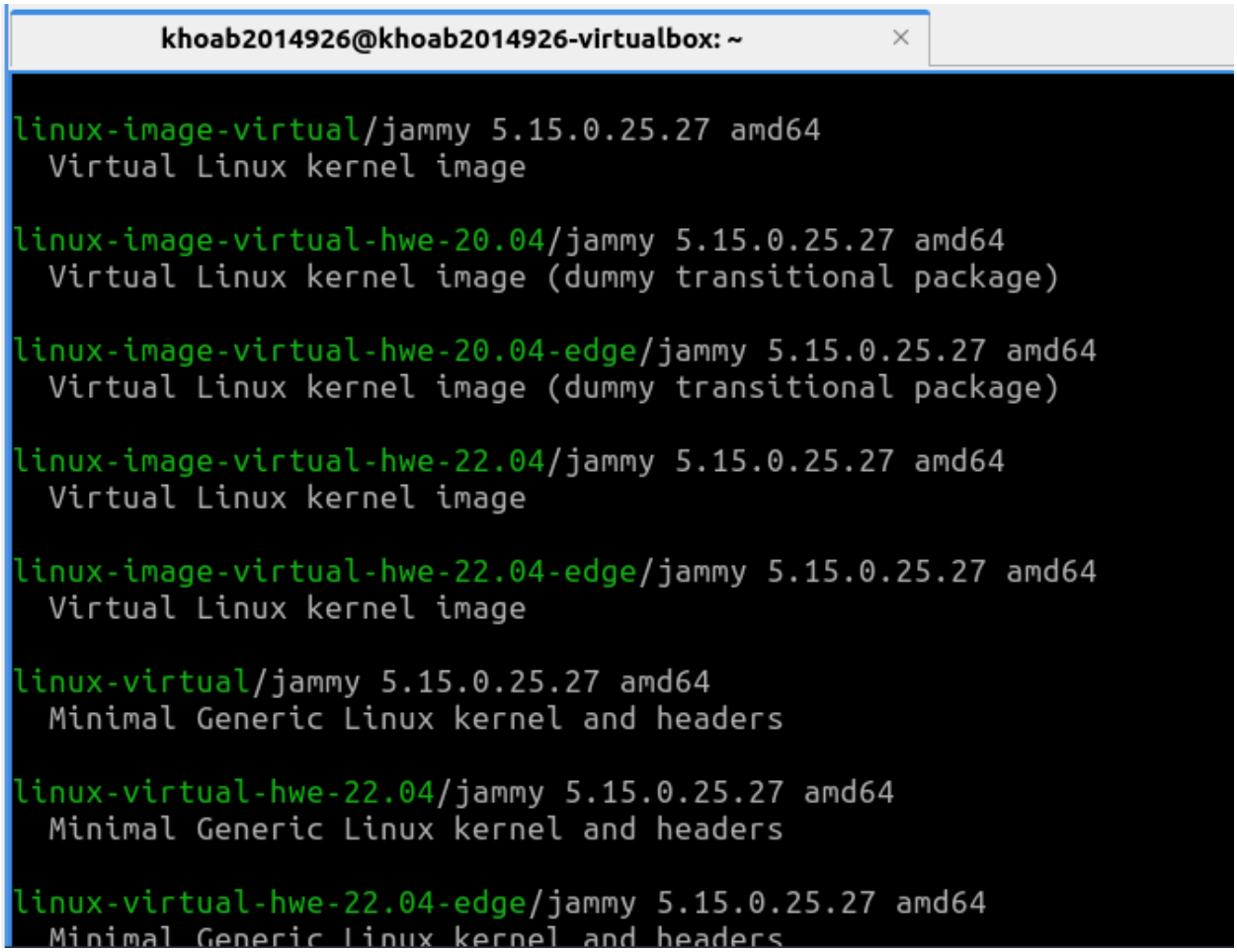
- Show your current kernel version:

```
uname -r
```

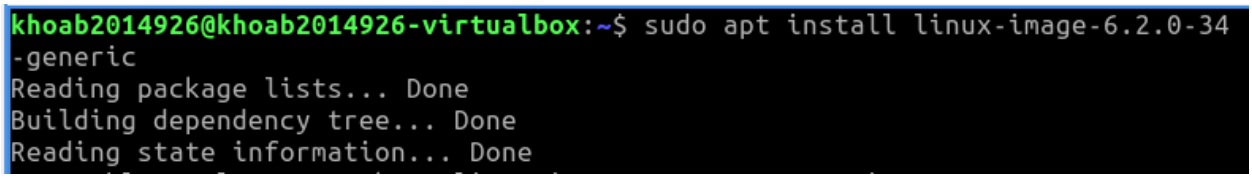
```
khoab2014926@khoab2014926-virtualbox:~$ uname -r
5.15.0-25-generic
khoab2014926@khoab2014926-virtualbox:~$
```

- Search for newer versions:

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



- Install the latest version you find:
sudo apt install linux-image-6.2.0-34-generic



=> Result: Accomplished

```
Preparing to unpack .../linux-modules-6.2.0-34-generic_6.2.0-34.34~22.04.1_amd64.deb ...
Unpacking linux-modules-6.2.0-34-generic (6.2.0-34.34~22.04.1) ...
Selecting previously unselected package linux-image-6.2.0-34-generic.
Preparing to unpack .../linux-image-6.2.0-34-generic_6.2.0-34.34~22.04.1_amd64.deb ...
Unpacking linux-image-6.2.0-34-generic (6.2.0-34.34~22.04.1) ...
Setting up linux-modules-6.2.0-34-generic (6.2.0-34.34~22.04.1) ...
Setting up linux-image-6.2.0-34-generic (6.2.0-34.34~22.04.1) ...
I: /boot/vmlinuz is now a symlink to vmlinuz-6.2.0-34-generic
I: /boot/initrd.img is now a symlink to initrd.img-6.2.0-34-generic
Processing triggers for linux-image-6.2.0-34-generic (6.2.0-34.34~22.04.1) ...
/etc/kernel/postinst.d/initramfs-tools:
update-initramfs: Generating /boot/initrd.img-6.2.0-34-generic
/etc/kernel/postinst.d/zz-update-grub:
Sourcing file `/etc/default/grub'
Sourcing file `/etc/default/grub.d/init-select.cfg'
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-6.2.0-34-generic
Found initrd image: /boot/initrd.img-6.2.0-34-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 partitions.
Systems on them will not be added to the GRUB boot configuration.
Check GRUB_DISABLE_OS_PROBER documentation entry.
done
khoab2014926@khoab2014926-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
```

```
khoab2014926@khoab2014926-virtualbox:~$ uname -r
6.2.0-34-generic
khoab2014926@khoab2014926-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
```

```
khoab2014926@khoab2014926-virtualbox:~$ sudo apt update
[sudo] password for khoab2014926:
Hit:1 http://archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:3 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:4 http://archive.ubuntu.com/ubuntu jammy-backports InRelease [109 kB]
Get:5 http://archive.ubuntu.com/ubuntu jammy-updates/main i386 Packages [508 kB]
Get:6 http://security.ubuntu.com/ubuntu jammy-security/main i386 Packages [346 kB]
Get:7 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1.066 kB]
Get:8 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [859 kB]
Get:9 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 DEP-11 Metadata [101 kB]
Get:10 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 c-n-f Metadata [16,0 kB]
Get:11 http://archive.ubuntu.com/ubuntu jammy-updates/universe i386 Packages [659 kB]
Get:12 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [993 kB]
Get:13 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [176 kB]
Get:14 http://archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [217 kB]
Get:15 http://security.ubuntu.com/ubuntu jammy-security/main amd64 DEP-11 Metadata [43,0 kB]
Get:16 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64 DEP-11 Metadata [305 kB]
Get:17 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [955 kB]
Get:18 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64 c-n-f Metadata [22,0 kB]
Get:19 http://archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 DEP-11 Metadata [940 B]
Get:20 http://archive.ubuntu.com/ubuntu jammy-backports/main amd64 DEP-11 Metadata [4.916 B]
Get:21 http://archive.ubuntu.com/ubuntu jammy-backports/universe amd64 DEP-11 Metadata [17,7 kB]
Get:22 http://security.ubuntu.com/ubuntu jammy-security/restricted Translation-en [154 kB]
Get:23 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [791 kB]
Get:24 http://security.ubuntu.com/ubuntu jammy-security/universe i386 Packages [561 kB]
Get:25 http://security.ubuntu.com/ubuntu jammy-security/universe Translation-en [145 kB]
Get:26 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 DEP-11 Metadata [55,1 kB]
Fetched 8.333 kB in 12s (707 kB/s)
Reading package lists... Done
```

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

```
khoab2014926@khoab2014926-virtualbox:~$ sudo apt-get install build-essential vim git cscope libncur
ses-dev libssl-dev bison flex libelf-dev bc git-email -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
bc is already the newest version (1.07.1-3build1).
bison is already the newest version (2:3.8.2+dfsg-1build1).
build-essential is already the newest version (12.9ubuntu3).
flex is already the newest version (2.6.4-8build2).
libelf-dev is already the newest version (0.186-1build1).
cscope is already the newest version (15.9-1).
git is already the newest version (1:2.34.1-1ubuntu1.10).
libncurses-dev is already the newest version (6.3-2ubuntu0.1).
libssl-dev is already the newest version (3.0.2-0ubuntu1.10).
vim is already the newest version (2:8.2.3995-1ubuntu2.12).
git-email is already the newest version (1:2.34.1-1ubuntu1.10).
0 upgraded, 0 newly installed, 0 to remove and 423 not upgraded.
khoab2014926@khoab2014926-virtualbox:~$
```

=> Result: Accomplished

```
Setting up cscope (15.9-1) ...
Setting up libnet-dns-perl (1.33-1) ...
Setting up vim-tiny (2:8.2.3995-1ubuntu2.12) ...
Setting up libgcc-11-dev:amd64 (11.4.0-1ubuntu1~22.04) ...
Setting up bison (2:3.8.2+dfsg-1build1) ...
update-alternatives: using /usr/bin/bison.yacc to provide /usr/bin/yacc (yacc) in auto mode
Setting up libc6-dev:amd64 (2.35-0ubuntu3.4) ...
Setting up binutils-x86-64-linux-gnu (2.38-4ubuntu2.3) ...
Setting up flex (2.6.4-8build2) ...
Setting up libncurses-dev:amd64 (6.3-2ubuntu0.1) ...
Setting up libemail-valid-perl (1.202-1) ...
Setting up libnet-dns-sec-perl (1.19-1build2) ...
Setting up binutils (2.38-4ubuntu2.3) ...
Setting up libfl-dev:amd64 (2.6.4-8build2) ...
Setting up dpkg-dev (1.21.1ubuntu2.2) ...
Setting up libstdc++-11-dev:amd64 (11.4.0-1ubuntu1~22.04) ...
Setting up zlib1g-dev:amd64 (1:1.2.11.dfsg-2ubuntu9.2) ...
Setting up gcc-11 (11.4.0-1ubuntu1~22.04) ...
Setting up g++-11 (11.4.0-1ubuntu1~22.04) ...
Setting up gcc (4:11.2.0-1ubuntu1) ...
Setting up libelf-dev:amd64 (0.186-1build1) ...
Setting up g++ (4:11.2.0-1ubuntu1) ...
update-alternatives: using /usr/bin/g++ to provide /usr/bin/c++ (c++) in auto mode
Setting up build-essential (12.9ubuntu3) ...
Processing triggers for mailcap (3.70+nmu1ubuntu1) ...
Processing triggers for desktop-file-utils (0.26-1ubuntu3) ...
Processing triggers for hicolor-icon-theme (0.17-2) ...
Processing triggers for libc-bin (2.35-0ubuntu3) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for install-info (6.8-4build1) ...
khoab2014926@khoab2014926-virtualbox:~$ c
```

- Clone a mainline kernel source code to your computer:

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

```
khoab2014926@khoab2014926-virtualbox:~$ git clone --depth=1 https://github.com/torvalds/linux.git
Cloning into 'linux'...
remote: Enumerating objects: 86634, done.
remote: Counting objects: 100% (86634/86634), done.
remote: Compressing objects: 100% (77795/77795), done.
remote: Total 86634 (delta 8480), reused 71413 (delta 7946), pack-reused 0
Receiving objects: 100% (86634/86634), 240.20 MiB | 1.19 MiB/s, done.
Resolving deltas: 100% (8480/8480), done.
Updating files: 100% (81762/81762), done.
khoab2014926@khoab2014926-virtualbox:~$
```

- 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
```

```
khoab2014926@khoab2014926-virtualbox:~$ lsmod > /tmp/my-lsmod
khoab2014926@khoab2014926-virtualbox:~$ make LSMOD=/tmp/my-lsmod localmodconfig
```

=> Result: Accomplished

```
  Test BPF filter functionality (TEST_BPF) [N/m/?] n
  Test blackhole netdev functionality (TEST_BLACKHOLE_DEV) [N/m/?] n
  Test find_bit functions (FIND_BIT_BENCHMARK) [N/m/y/?] n
  Test firmware loading via userspace interface (TEST_FIRMWARE) [N/m/y/?] n
  sysctl test driver (TEST_SYSCTL) [N/m/y/?] n
  udelay test driver (TEST_UDELAY) [N/m/y/?] n
  Test static keys (TEST_STATIC_KEYS) [N/m/?] n
  Test DYNAMIC_DEBUG (TEST_DYNAMIC_DEBUG) [N/m/y/?] (NEW)
  kmod stress tester (TEST_KMOD) [N/m/?] n
  Test memcat_p() helper function (TEST_MEMCAT_P) [N/m/y/?] n
  Test livepatching (TEST_LIVEPATCH) [N/m/?] n
  Test heap/page initialization (TEST_MEMINIT) [N/m/y/?] n
  Test HMM (Heterogeneous Memory Management) (TEST_HMM) [N/m/y/?] n
  Test freeing pages (TEST_FREE_PAGES) [N/m/y/?] n
  Test floating point operations in kernel space (TEST_FPU) [N/m/y/?] n
  Test clocksource watchdog in kernel space (TEST_CLOCKSOURCE_WATCHDOG) [N/m/
y/?] n
#
# configuration written to .config
#
```

- Disable certificate stuff:

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

```
khoadangtran@khoadangtran-virtualbox:~/linux$ scripts/config --disable SYSTEM
_TRUSTED_KEYS
khoadangtran@khoadangtran-virtualbox:~/linux$ scripts/config --disable SYSTEM
_REVOCATION_KEYS
khoadangtran@khoadangtran-virtualbox:~/linux$
```

- 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
```

```
  AR      drivers/ras/built-in.a
  CC [M]  drivers/parport/share.o
  AR      drivers/nvmem/built-in.a
  CC [M]  drivers/parport/ieee1284.o
  CC [M]  drivers/parport/ieee1284_ops.o
  CC      drivers/interconnect/bulk.o
  CC [M]  drivers/parport/procfs.o
  CC      drivers/interconnect/debugfs-client.o
  CC [M]  drivers/parport/daisy.o
  CC [M]  drivers/parport/probe.o
  AR      drivers/interconnect/built-in.a
  AR      drivers/built-in.a
  CC [M]  drivers/parport/parport_pc.o
  LD [M]  drivers/parport/parport.o
  AR      built-in.a
  AR      vmlinux.a
  LD      vmlinux.o
Killed
make[2]: *** [scripts/Makefile.vmlinux_o:61: vmlinux.o] Error 137
make[2]: *** Deleting file 'vmlinux.o'
make[1]: *** [/home/khoab2014926/linux/Makefile:1146: vmlinux_o] Error 2
make: *** [Makefile:234: __sub-make] Error 2
khoab2014926@khoab2014926-virtualbox:~/linux$
```

- Install the new kernel:

```
sudo make modules_install install
```

```
khoab2014926@khoab2014926-virtualbox:~/linux$   sudo make modules_install ins
tall
[sudo] password for khoab2014926:
  INSTALL /lib/modules/6.6.0-rc6+/modules.order
  SYMLINK /lib/modules/6.6.0-rc6+/build
make[3]: *** No rule to make target 'modules.builtin', needed by '/lib/module
s/6.6.0-rc6+/modules.builtin'.  Stop.
make[2]: *** [Makefile:1821: modules_install] Error 2
make[1]: *** [/home/khoab2014926/linux/Makefile:359: __build_one_by_one] Erro
r 2
make: *** [Makefile:234: __sub-make] Error 2
```

- 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
```

(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
```

```
khoab2014926@khoab2014926-virtualbox:~/linux$ git checkout -b first-patch
Switched to a new branch 'first-patch'
khoab2014926@khoab2014926-virtualbox:~/linux$
```

- Update the kernel

```
git fetch origin
```

```
khoab2014926@khoab2014926-virtualbox:~/linux$ git fetch origin
khoab2014926@khoab2014926-virtualbox:~/linux$ git branch
* first-patch
  master
khoab2014926@khoab2014926-virtualbox:~/linux$
```

- 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.

```
khoab2014926@khoab2014926-virtualbox:~/linux$ lsmod | grep e1000
e1000                 155648  0
khoab2014926@khoab2014926-virtualbox:~/linux$
```

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

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

```
khoab2014926@khoab2014926-virtualbox:~/linux$ git grep e1000 -- '*Makefile'
arch/arm64/boot/dts/arm/Makefile:dtb-$(CONFIG_ARCH_VEXPRESS) += corstone1000-
fvp.dtb corstone1000-mps3.dtb
drivers/net/ethernet/intel/Makefile:obj-$(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 e1000_hw
.o e1000_ethtool.o e1000_param.o
drivers/net/ethernet/intel/e1000e/Makefile:obj-$(CONFIG_E1000E) += e1000e.o
drivers/net/ethernet/intel/e1000e/Makefile:e1000e-objs := 82571.o ich8lan.o 8
0003es2lan.o \
drivers/net/ethernet/intel/igb/Makefile:igb-objs := igb_main.o igb_ethtool.o
e1000_82575.o \
drivers/net/ethernet/intel/igb/Makefile:            e1000_mac.o e1000_nvm.o e
1000_phy.o e1000_mbx.o \
drivers/net/ethernet/intel/igb/Makefile:            e1000_i210.o igb_ptp.o ig
b_hwmon.o
khoab2014926@khoab2014926-virtualbox:~/linux$
```

- 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;
...
```

```
khoab2014926@khoab2014926-virtualbox:~/linux$ nano drivers/net/ethernet/intel
/e1000/e1000_main.c
```

```
khoab2014926@khoab2014926-virtualbox: ~/linux
  GNU nano 6.2    drivers/net/ethernet/intel/e1000/e1000_main.c *
 * e1000_probe initializes an adapter identified by a pci_dev structure.
 * The OS initialization, configuring of the adapter private structure,
 * and a hardware reset occur.
 **/
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;

        static int cards_found;
        static int global_quad_port_a; /* global ksp3 port a indication */
        int i, err, pci_using_dac;
        u16 eeprom_data = 0;
        u16 tmp = 0;
        u16 eeprom_apme_mask = E1000_EEPROM_APME;
        int bars, need_ioport;
        bool disable_dev = false;
```

- Compile and install your changes:

```
make -j3
```

```
khoab2014926@khoab2014926-virtualbox:~/linux$ make -j3
  DESCEND objtool
  INSTALL libsubcmd_headers
  CALL    scripts/checksyscalls.sh
  CC [M]  drivers/net/ethernet/intel/e1000/e1000_main.o
  LD [M]  drivers/net/ethernet/intel/e1000/e1000.o
  LD      vmlinux.o
Killed
```

```
sudo make modules_install install
```

```
khoab2014926@khoab2014926-virtualbox:~/linux$ sudo make modules_install insta
ll
  INSTALL /lib/modules/6.6.0-rc6+/modules.order
  SYMLINK /lib/modules/6.6.0-rc6+/build
make[3]: *** No rule to make target 'modules.builtin', needed by '/lib/module
s/6.6.0-rc6+/modules.builtin'.  Stop.
make[2]: *** [Makefile:1821: modules_install] Error 2
make[1]: *** [/home/khoab2014926/linux/Makefile:359: __build_one_by_one] Erro
r 2
make: *** [Makefile:234: __sub-make] Error 2
```

- 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"
```

- Committing changes, and view your commit

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

```
khoab2014926@khoab2014926-virtualbox:~/linux$ git add .
khoab2014926@khoab2014926-virtualbox:~/linux$ git commit -s -v -m "My first k
ernel patch"
```

```
git show HEAD
```

```
commit dd72f9c7e512da377074d47d990564959b772643 (grafted, HEAD -> first-patch
, origin/master, origin/HEAD, master)
Author: Linus Torvalds <torvalds@linux-foundation.org>
Date:   Wed Oct 18 09:37:36 2023 -0700

    Merge tag 'spi-fix-v6-6-rc4' of git://git.kernel.org/pub/scm/linux/kernel
/git/broonie/spi

    Pull spi fix from Mark Brown:
     "A fix for the npcm-fiu driver in cases where there are no dummy bytes
      during reads"

    * tag 'spi-fix-v6-6-rc4' of git://git.kernel.org/pub/scm/linux/kernel/git
/broonie/spi:
      spi: npcm-fiu: Fix UMA reads when dummy.nbytes == 0

diff --git a/.clang-format b/.clang-format
new file mode 100644
index 000000000..0bbb1991d
--- /dev/null
+++ b/.clang-format
@@ -0,0 +1,689 @@
:
```

- Find whom to send the patch to

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

- 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.
```

- 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>
```

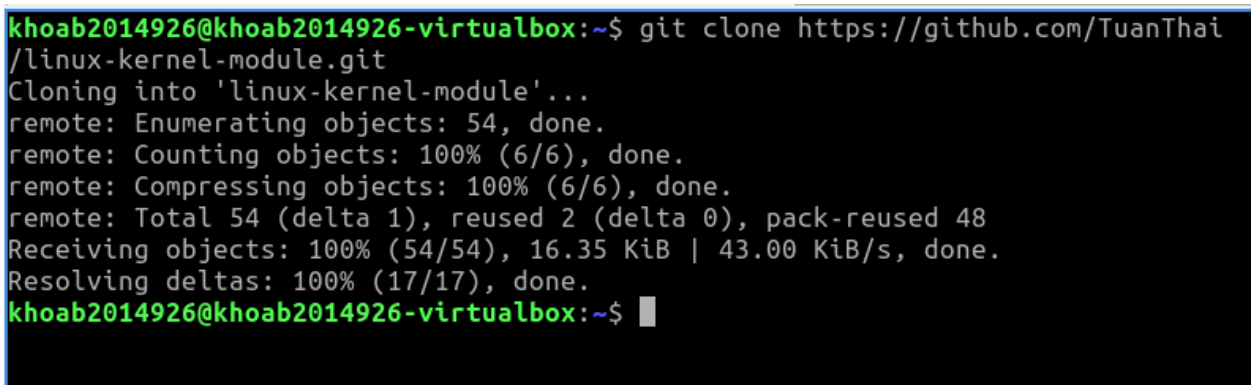
(take screenshots to show that you finish this exercise)

5. Writing a simple Linux kernel module: Greeter sample

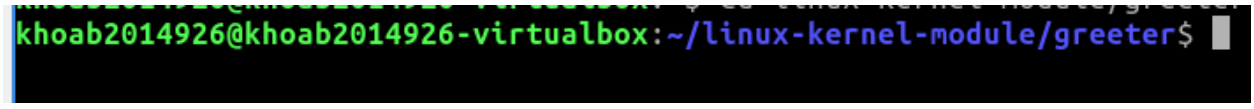
This module simply takes a name as a parameter, and writes a greeting to the kernel log (/var/log/kern.log):

- Clone this repository to your computer:

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



- Move into `greeter/` directory.



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

```
khoab2014926@khoab2014926-virtualbox:~/linux-kernel-module/greeter$ make
make -C /lib/modules/5.15.0-25-generic/build/ M=/home/khoab2014926/linux-kern
el-module/greeter modules
make[1]: Entering directory '/usr/src/linux-headers-5.15.0-25-generic'
warning: the compiler differs from the one used to build the kernel
  The kernel was built by: gcc (Ubuntu 11.2.0-19ubuntu1) 11.2.0
  You are using:           gcc (Ubuntu 11.4.0-1ubuntu1~22.04) 11.4.0
  CC [M]  /home/khoab2014926/linux-kernel-module/greeter/greeter.o
  MODPOST /home/khoab2014926/linux-kernel-module/greeter/Module.symvers
  CC [M]  /home/khoab2014926/linux-kernel-module/greeter/greeter.mod.o
  LD [M]  /home/khoab2014926/linux-kernel-module/greeter/greeter.ko
  BTF [M] /home/khoab2014926/linux-kernel-module/greeter/greeter.ko
Skipping BTF generation for /home/khoab2014926/linux-kernel-module/greeter/gr
eeter.ko due to unavailability of vmlinux
make[1]: Leaving directory '/usr/src/linux-headers-5.15.0-25-generic'
khoab2014926@khoab2014926-virtualbox:~/linux-kernel-module/greeter$
```

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

```
khoab2014926@khoab2014926-virtualbox:~/linux-kernel-module/greeter$ sudo insm
od greeter.ko
[sudo] password for khoab2014926:
khoab2014926@khoab2014926-virtualbox:~/linux-kernel-module/greeter$ lsmod | g
rep greeter
greeter                16384  0
khoab2014926@khoab2014926-virtualbox:~/linux-kernel-module/greeter$
```

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

```
khoab2014926@khoab2014926-virtualbox:~/linux-kernel-module/greeter$ modinfo g
reeter.ko
filename:       /home/khoab2014926/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:       5.15.0-25-generic SMP mod_unload modversions
parm:           name:The name to display in /var/log/kern.log (charp)
khoab2014926@khoab2014926-virtualbox:~/linux-kernel-module/greeter$
```

- Show kernel log with `dmesg`

```
[ 6954.670597] [    910]  1000   910      722        0    45056       27
        0 lubuntu-upg-not
[ 6954.670604] [    924]  1000   924    79400      209   110592       31
        0 gvfsd-trash
[ 6954.670612] [    932]  1000   932    79659      159   114688      188
        0 gvfs-udisks2-vo
[ 6954.670620] [    937]  1000   937    59832      105    98304       38
        0 gvfs-mtp-volume
[ 6954.670628] [    941]  1000   941    60071      111    98304       61
        0 gvfs-gphoto2-vo
[ 6954.670635] [    945]  1000   945    59874      110    90112       38
        0 gvfs-goa-volume
[ 6954.670643] [    949]  1000   949    79517      148   110592      105
        0 gvfs-afc-volume
[ 6954.670651] [    961]     0   961    61304      138   102400      151
        0 upowerd
[ 6954.670658] [    973]  1000   973   127921      486   294912     2743
        0 lxqt-powermanag
[ 6954.670666] [    975]  1000   975    68254      274   241664     1116
        0 qlipper
[ 6954.670673] [    977]  1000   977    68439       31   237568     1214
        0 nm-tray
[ 6954.670681] [    993]  1000   993     2811        0    57344       25
```

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

```
khoab2014926@khoab2014926-virtualbox:~/linux-kernel-module/greeter$ lsmod |
grep greeter
khoab2014926@khoab2014926-virtualbox:~/linux-kernel-module/greeter$
```

- Show kernel log with `dmesg`

```
[13633.985266] [  93190]  1000 93190   596653     1855   434176     1077
        0 Web Content
[13633.985273] [  99323]  1000 99323   101703      990   454656     3783
        0 qterminal
[13633.985281] [  99344]  1000 99344     3549       90    73728      326
        0 bash
[13633.985288] [ 103029]  1000 103029     3007        6    61440      164
        0 make
[13633.985295] [ 103030]  1000 103030      724        0    40960       27
        0 sh
[13633.985302] [ 103056]  1000 103056   331782   330979  2695168        0
        0 objtool
[13633.985309] oom-kill:constraint=CONSTRAINT_NONE,nodemask=(null),cpuset=/,m
ems_allowed=0,global_oom,task_memcg=/user.slice/user-1000.slice/session-3.sco
pe,task=objtool,pid=103056,uid=1000
[13633.985334] Out of memory: Killed process 103056 (objtool) total-vm:132712
8kB, anon-rss:1323916kB, file-rss:0kB, shmem-rss:0kB, UID:1000 pgtables:2632k
B oom_score_adj:0
[13634.241393] oom_reaper: reaped process 103056 (objtool), now anon-rss:0kB,
 file-rss:0kB, shmem-rss:0kB
[13904.379002] greeter: goodbye Bilbo
[13904.379007] greeter: module unloaded from 0x00000000cf9f1f15
khoab2014926@khoab2014926-virtualbox:~/linux-kernel-module/greeter$
```

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

khoab2014926@khoab2014926-v...linux-kernel-module/greeter

GNU nano 6.2 greeter.c

```c
#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);
```

[Read 30 lines]

greeter_init: This function is called when the module is loaded into the kernel
with insmod. It logs the name of the module and its memory address, and then logs a

greeting message with the name parameter.
greeter_exit: This function is called when the module is removed from the kernel
with rmmod. It logs a goodbye message with the name parameter, and then logs that the
module has been unloaded along with its memory address.
module_init(greeter_init): The greeter_init function will be called to initialize the
module and perform the necessary work.
module_exit(greeter_exit: The greeter_exit function will be called to perform cleanup
and free up resources.
(take screenshots to show that you finish this exercise)

---END---