

Goal: Compiling a custom Linux kernel on a machine running Ubuntu OS. Create and boot a Ubuntu virtual machine (VM) using the custom kernel.

We will also call your desktop/laptop the **host** machine.

The virtual machine (VM) is sometimes called a **guest** machine.

Create a folder named **project**

Install QEMU

```
sudo apt install qemu-kvm
```

Creating a virtual machine

Download Ubuntu desktop iso from <https://ubuntu.com/download/desktop> in the project folder.

From the project folder, run the following commands.

```
qemu-img create -f qcow2 linux.qcow2 20G
qemu-system-x86_64 -enable-kvm -m 6G -cdrom ${PWD}/ubuntu-image-name.iso -boot d
linux.qcow2
```

Follow the instructions to install Ubuntu

After the installation is done, boot the virtual machine to check if everything is okay.

```
qemu-system-x86_64 -enable-kvm -m 6G linux.qcow2
```

Download kernel version 6.16 from <https://www.kernel.org/> in the project folder.

Untar it

Install basic packages

```
sudo apt install build-essential flex bison libelf-dev libssl-dev libdw-dev gawk
```

```
cd linux-6.16
cp /boot/config-$(uname -r) .config
make localmodconfig
```

Press Enter for all options

Run:

```
scripts/config --disable SYSTEM_TRUSTED_KEYS
scripts/config --disable SYSTEM_REVOCATION_KEYS
scripts/config --module E1000
```

```
make -j `nproc`
```

Press Enter for all options

Install modules

```
make modules_install INSTALL_MOD_PATH=`pwd`/..
```

The previous command will create a folder named lib in the **project** folder.

```
cd ..
```

Create a tar file from the lib directory

```
tar -cvf lib.tar.bz2 lib
```

Enable SSH on host machine:

```
sudo apt install openssh-server
```

```
sudo systemctl enable ssh
```

```
sudo systemctl start ssh
```

Boot VM again:

```
qemu-system-x86_64 -enable-kvm -m 6G linux.qcow2
```

Copy lib.tar.bz2 from host to guest.

If the host ip is 192.168.11.23

(Use ifconfig on the host to find its IP address.)

We will assume that the host IP address is 192.168.11.23 in the rest of the document.

You can run:

```
scp username@192.168.11.23:/path-to-project/lib.tar.bz2 .
```

There are other ways you can also copy. Pick the best that works for you.

Untar it:

```
tar -xvf lib.tar.bz2
```

Run:

```
sudo cp -r lib/modules/6.16.0 /lib/modules
```

Now, boot the VM using:

```
qemu-system-x86_64 -enable-kvm -kernel ./linux-6.16/arch/x86_64/boot/bzImage -append  
"root=/dev/sda2 console=ttyS0 nokaslr" -m 6G -hda linux.qcow2
```

Hopefully it'll boot.

Ping to host using:

```
ping 192.168.11.23
```

It should work.

Now edit **drivers/net/ethernet/intel/e1000/e1000_main.c** file in the **linux-6.16** folder as follows.

Locate the function **e1000_xmit_frame**

Add the following lines at the top of the function

```

struct iphdr *iph = ip_hdr(skb);

if (iph) {
    __be32 saddr = iph->saddr;
    __be32 daddr = iph->daddr;
    printk("src IP: %pI4, dst IP: %pI4\n", &saddr, &daddr);
}

```

This function copies the packet to the network device's send buffer for transmission.

Compile the e1000 module using the following command from the linux-6.16 folder:

```
make M=drivers/net/ethernet/intel/e1000
```

It will take a few seconds.

Boot the VM again using the custom kernel using:

```
qemu-system-x86_64 -enable-kvm -kernel ./linux-6.16/arch/x86_64/boot/bzImage -append
"root=/dev/sda2 console=ttyS0 nokaslr" -m 6G -hda linux.qcow2
```

Copy the driver from the host to the VM using

```
scp username@192.168.11.23:/path-to-project/linux-6.16/drivers/net/ethernet/intel/e1000/e1000.ko .
```

Unload the original module using:

```
sudo modprobe -r e1000
```

Load the modified module using

```
sudo insmod e1000.ko
```

Ping google using

```
ping www.google.com
```

It should work

run **sudo dmesg** to see kernel debug messages or output of printk.

If everything is okay, you should see messages like **src IP: ...**

Dump the output of dmesg in a file using:

```
sudo dmesg > out.txt
```

Send the file to the host using:

```
scp username@192.168.11.23:
```

It will copy the file to the home folder of the host.

Submit out.txt file.