Lab 10: Kernel Building and Virtual Machine

In the next part of this class, you are going to learn how to develop device drivers. For now, a device driver is simply a program that runs in a kernel (not in a user's process address space). Therefore, to run a device driver, you need a root privileges. For this lab, we will give you a very small virtual machine that you can login as root. As a result, you can install and run your program in its kernel. To keep it small, we cannot incorporate the C compiler (`gcc`) into that virtual machine. Therefore, you need to write your program on `thoth.cs.pitt.edu` (`cs449.cs.pitt.edu`), compile it, and transfer it into the virtual machine, and run.

1 Build and Load a Kernel Module

1. Login to `thoth.cs.pitt.edu` (`cs449.cs.pitt.edu`), and go to your `/u/SysLab/USERNAME` directory using the following commmand:

   ```
   cd /u/SysLab/USERNAME
   ```

   and replace `USERNAME` by your username.

2. Copy the file `hello_dev.tar.gz` to your directory using the following command:

   ```
   cp /u/SysLab/shared/hello_dev.tar.gz .
   ```

3. Unzip the above file using the following command:

   ```
   gunzip hello_dev.tar.gz
   ```

   The result is an archive file named `hello_dev.tar`

4. Uncompress it using the following command:

   ```
   tar -xvf hello_dev.tar
   ```

   The result is a directory named `hello_dev` which contains three files, `hello.rules`, `hello_dev.c`, and `Makefile`.

5. Go in to `hello_dev` directory using the following command:

   ```
   cd hello_dev
   ```

6. Use an editor of your choice (e.g., `nano`, `pico`, `vim`, etc) to open the file named `Makefile`.

7. Change the line

   ```
   KDIR := /lib/modules/$(shell uname -r)/build
   ```

   to

```
KDIR := /u/SysLab/shared/linux-2.6.23.1
```

Save the file and exit from your text editor.

8. Build the kernel object using the following command:

```
make ARCH=i386
```

The `ARCH=i386` is important because we are building a 32-bit kernel on a 64-bit machine. If all went well, you will get a new file named `hello_dev.ko` and this is your kernel module.

9. Back to your Windows or Mac machine, download the file named `qemu-449.zip` from the CourseWeb and extract it to a directory of your choice. Once you extracted it, you will get a directory named `qemu`.

10. **On Windows** machine, go into the directory `qemu` and double click `qemu-win.bat` to start the virtual machine

    On Mac, you can download `Q-0.9.1d118.dmg` from the following URL:

```
http://people.cs.pitt.edu/~jmisurda/teaching/cs1550/qemu/Q-0.9.1d118.dmg
```

    and install `Q.app`. Open `Q.app` and use `tty.qcow2` as the disk image which is located in `qemu` directory from previous step.

    On Ubuntu, if you do not have `qeum` installed on your system, use the following command:

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

    and execute the following command:

```
qemu-system-i386 tty.qcow2
```

11. When Linux virtual machine boots under `qemu`, login using the `root` as your username and `root` as your password.

12. Now, you need to download the kernel module that you just build in step 8 into your virtual machine using the following command:

```
scp USERNAME@thoth.cs.pitt.edu:/u/SysLab/USERNAME/hello_dev/hello_dev.ko .
```

    Replacing `USERNAME` by your username. **Note** that this is a space and a dot at the end of the above command.

13. To load the kernel module into your kernel, use the following command:

```
insmod hello_dev.ko
```

14. The next step is to make the device file in the `/dev` directory. First, we need to know the MAJOR and MINOR numbers that identify the new device using the following commands:

```
cd /sys/class/misc/hello
cat dev
```

    The output should look like the following:

```
10:63
```

    The 10 indicates the MAJOR number and the 63 indicates the MINOR number.

15. We use the command `mknod` to make a special file. The name will be `hello` and it is a character device. The 10 and the 63 correspond to the MAJOR and MINOR numbers we discovered above (if different, use the ones you saw.) To create the device file, use the following commands:

```
cd /dev
mnod hello c 10 63
```

16. We can now read the data out of `/dev/hello` with a command below:

```
cat /dev/hello
```

    and the output should be

```
Hello World!
```

    which came from the driver.

17. To unload the module from the kernel, use the following commands:

```
rm /dev/hello
rmmod hello_dev.ko
```

    Note that if you want to load this module again, you do not need to complie it. Simply follow step 13 to 15.

Use a Program to Read from a Module

A module can be read/write by a user program as you already saw in step 16 above. In this section, you will use a regular C program to interface with this `hello_dev` module. **Note** that the `hello_dev` module simply send out 14 characters `"Hello, World!\n"` without null-character at the end. To interact with this module, we can simply create a program that read exactly 14 character at once from the device file (`/dev/hello`). This can be done by using the system call `open` to open the device file with the flag `O_RDWR`, then use the system call `read` to read exactly 14 character from the file descriptor as shown below:

```
#include <stdio.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <unistd.h>
#include <errno.h>

int main(void)
{
        char buffer[15];
        int fp;
        int returnValue;

        fp = open("/dev/hello", O_RDWR);
        if(fp == EACCES)
        {
                printf("Unable to open /dev/hello.\n");
                return -1;
        }

        returnValue = read(fp, buffer, 14);
        if(returnValue != 14)
        {
                printf("Error reading\n");
                return -1;
        }
        buffer[14] = '\0';

        printf("%s", buffer);

        close(fp);

        return 0;
}
```

1. Go back to `thoth.cs.pitt.edu` (`cs449.cs.pitt.edu`). Then go to your `/u/SysLab/USERNAME` directory using the following command:

   ```
   cd /u/SysLab/USERNAME
   ```

   by replacing the `USERNAME` by your username.

2. Create `lab10` directory and go there using the following commands:

   ```
   mkdir lab10
   cd lab10
   ```

3. Copy the above file to your directory using the following command:

   ```
   cp /afs/cs.pitt.edu/usr0/tkosiyat/public/cs0449/readHello.c .
   ```

4. Compile the above file using the following command:

   ```
   gcc -m32 -o readHello readHello.c
   ```

5. Go back to the `qeme` virtual machine and transfer the file `readHello` to your virtual machine using the following command:

   ```
   scp USERNAME@thoth.cs.pitt.edu:/u/SysLab/USERNAME/lab10/readHello .
   ```

6. If you already remove the module from previous section, insert the module `hello_dev` again by using step 13 to 15 from previous section.

7. Execute the `readHello` program using the following command:

   ```
   ./readHello
   ```

   and you should see the output generated by `readHello` as shown below:

   ```
   Hello, World!
   ```

What to do?

What you need to do is to get used to all those steps explained above as well as `qemu` and the given virtual machin. You will have to perform the same series of steps many times for your forth project. Make sure there is no problem compiling the module, install module, remove module, etc.

What to Hand In

There is nothing to hand in for this lab. If you attain the lab, you will get full score for this lab.