Student ID: 406410035
Name: 秦紫頤
E-mail: chinjoyce30@gmail.com

Lab Title: Linux Driver I

Lab Purpose:

Write a driver that can be mounted and unmounted. Getting to know the development of a driver and its' operation.

Lab Procedure:

1. Module Compilation and Testing
    a. Write a basic driver called hello.c
    b. Write a Makefile for hello.c
    c. Compile the hello module: make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf-
    d. Test if the module works
        i. Put hello.ko that generate in the previous step into the SD card
        ii. Mount the module: insmod hello.ko
        iii. Unmount the module: rmmod hello.ko

```
Welcome to Buildroot
buildroot login: root
# ls
demsg.txt  messages
# cd ..
# ls
bin             lib32           mnt             proc            tmp
dev             linuxrc         mydmesg.log     root            usr
etc             lost+found      mymessage       run             var
exp10           media           mytestsys.exe   sbin
lib             messages        opt             sys
# cd ecp10
-sh: cd: can't cd to ecp10: No such file or directory
# cd exp10/
# insmod hello.ko
[   83.465749] hello: loading out-of-tree module taints kernel.
[   83.475177] <1>I am the initial function!
# rmmod hello.ko
[   91.526253] <1>I am the exit function!
# $
```

2. Write a complete driver
    a. A complete driver: support read, write, unclocked_ioctl, open and release
    b. Use the Makefile in the last part to compile the module and generate the module
    c. Write a test file to test the driver: (fread, fwrite, fclose)
    d. Compile the test file: arm-linux-gnueabihf-gcc -static -g test.c -o test.exe
    e. Put the driver module and test.exe in the SD card and plug the SD card back into Pi
    f. Power on Pi
    g. Create device node: mknod /dev/demo c 60 0
    h. Mount the driver: insmod hello.ko

i. Execute the test program: ./test.exe
j. Unmount the driver: rmmod hello.ko

```
..........[   22.721804] Indeed it is in host mode hprt0 = 00001101
.[   22.921782] usb 1-1: device descriptor read/64, error -110
[   23.041835] Indeed it is in host mode hprt0 = 00001101
[   23.241772] usb 1-1: new high-speed USB device number 3 using dwc_otg
. timeout!
run-parts: /etc/network/if-pre-up.d/wait_iface: exit status 1
FAIL

Welcome to Buildroot
buildroot login: [   28.321799] Indeed it is in host mode hprt0 = 00001101
[   28.521774] usb 1-1: device descriptor read/64, error -110
[   43.761775] Indeed it is in host mode hprt0 = 00001101
[   43.961771] usb 1-1: device descriptor read/64, error -110
[   44.081799] usb usb1-port1: attempt power cycle

Welcome to Buildroot
buildroot login: root
# cd ..
# mknod /dev/demo c 60 0
# cd exp10/test/
# insmod hello.ko
[  129.854896] hello: loading out-of-tree module taints kernel.
[  129.864370] <1>DEMO: started
# ./test.exe
[  134.415413] device open
[  134.420960] device ioctl
[  134.426505] device read
[  134.431997] device close
# rmmod hello.ko
[  143.349614] <1>DEMO: removed
#
```

Problems and Discussions

- Q&A
  What are the macros MODULE_LICENSE(), MODULE_DESCRIPTION() and MODULE_AUTHOR() for in Linux driver?
  1. MODULE_LICENSE(): provide sufficient information whether the module is free software or proprietary for the kernel module loader and for userspace tools
  2. MODULE_DESCRIPTION(): used to describe what the module does
  3. MODULE_AUTHOR(): declares the module's author

- Discussions
  The first problem I encountered is in the second part of writing a complete driver module. I want to differentiate this module from the hello.c in the first part so I name this module **mydriver.c** and change the Makefile to **mydriver.o**, but I got an error while compiling. So I changed the name back to hello.c and it can compile properly. This really confused me, because I think I can name my driver whatever I want to,

  The second problem I encountered is that I forgot to add MODULE_LICENSE() in the complete driver in second part, so when I mount the module in Pi (insmod hello.ko) it appeared a warning. So I add **MODULE_LICENSE()** into my driver module and the warning disappeared.

The third problem I encountered is from experiment handout. Because it isn't clear enough so I thought I need to create a device node in my host system (Ubuntu). But actually it should be created in Pi. I didn't discover the problem because the device node did appear in Pi when I put test.exe and hello.ko into Pi but that is not the right device node (not compatible with Pi). I discovered this problem until I execute test.exe and find that the messages that supposed to show on the screen haven't shown. Then I check the /dev for the node demo and find that the color is different from the other device nodes. So I delete that node and create the same node on Pi and execute test.exe again and finally succeed.