

學號：405410117

姓名：郭紘安

Email：andy8766kuo@gmail.com

實驗名稱：Device driver (I)

實驗目的：

- 寫一個 Driver 可以讓 kernel 掛載和卸載
- 了解 Driver 的開發過程和運作

實驗步驟：

➤ Part 1 - module 的編譯和測試

Step 1: Hello-World 模組的形態

```
hello.c (86 GB Volume ~/disk/Lab10-1) - gedit
Open [?]

#include <linux/init.h>
#include <linux/module.h>
#include <linux/kernel.h>

MODULE_LICENSE ("Dual BSD/GPL");

static int demo_init(void) {
    printk("<1>I am the initial function!\n");
    return 0;
}

static void demo_exit(void) {
    printk("<1>I am the exit function!\n");
}

module_init(demo_init);
module_exit(demo_exit);
```

Step 2: module 的 makefile

```
Makefile (86 GB Volume ~/disk/Lab10-1) - gedit
Open [?]

obj-m := hello.o
all:
    make -C /home/andy/disk/buildroot/output/build/linux-custom M=$(PWD) modules
clean:
    make -C /home/andy/disk/buildroot/output/build/linux-custom M=$(PWD) clean
```

Step 3: make

```
andy@ubuntu: ~/disk/Lab10-1
andy@ubuntu:~/disk/Lab10-1$ make ARCH=arm CROSS_COMPILE=arm-linux-gnueabi-
make -C /home/andy/disk/buildroot/output/build/linux-custom M=/home/andy/disk/Lab10-1 modules
make[1]: Entering directory '/home/andy/disk/buildroot/output/build/linux-custom'
Building modules, stage 2.
MODPOST 1 modules
make[1]: Leaving directory '/home/andy/disk/buildroot/output/build/linux-custom'
andy@ubuntu:~/disk/Lab10-1$
```

Step 4: insmod demo

```
# insmod hello.ko
[ 304.425877] hello: loading out-of-tree module taints kernel.
[ 304.435301] <1>I am the initial function!
#
```

學號：405410117

姓名：郭紘安

Email：andy8766kuo@gmail.com

Step 5:rmmod demo

```
# rmmod hello.ko
[ 351.464655] <1>I am the exit function!
#
```

➤ Part 2 - 完整 Driver 的型態

Step 1: 建立完整 Driver

```
#include <linux/init.h>
#include <linux/module.h>
#include <linux/fs.h>

static int example_open(struct inode *inode, struct file *filp) {
    printk("<1>EXAMPLE: open\n");
    return 0;
}

static int example_close(struct inode *inode, struct file *filp) {
    printk("<1>EXAMPLE: close\n");
    return 0;
}

static ssize_t example_read(struct file *filp, char *buf, size_t size, loff_t *f_pos) {
    printk("<1>EXAMPLE: read (size=%zu)\n", size);
    return 0;
}

static ssize_t example_write(struct file *filp, const char *buf, size_t size, loff_t *f_pos) {
    printk("<1>EXAMPLE: write (size=%zu)\n", size);
    return size;
}

static struct file_operations example_fops = {
    .open = example_open,
    .release = example_close,
    .read = example_read,
    .write = example_write,
};
MODULE_LICENSE("Dual BSD/GPL");

#define EXAMPLE_MAJOR 60
#define EXAMPLE_NAME "example"

static int example_init(void) {
    int result;
    printk("<1>EXAMPLE: init\n");

    /* Register character device */
    result = register_chrdev(EXAMPLE_MAJOR, EXAMPLE_NAME, &example_fops);
    if (result < 0) {
        printk("<1>EXAMPLE: Failed to register character device\n");
        return result;
    }

    return 0;
}

static void example_exit(void) {
    printk("<1>EXAMPLE: exit\n");

    /* Unregister character device */
    unregister_chrdev(EXAMPLE_MAJOR, EXAMPLE_NAME);
}

module_init(example_init);
module_exit(example_exit);
```

學號：405410117

姓名：郭紘安

Email：andy8766kuo@gmail.com

Step 2: module 的 makefile

```
Makefile (86 GB Volume ~/disk/Lab10-1) - gedit
Open  [icon]

obj-m := hello.o
all:
    make -C /home/andy/disk/buildroot/output/build/linux-custom M=$(PWD) modules
clean:
    make -C /home/andy/disk/buildroot/output/build/linux-custom M=$(PWD) clean
```

Step 3: make

```
andy@ubuntu:~/disk/Lab10-1$ make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf-
make -C /home/andy/disk/buildroot/output/build/linux-custom M=/home/andy/disk/Lab10-1 modules
make[1]: Entering directory '/home/andy/disk/buildroot/output/build/linux-custom'
CC [M] /home/andy/disk/Lab10-1/hello.o
Building modules, stage 2.
MODPOST 1 modules
CC /home/andy/disk/Lab10-1/hello.mod.o
LD [M] /home/andy/disk/Lab10-1/hello.ko
make[1]: Leaving directory '/home/andy/disk/buildroot/output/build/linux-custom'
andy@ubuntu:~/disk/Lab10-1$
```

Step 4: 建立並編譯一個測試檔

```
test.c (86 GB Volume ~/disk/Lab10-1) - gedit
Open  [icon]

#include <stdio.h>

int main(){
    char buf[512];
    FILE *fp = fopen("/dev/demo", "w+");
    if (fp == NULL) {
        printf("cannot open device!\n");
        return 0;
    }
    fread(buf, sizeof(buf), 1, fp);
    fwrite(buf, sizeof(buf), 1, fp);
    fclose(fp);
    return 0;
}
```

```
andy@ubuntu:~/disk/Lab10-1$ arm-linux-gnueabihf-gcc -static -g test.c -o test
andy@ubuntu:~/disk/Lab10-1$
```

Step 5: 測試結果

```
# insmod hello.ko
[ 44.801620] hello: loading out-of-tree module taints kernel.
[ 44.811232] <1>EXAMPLE: init
# mknod /dev/demo c 60 0
# chmod 777 ./test
# ./test
[ 124.642762] <1>EXAMPLE: open
[ 124.648959] <1>EXAMPLE: read (size=4096)
[ 124.656365] <1>EXAMPLE: write (size=512)
[ 124.663428] <1>EXAMPLE: close
#
```

學號：405410117

姓名：郭紘安

Email：andy8766kuo@gmail.com

問題與討論：

- 請問在撰寫 Linux driver 時，以下的巨集是做什麼用的？

`MODULE_LICENSE()`

`MODULE_DESCRIPTION()`

`MODULE_AUTHOR()`

Ans：

`MODULE_LICENSE()` 巨集用來宣告程式 License，

`MODULE_DESCRIPTION()` 用來描述此模組或者是 Driver 的功用跟簡介，

`MODULE_AUTHOR()` 用來定義此模組作者，

這些巨集都可以在 `linux/module.h` 裡找到，但是這些並非用於 Kernel 本身。

心得：

這次實驗讓我知道如何使用 `insmod` 來掛載自己建立的模組，以及使用 `dmesg` 來查看模組是否掛載成功，跟最後如何使用 `rmmod` 來卸載模組。