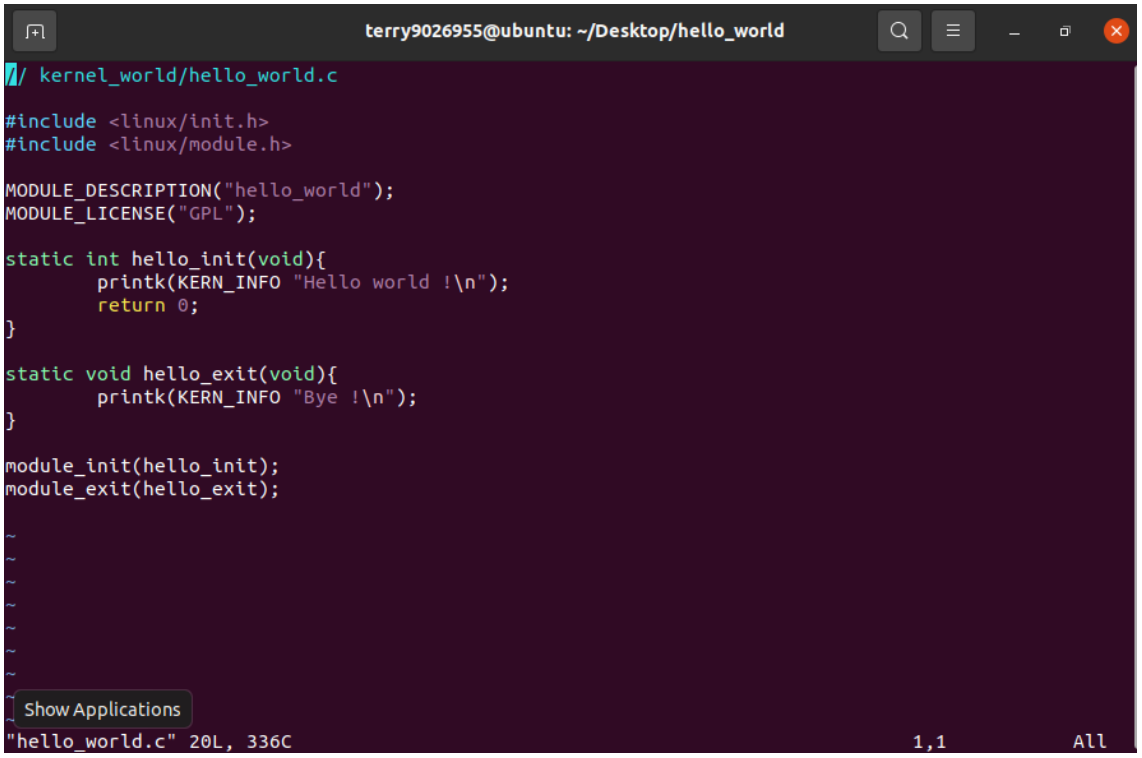
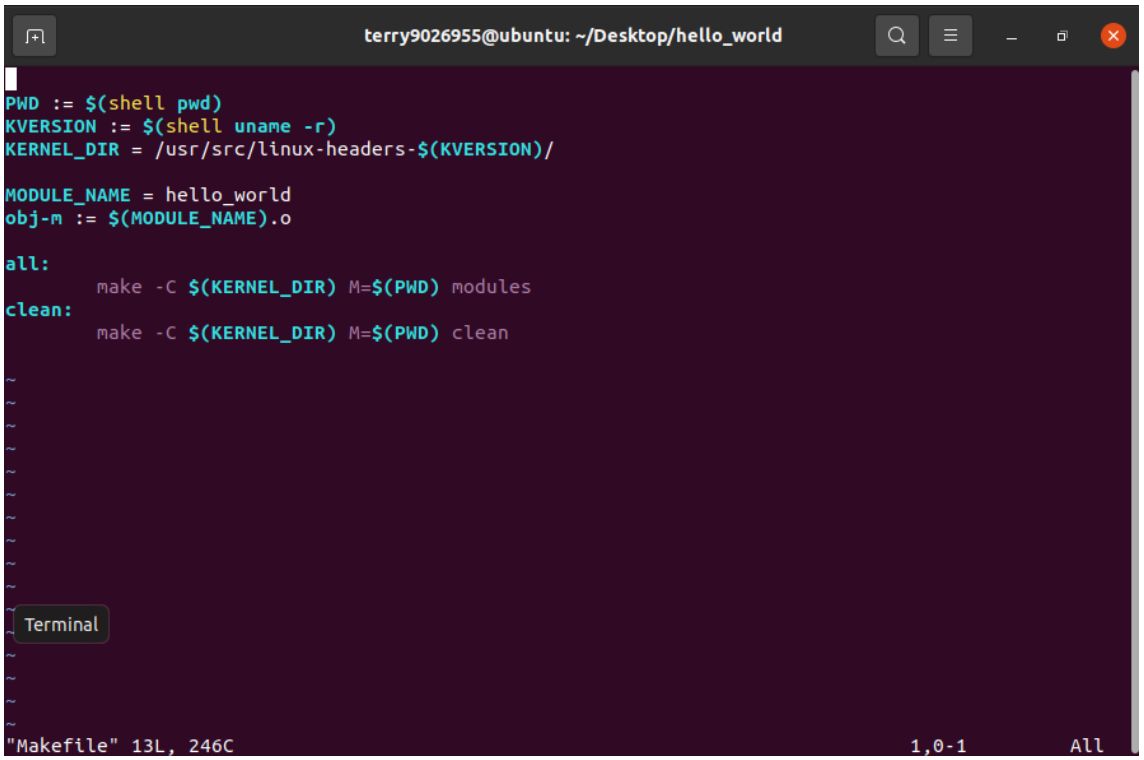
ES_HW01_110598040

陳廷豪

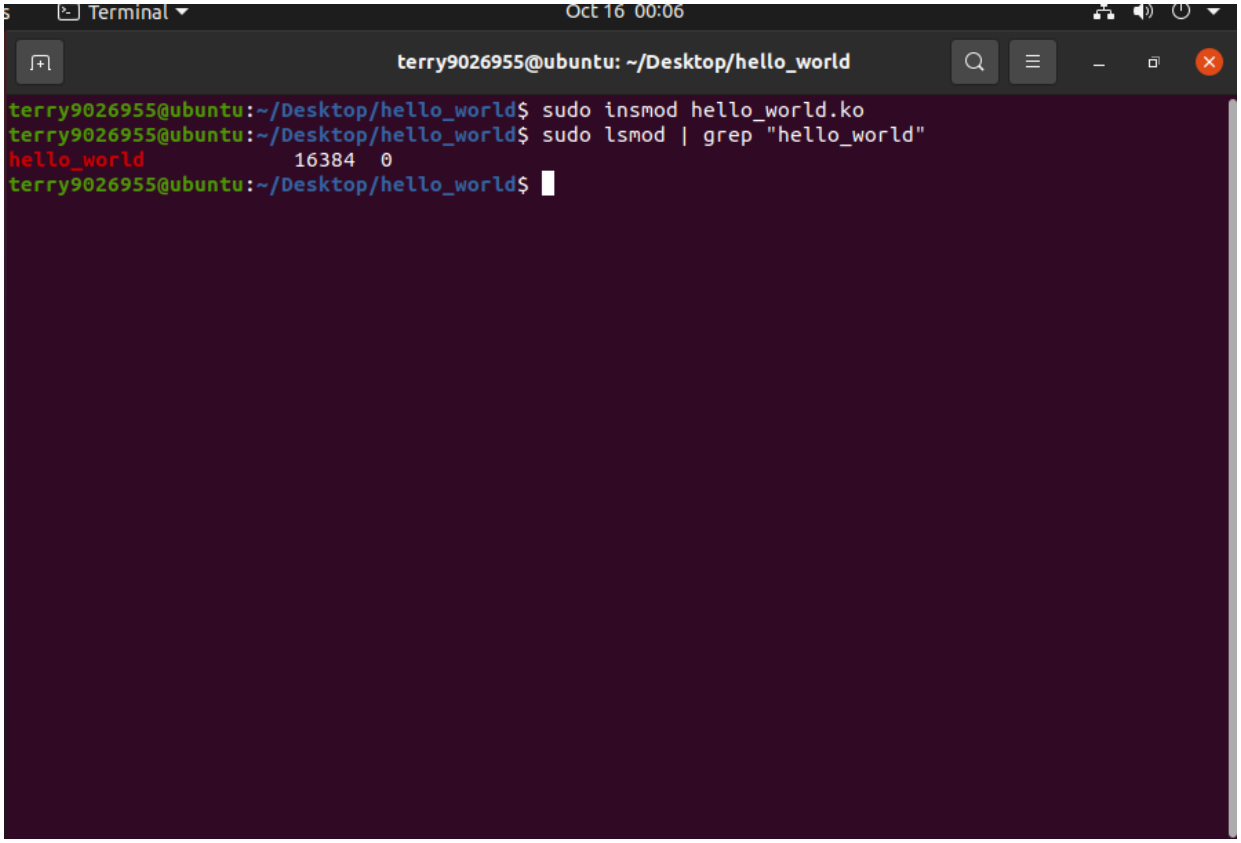
HW01_Part1:

Part 1：Your hello_world.c & Makefile





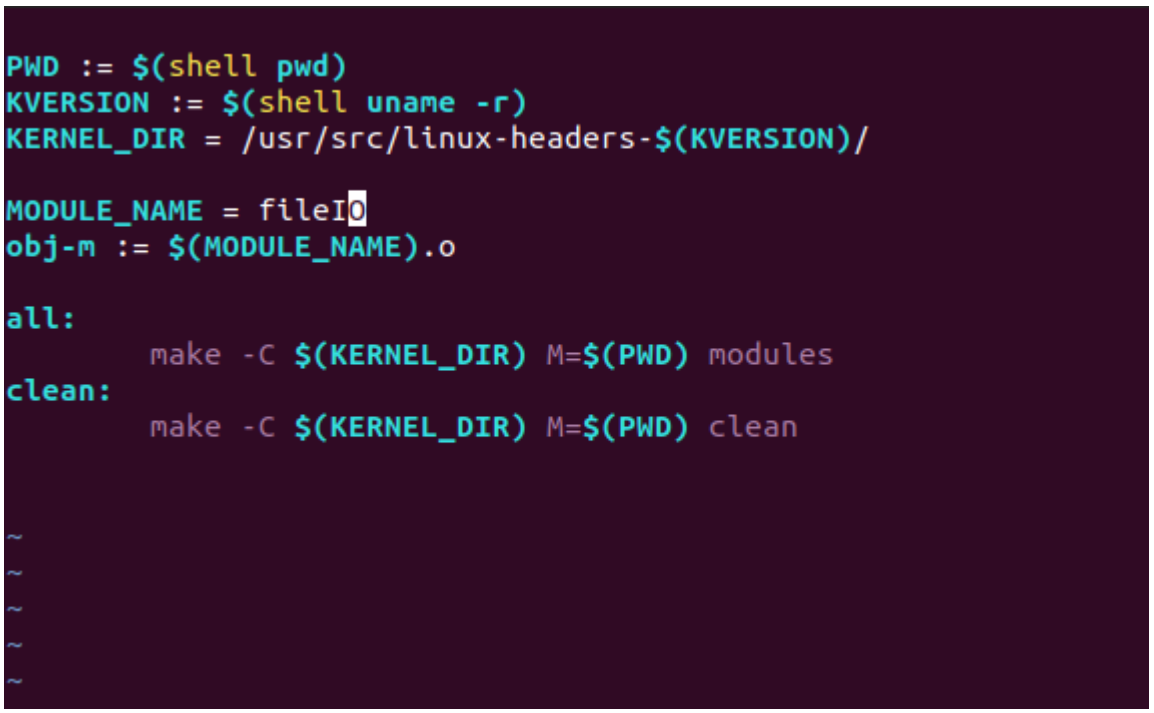
Part 2：Using lsmod to check you module by following slides of Week 4



Part 3：**Check the HelloWorld Message through demsg by following slides of Week 4**

HW01_Part2:

Part1: fileIO.c & Makefile



```
PWD := $(shell pwd)
KVERSION := $(shell uname -r)
KERNEL_DIR = /usr/src/linux-headers-$(KVERSION)/

MODULE_NAME = fileIO
obj-m := $(MODULE_NAME).o

all:
	make -C $(KERNEL_DIR) M=$(PWD) modules
clean:
	make -C $(KERNEL_DIR) M=$(PWD) clean
```

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

static char buf1[261];
static char buf2[261];
static int IO_init(void)
{
    int i = 0, j = 0, k = 0;
    struct file *fp;
    struct file *output;
    loff_t pos = 0;
    //Read the textfile and check file
    printk("readfile enter!\n");
    fp = filp_open("/home/terry9026955/Desktop/fileIO/input.txt", O_RDWR, 0);
    if (IS_ERR(fp)) {
        printk("create file error\n");
        return -1;
    }
    //Use kernel_read() instead of vfs_write or vfs_read
    kernel_read(fp, buf1, sizeof(buf1), &pos); //read fp's content into buf1
    printk("read: %s\n", buf1);
    //Reverse buf1's content and write into buf2
    while(buf1[i] != '\0'){
        i++;
    }
    for(j = i-1; j >= 0; j--){
        buf2[k] = buf1[j];
        k++;
    }
    printk("output: %s\n", buf2);

    //Write buf2 into a new textfile
    output = filp_open("/home/terry9026955/Desktop/fileIO/output.txt", O_RDWR, 0);
    pos = 0; //pos表示當前文件讀寫所處於的偏移位置
    kernel_write(output, buf2, sizeof(buf2), &pos);
    filp_close(fp, NULL);
    return 0;
}
static void IO_exit(void)
{
    printk(KERN_INFO "readfile  exit!\n");
}
module_init(IO_init);
module_exit(IO_exit);
MODULE_LICENSE("GPL");
```

Part2: Using lsmod to check you module

```
terry9026955@ubuntu:~/Desktop/fileIO$ sudo lsmod | grep "fileIO"
fileIO                 16384  0
terry9026955@ubuntu:~/Desktop/fileIO$
```

Part3: Check the fileIO Message through demsg

```
[52092.421959] readfile enter!
[52092.421982] read: abcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyz

[52092.421995] output:
                zyxwvutsrqponmlkjihgfedcbazyxwvutsrqponmlkjihgfedcbazyxwvutsrqponmlkjihgfedcbazyxwvutsrqponmlkjihgfedcbazyxwvutsrqponmlkjihgfedcbazyxwvutsrqponmlkjihgfedcbazyxwvutsrqponmlkjihgfedcbazyxwvutsrqponmlkjihgfedcbazyxwvutsrqponmlkjihgfedcba
[52095.254254] readfile  exit!
```