**微算機系統實習**

實驗五之二

日期： 2021.05.30

1. 組別與組員名單：

-第十三組

-資工二 108590002 陸永強

-資工二 108590016 陳琴蓮

1. 實驗步驟截圖與說明：

項目一：創建file\_operations資料結構 & 項目二：掛載與下載驅動程式

//

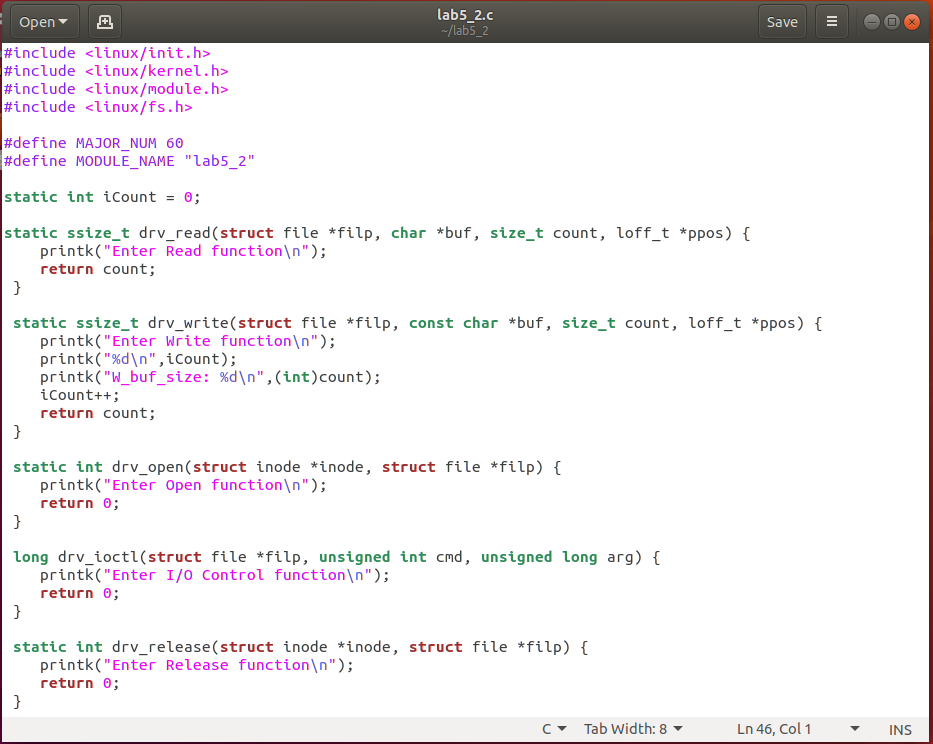
static ssize\_t drv\_read()：印出Read的訊息

static ssize\_t drv\_write()：印出Write的訊息

static int drv\_open()：印出Open的訊息

long drv\_ioctl()：印出I/O Control的訊息

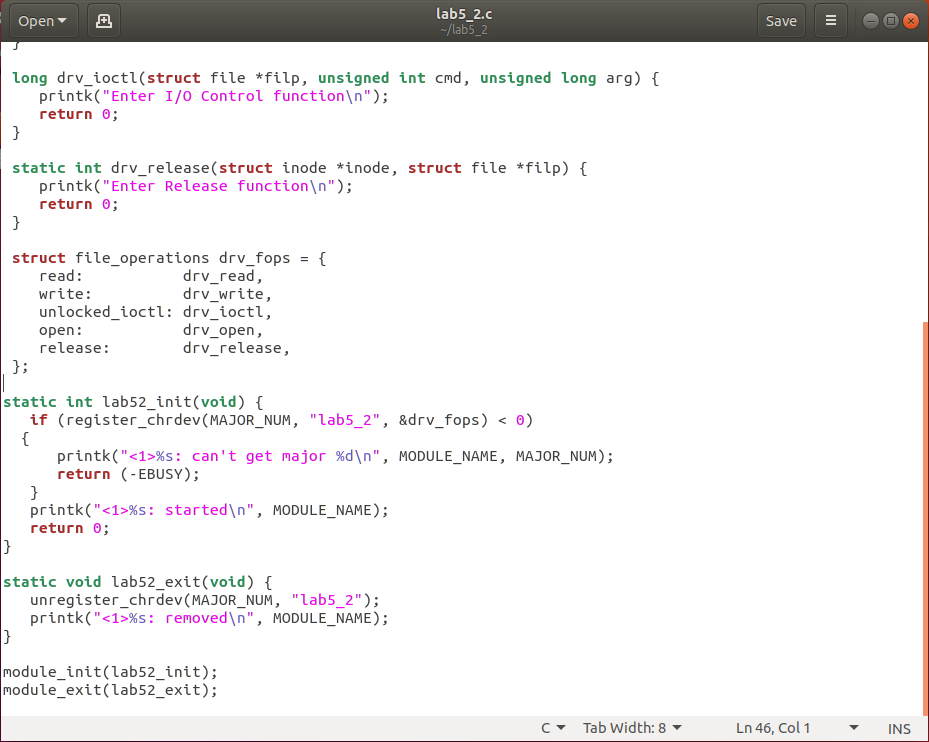
static int drv\_release()：印出Release的訊息



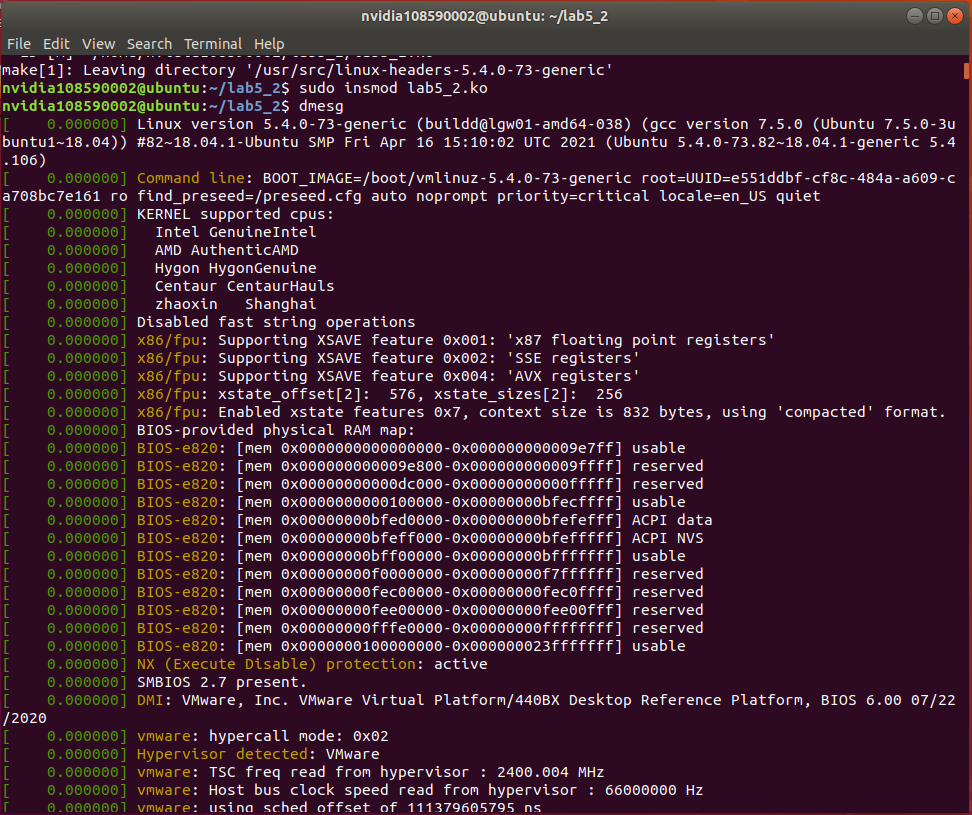
struct file\_operations drv\_fops = {} ：file\_operations的資料結構

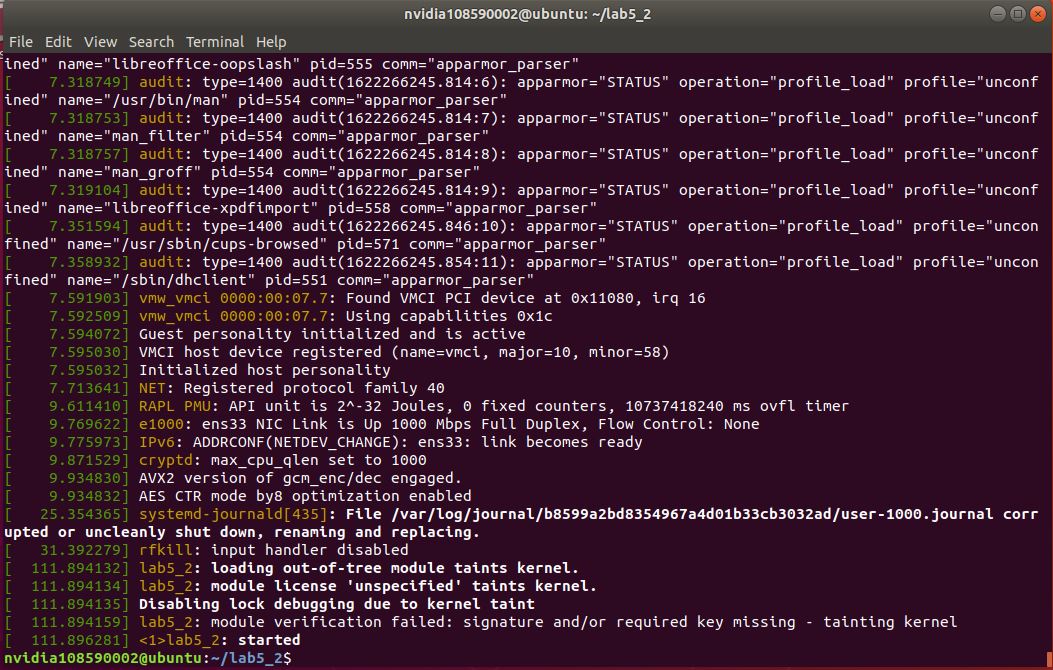
static int lab52\_init(void)：初始驅動程式

static void lab52\_exit(void)：離開驅動程式



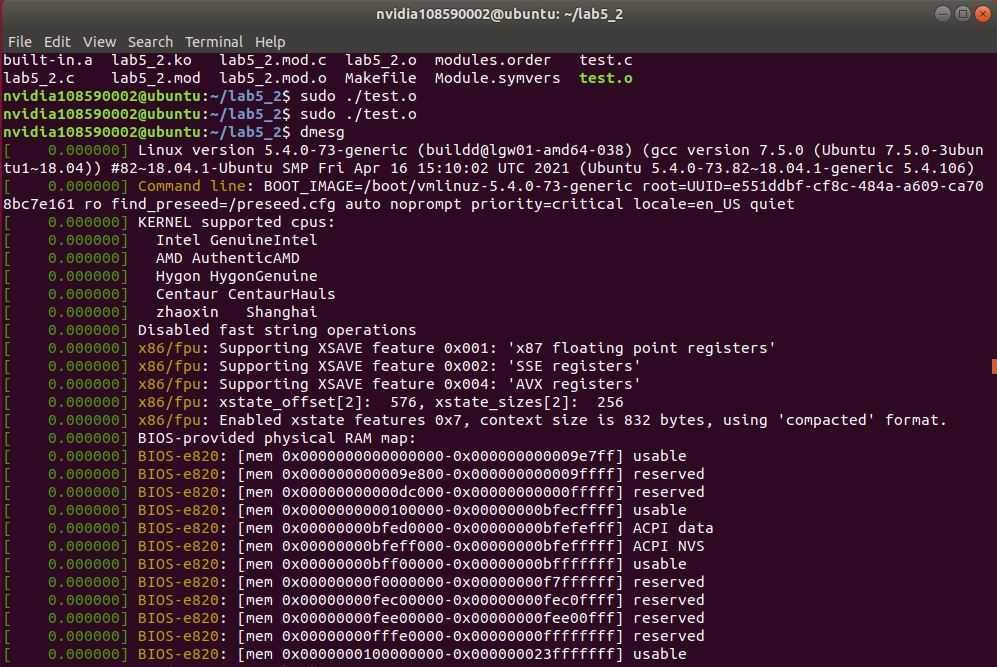
以下為執行結果

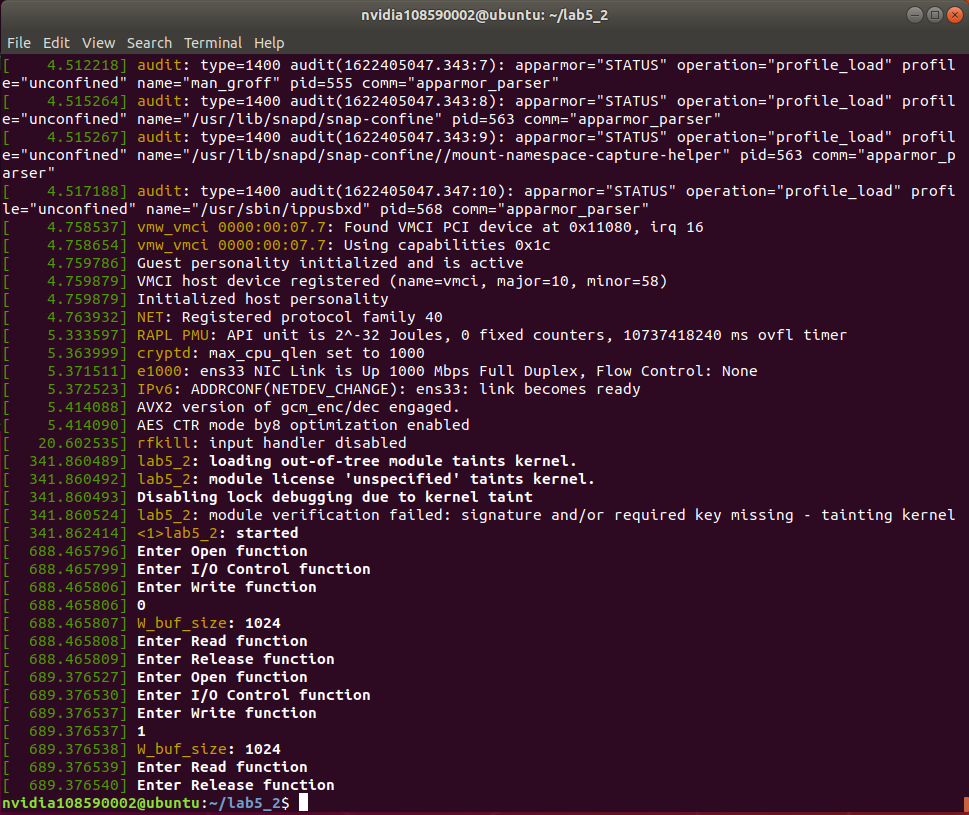
//掛載與初始驅動程式



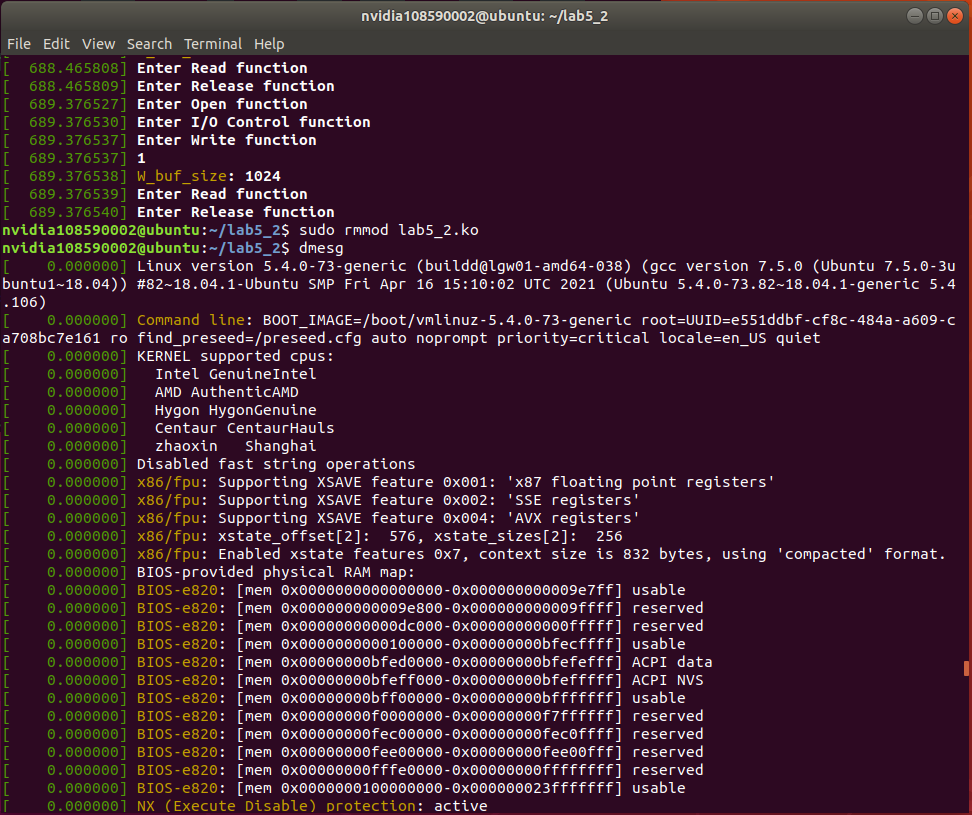
//執行兩次test.o

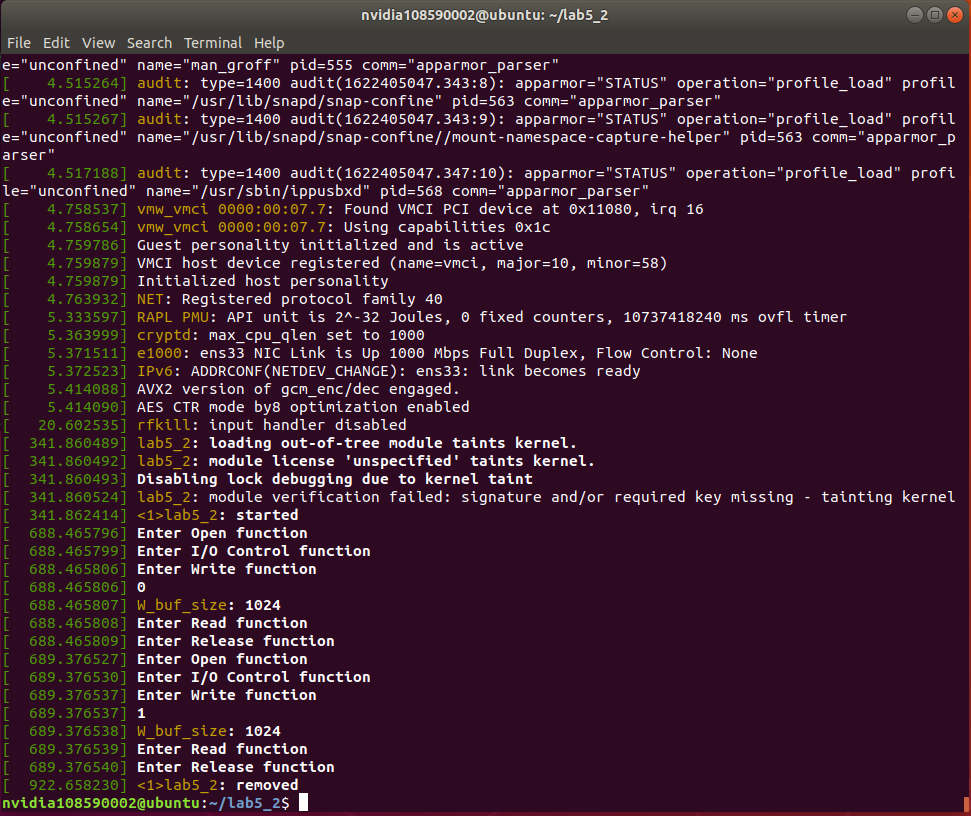
//看得出來有讀取成功test.o的資料





//卸載與離開驅動程式



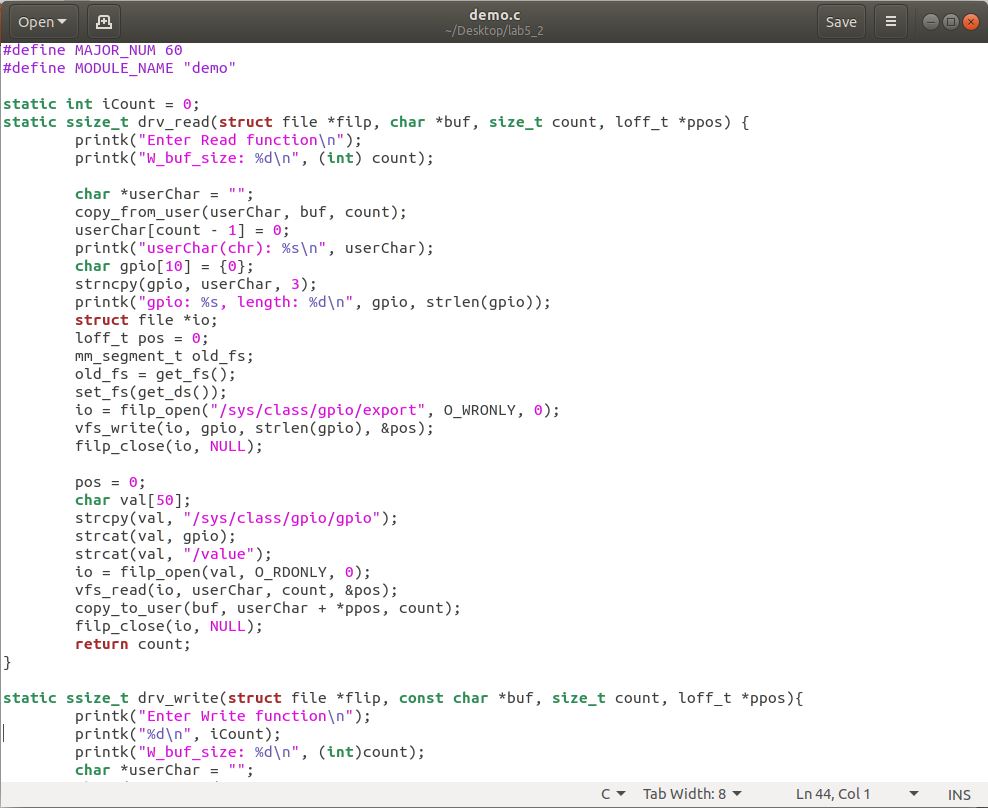


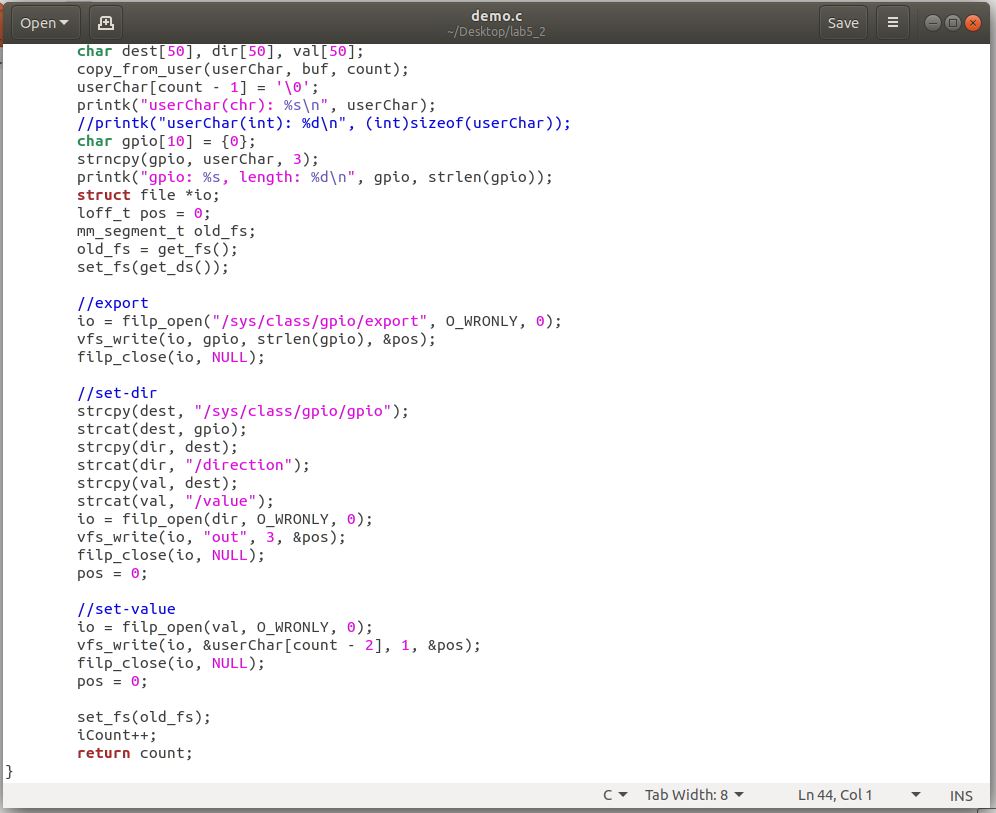
項目三：控制裝置驅動

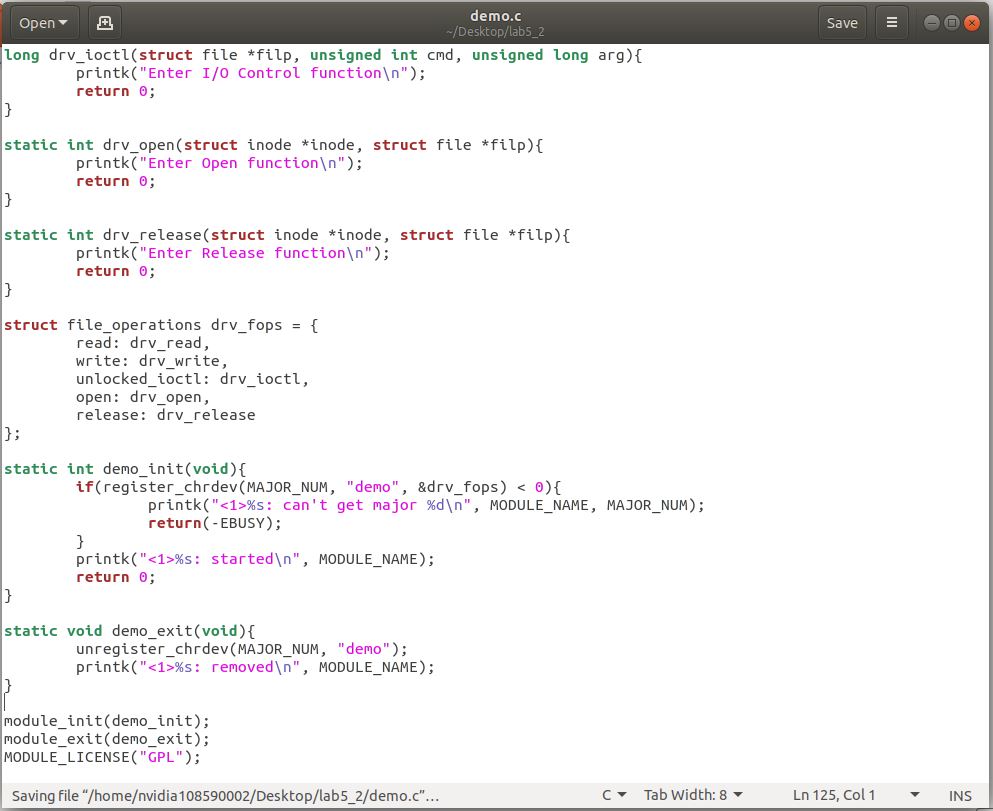
項目四：讀取裝置驅動狀態

static ssize\_t drv\_read()：讀取輸入資料

static ssize\_t drv\_write()：設定export，set\_dir和set\_value，寫入輸入資料

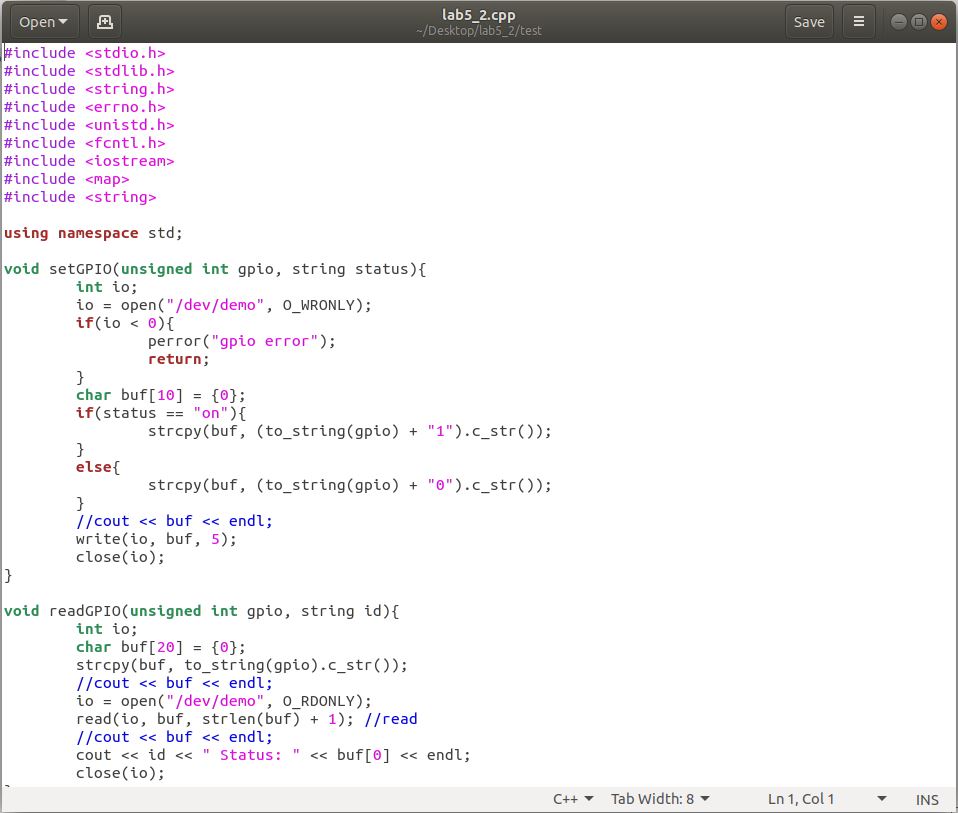




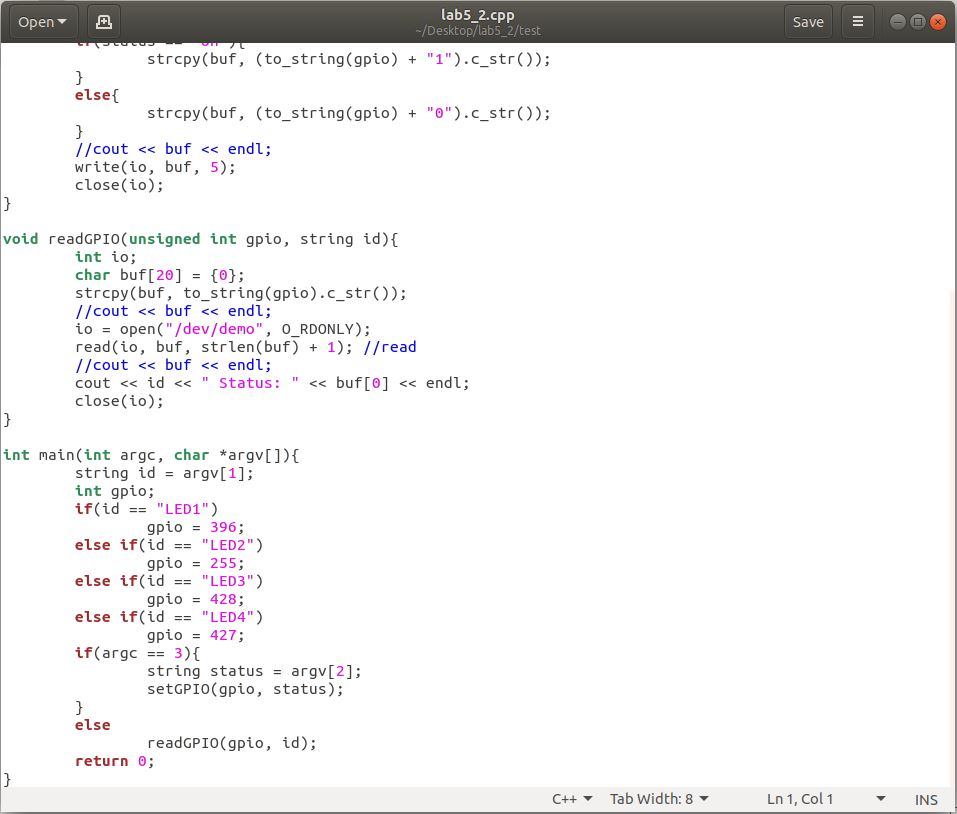


void setGPIO()：根據輸入LED的id與狀態來設置該腳位的狀態

void readGPIO()：根據輸入LED的id來讀取該腳位的狀態



int main()：設定LED燈的腳位，然後判斷是要控制裝置驅動(setGPIO)或讀取裝置驅動狀態(readGPIO)



影片連結：<https://youtu.be/HPrz_VaZIwI> (有說明操作步驟)

1. 組員貢獻度比例（組員%數加總必須等於100%）

陸永強：50%

陳琴蓮：50%

全部組員合起來100%

1. 實驗心得：

陸永強：

這次實驗因為疫情關係，我們採用遠距一起完成這次實驗。板子由組員保管，所以我負責前兩個項目，組員負責後兩個項目。一開始我沒有建立裝置檔案，所以就算初始驅動程式碼和執行test.o也沒辦法被讀取到資料，補上就可以了。然後執行驅動程式碼執行到一半就當機，重新執行也會這樣。後來花了一些時間和同學討論才知道是write函式裡的printk(“buf: %s\n”,buf)這樣行要拿掉，後來就執行成功了。

陳琴蓮：

這次實驗因為疫情的緣故，必須遠距，因為不敢到學校拿網路線，最後就自己買了一條，好不容易買到之後，又研究了很久才研究出怎麼連網路，但其實並沒有我想像中的困難，跟上課的方式一模一樣。這次實驗傳出了很多災情，我印象最深刻的是，在執行./Lab5 LEDX的時候會一直出現”Segmentation fault (core dump)”的錯誤，為了解決這問題真的是花了很多時間，後來在看dmesg的時候發現吃到的長度好像不太對，之後想了想發現好像是因為buf的長度不夠，給它多+1就成功解決了。在編譯lab5.cpp時，一開始是用gcc來編譯，結果出現了一個錯誤，上網找之後發現好像要利用g++來編譯，改為g++後就能成功編譯了。