**作業系統**

**HW1**

**Project程式作業**

**資工三 108590002 陸永強**

**資工三 108590008 林聖祐**

**資工三 108590450 羅傑**

**資工三 108590452 林峻霆**

**Chapter 2 Project**

**Part I:**

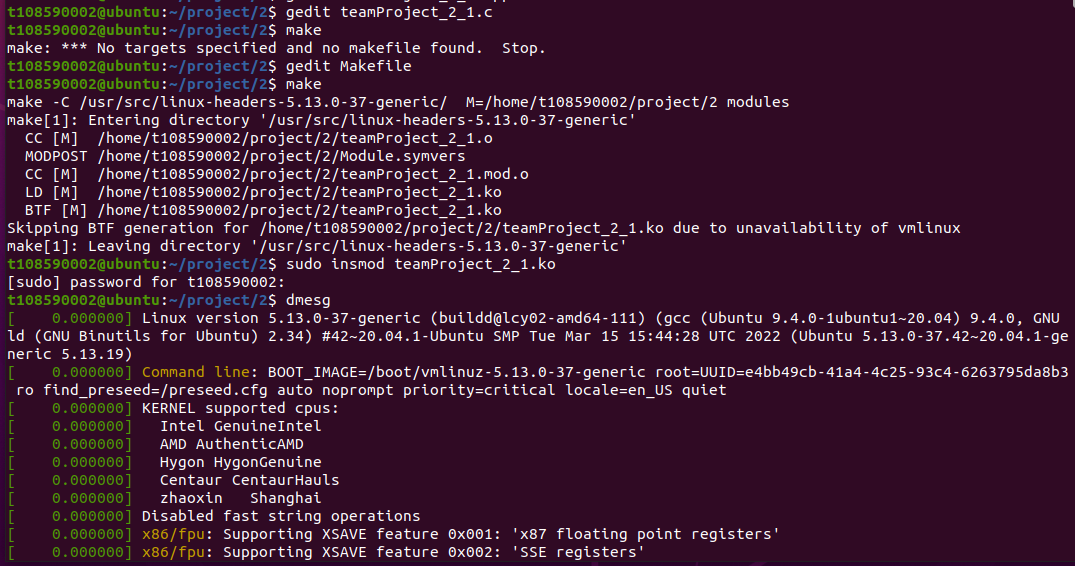
步驟說明：

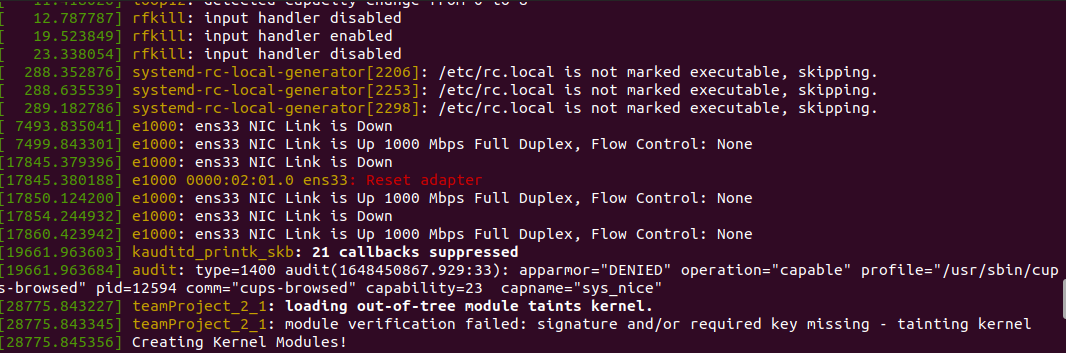
1. 在Linux環境安裝gcc：

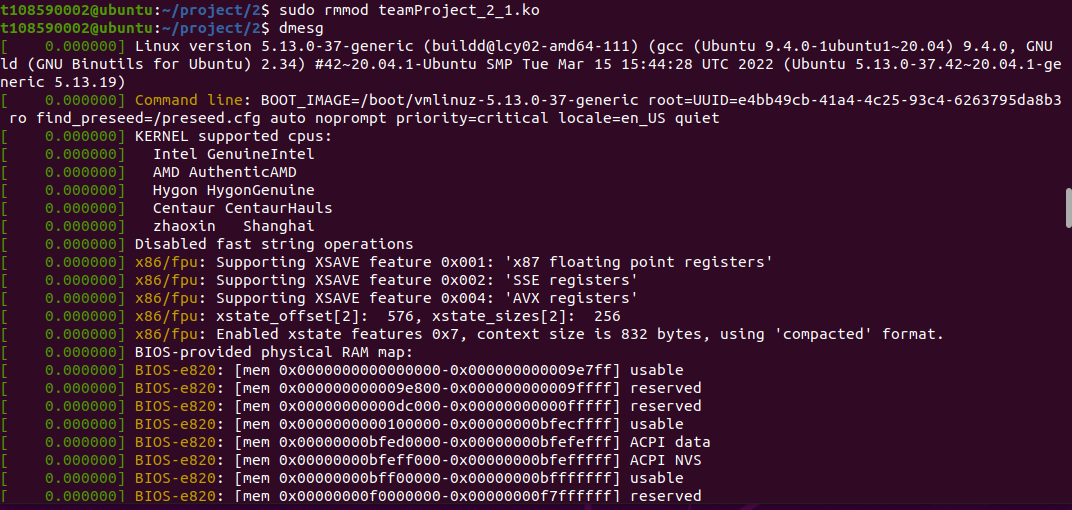
sudo apt-get install build-essential

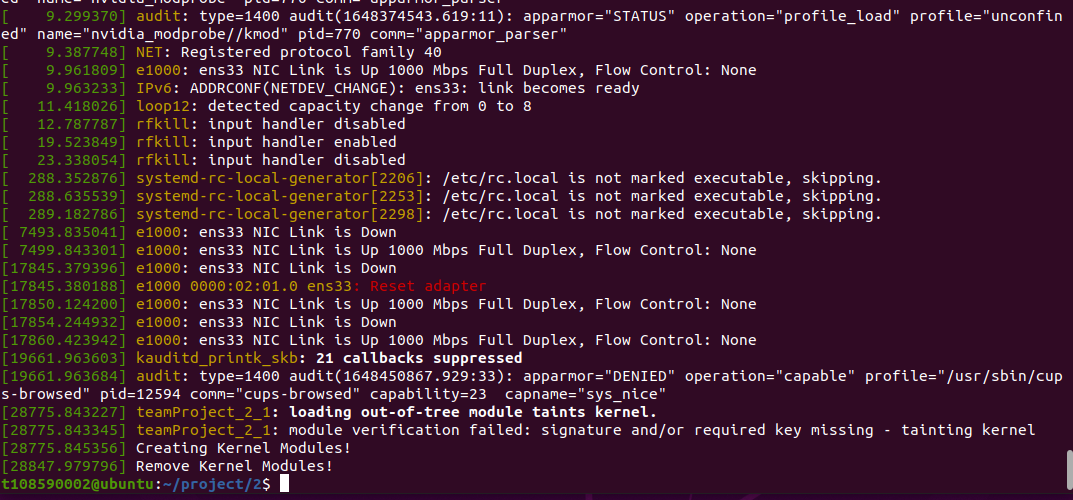
1. 使用c撰寫程式 (檔名：teamProject\_2\_1.c)
2. 撰寫Makefile
3. 輸入make執行Makefile
4. 輸入sudo insmod teamProject\_2\_1.ko來create kernel modules
5. 輸入dmesg指令來檢查buffer訊息，以確保kernel有被create成功
6. 輸入sudo rmsmod teamProject\_2\_1.ko來remove kernel modules
7. 輸入dmesg指令來檢查buffer訊息，以確保kernel有被remove成功

執行結果：







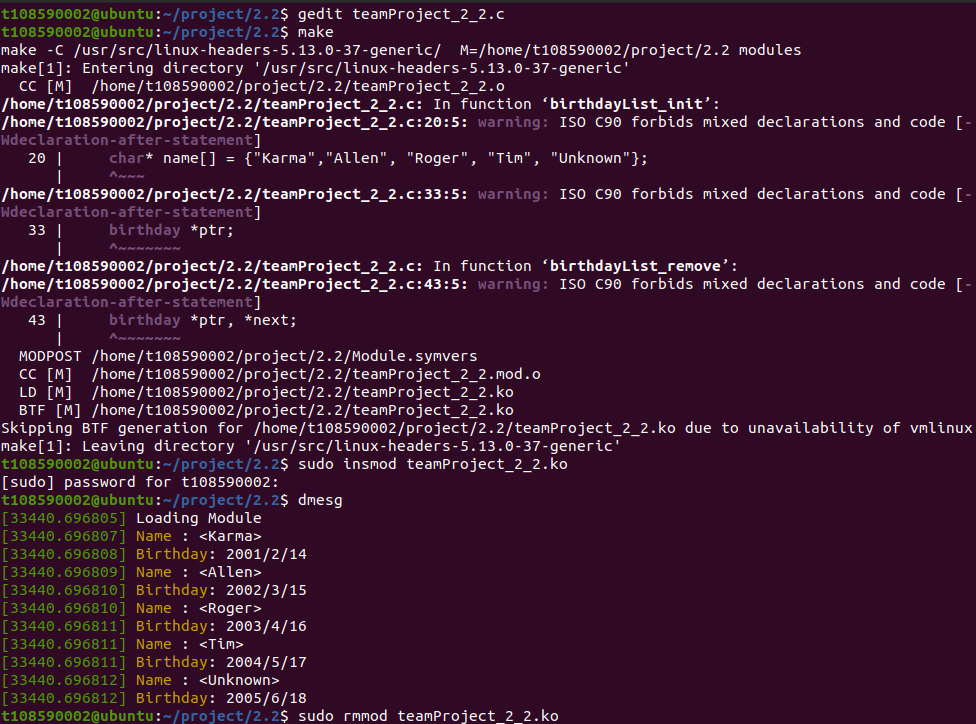


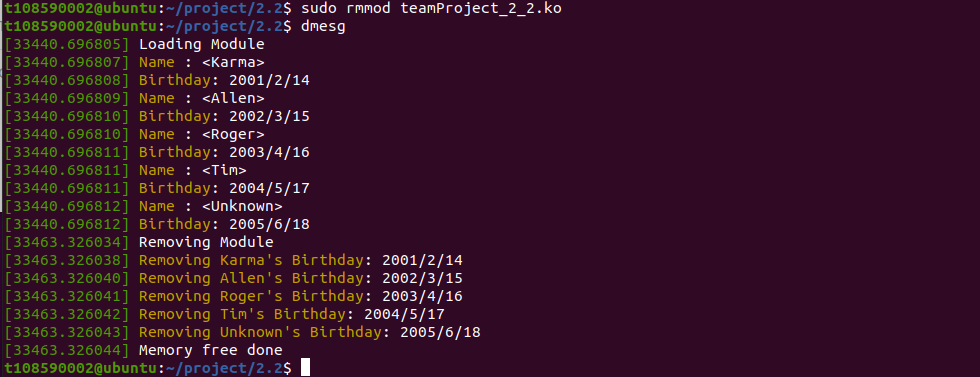
**Part II:**

步驟說明：

1. 使用c撰寫程式 (檔名：teamProject\_2\_2.c)
2. 撰寫Makefile
3. 輸入make執行Makefile
4. 輸入sudo insmod teamProject\_2\_2.ko來create kernel的data structures
5. 輸入dmesg指令來檢查buffer訊息，以確保kernel的data structures，包含5個struct\_birthday的elements都有被create成功
6. 輸入sudo rmsmod teamProject\_2\_2.ko來remove kernel的data structures
7. 輸入dmesg指令來檢查buffer訊息，以確保kernel的data structures，包含5個struct\_birthday的elements都有被remove成功，將記憶體return給kernel

執行結果：





**Chapter 3 Project 1**

**Part I & II:**

步驟說明：

1. 使用c撰寫程式 (檔名：teamProject\_3.1.c)
2. 撰寫Makefile
3. 輸入make使用Makefile編譯
4. 輸入 ./project\_3.1即開始執行程式
5. 進入child process進行分叉並執行指令
6. 輸入Linux Kernel Module的指令，ex: cd、ls、cat、touch、rm等等
7. 檢查指令是否預期行為
8. 輸入我們自定義的指令，以便提供允許使用者訪問最近輸入的指令的歷史記錄。

history!：查找所有歷程輸入指令

-h [N]：查找第前N個輸入指令

執行結果：

Ex:

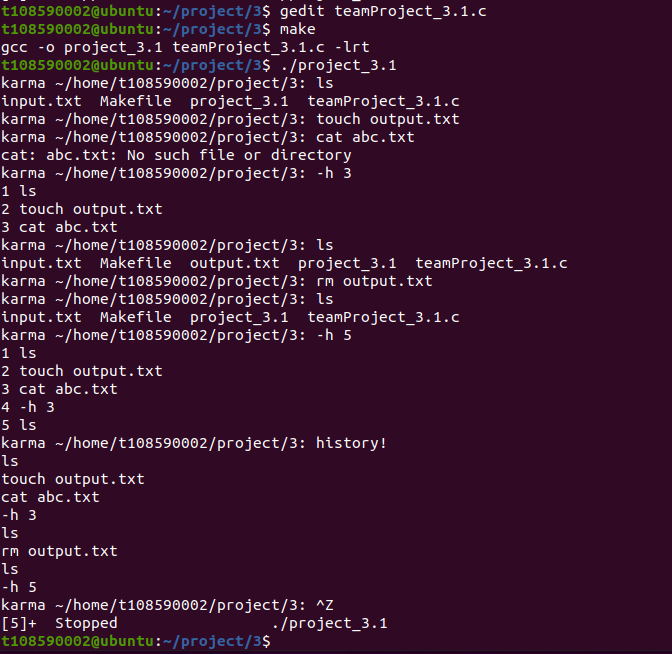
先輸入ls、touch、cat

輸入 -h 3會查詢到最近3個指令：ls、touch、cat

再輸入ls、rm、ls

輸入 -h 5會查詢到最近5個指令：ls、touch、cat、-h 3、ls (-h 3也算是一個指令)

最後輸入 history! 會查詢到所有歷程輸入的指令



**工作分配 (100%)：**

陸永強 (25%)：蒐集資料、報告製作

林聖祐 (25%)：程式碼撰寫、Debug

羅傑 (25%)：程式碼撰寫、報告製作

林峻霆 (25%)：蒐集資料、Debug