組合語言Midterm Project

108學年度第1學期

老師：朱守禮老師

系級：資訊二甲

組別：第14組

學生：

10727124 劉宇廷

10727131 陳彥綸

10727153 呂宜鴻

一、背景

利用Assembly Language完成一個能列印組別、組員姓名、學號與學號總和的

Assembly Program。

二、方法

‧程式說明：

1. name：印出組別及組員姓名。

2. ID：輸入組員學號且再輸入p後，將組員學號及學號總和列印出來。

3. main：分別呼叫name、ID兩個函數，完成各函數原始功能，最後整合兩

函數的資料，印出完整的組別、組員姓名、學號與學號總和。

‧設計重點說明：

name：

1.分別配置記憶體給team、one、two、three四個變數。

2.分別存入組別、組員一姓名、組員二姓名、組員三姓名。

3.將變數存入r0，並且使用bl printf印出來，從team開始重複以上步驟直到

將three印出來。

4.分別將team、one、two、three存入r8、r9、r10、r11，方便能在main

裡使用。

ID：

1.分別配置記憶體給a、b、c、w、ans五個變數。

2.將格式字串"%d"存入r0，並將a、b、c分別存入r1，利用bl scanf將三

個組員的學號分別讀入a、b、c。

3.將格式字串"%s"存入r0，並將w存入r1，利用相同指令bl scanf 讀入，

再利用cmp比較w與字元p，如果eq成立執行步驟4、5、6，反之ne

成立印出"error command"。

4.將格式字串"%d\n"存入r0，並且將變數a存入r1印出，變數b、c同

樣執行以上動作，將組員學號分行印出。

5.分別將a、b、c存入r1、r2、r3，利用add計算出r1+r2+r3的結果，並將

結果放入變數ans，代表學號總和。

6.分別將a、b、c、ans存入r4、r5、r6、r7，方便能在main裡印出

學號及學號總和。

main：

1.呼叫name及ID兩函數，執行各函數功能。

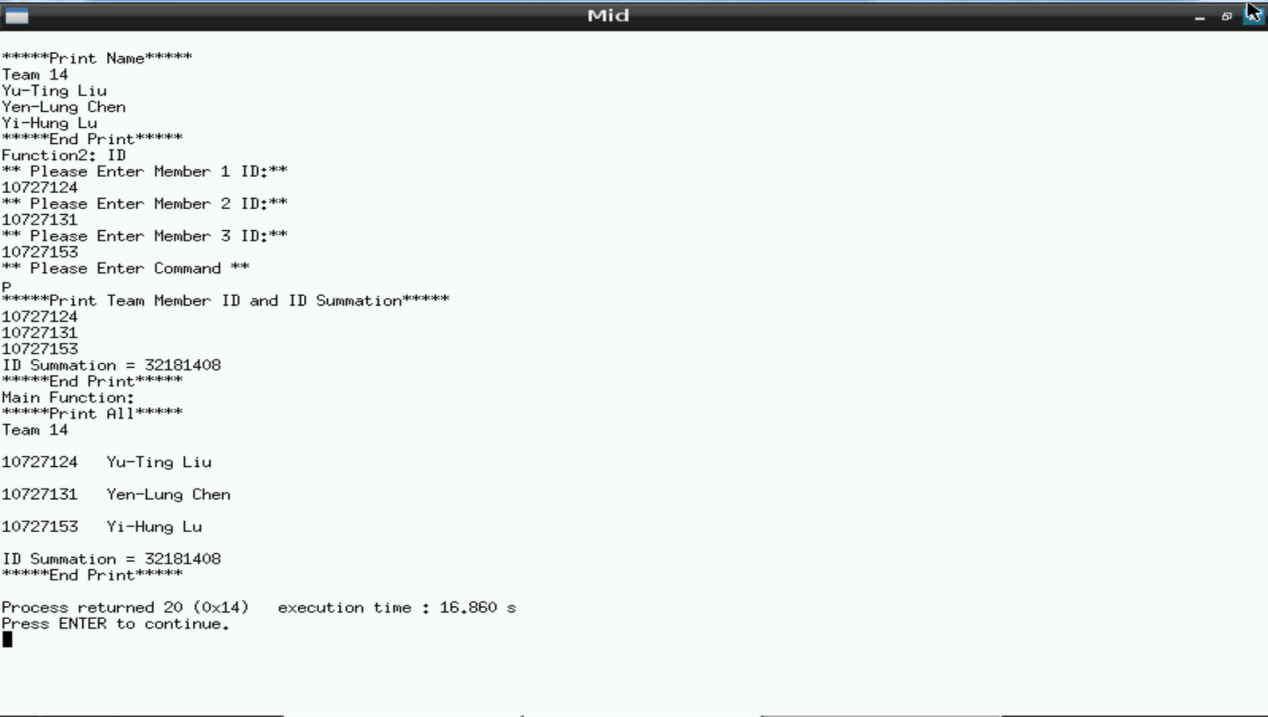
2.利用在執行name和ID時存入暫存器的資訊，依序放入r0並利用bl printf

將組別、組員姓名、學號與學號總和完整列印出來。

三、結果

‧程式執行結果

>>輸入學號 輸入p



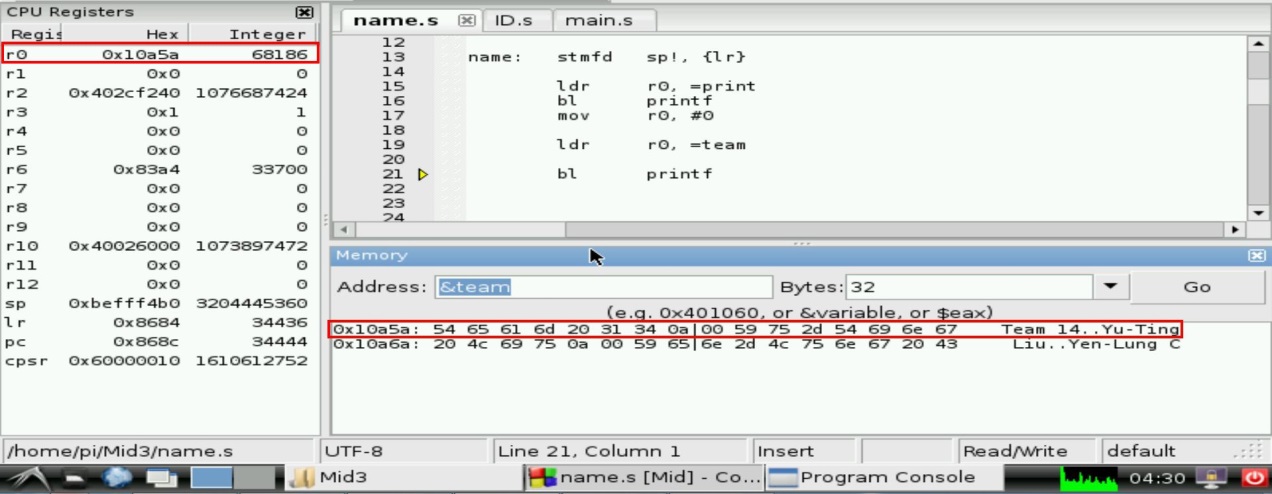
>>輸入學號 輸入Q(非p字元)



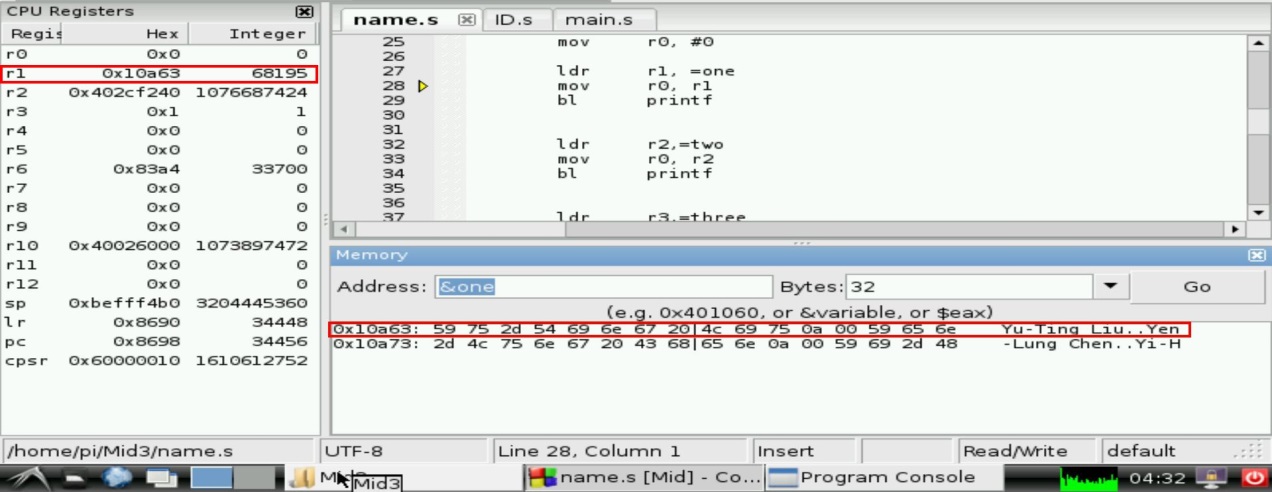
四、討論

‧name：

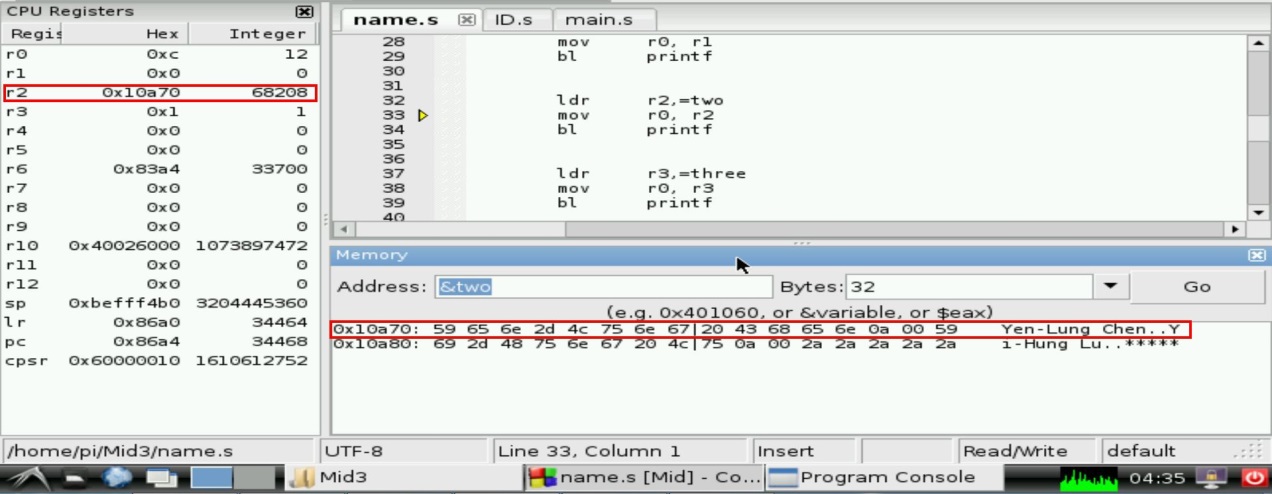
儲存組別之記憶體位址及內容：



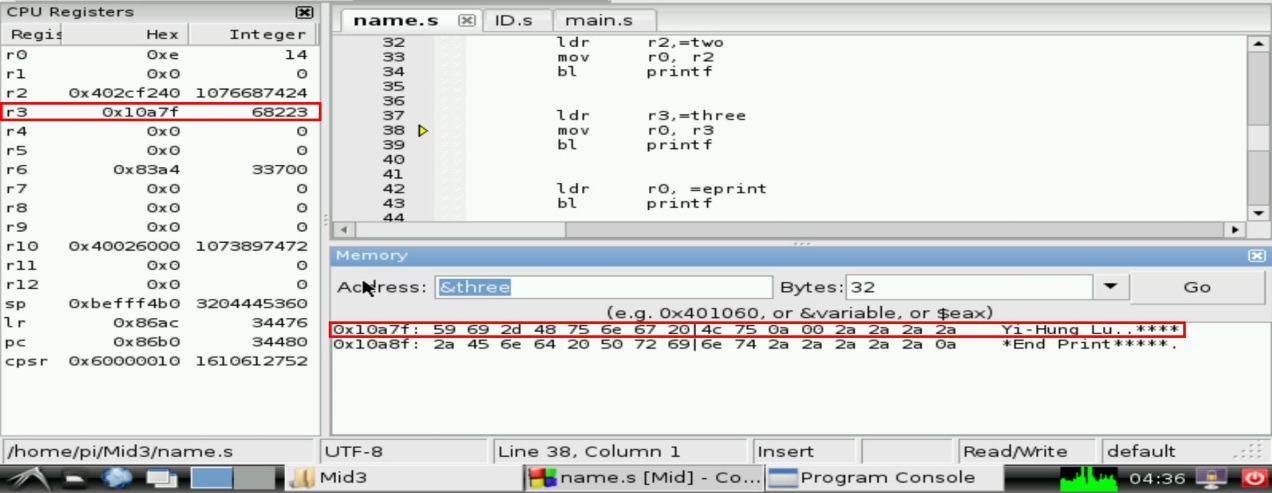
儲存組員一姓名之記憶體位址及內容：



儲存組員二姓名之記憶體位址及內容：

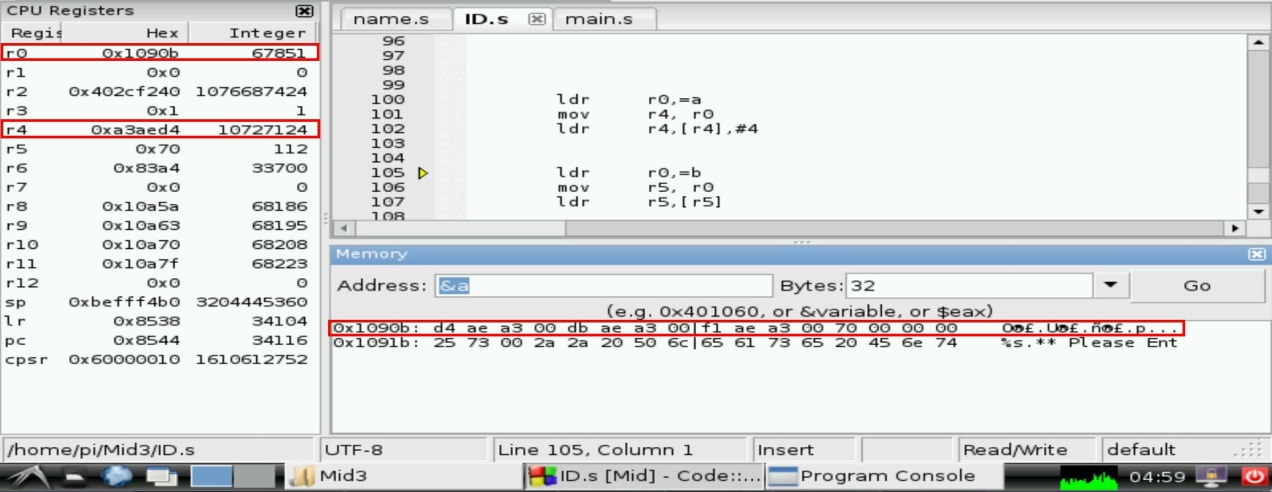


儲存組員三姓名之記憶體位址及內容：

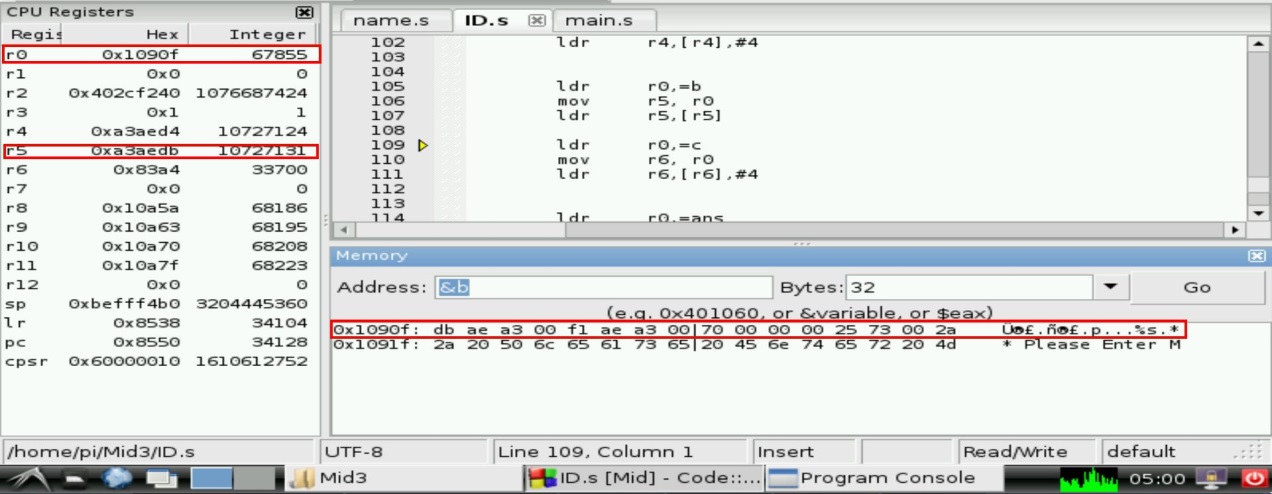


‧ID：

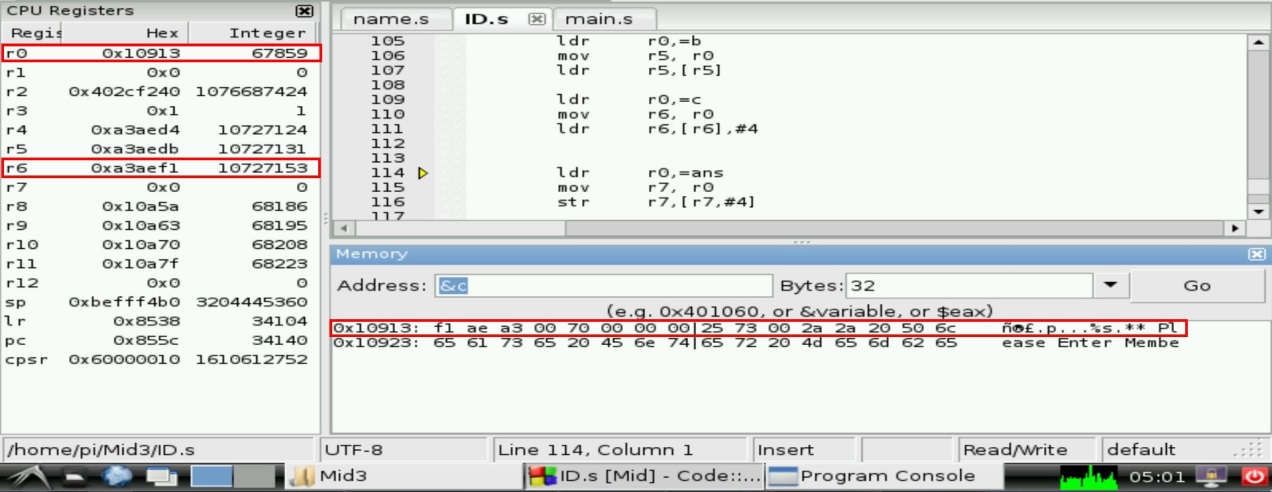
儲存組員一學號之記憶體位址及內容：



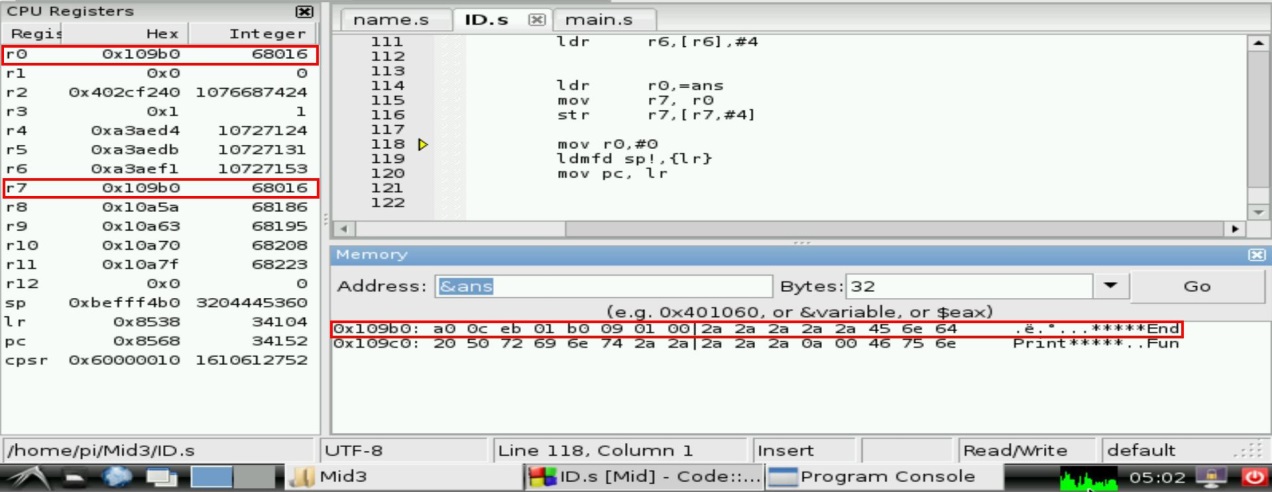
儲存組員二學號之記憶體位址及內容：



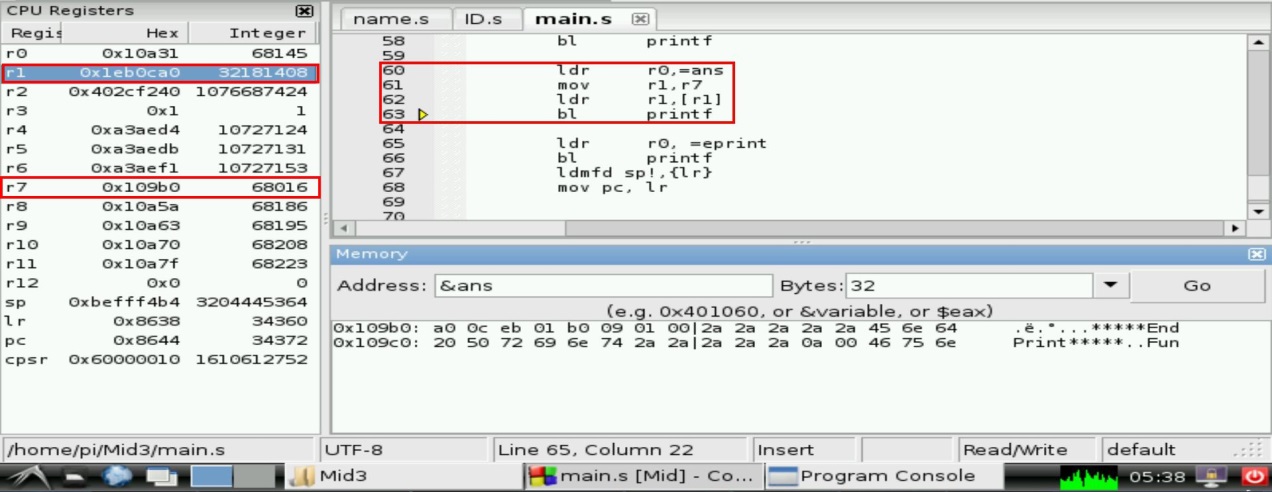
儲存組員三學號之記憶體位址及內容：



儲存學號總和之記憶體位址及內容：



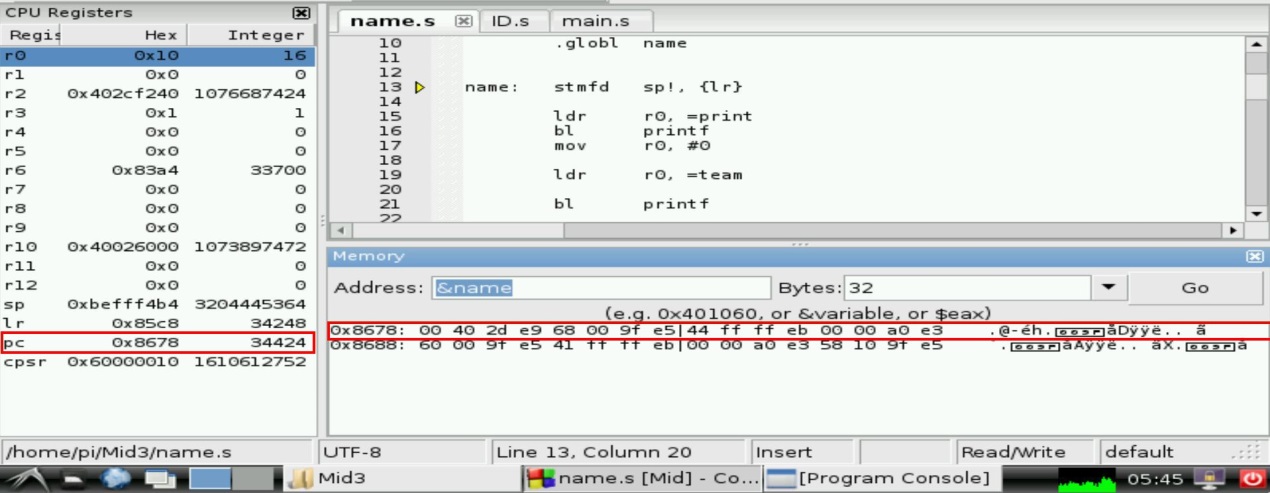
(r7所儲存之學號總和設計在main中輸出時顯示)



‧main：

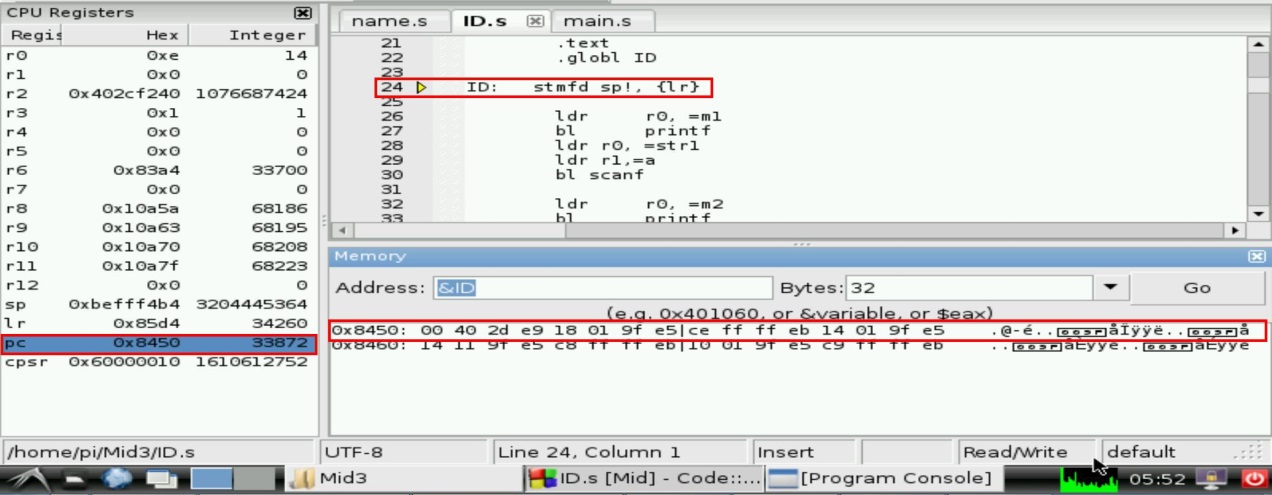
name函數之記憶體位址及內容：

(name的address指向第一道指令)



ID函數之記憶體位址及內容：

(ID的address指向第一道指令)



五、結論

‧完成目標：

>>輸入學號10727124、10727131、10727153，輸入p。

>>分行輸出組別"Team 14"、組員一姓名"Yu-Ting Liu"、組員二姓名"Yen-Lung

Chen"、組員三姓名"Yi-Hung Lu"。

>>分行輸出組員學號10727124、10727131、10727153、學號總和"ID Summation

= 32181408"。

>>分行輸出組別、學號搭配姓名、總和。

‧特殊案例：

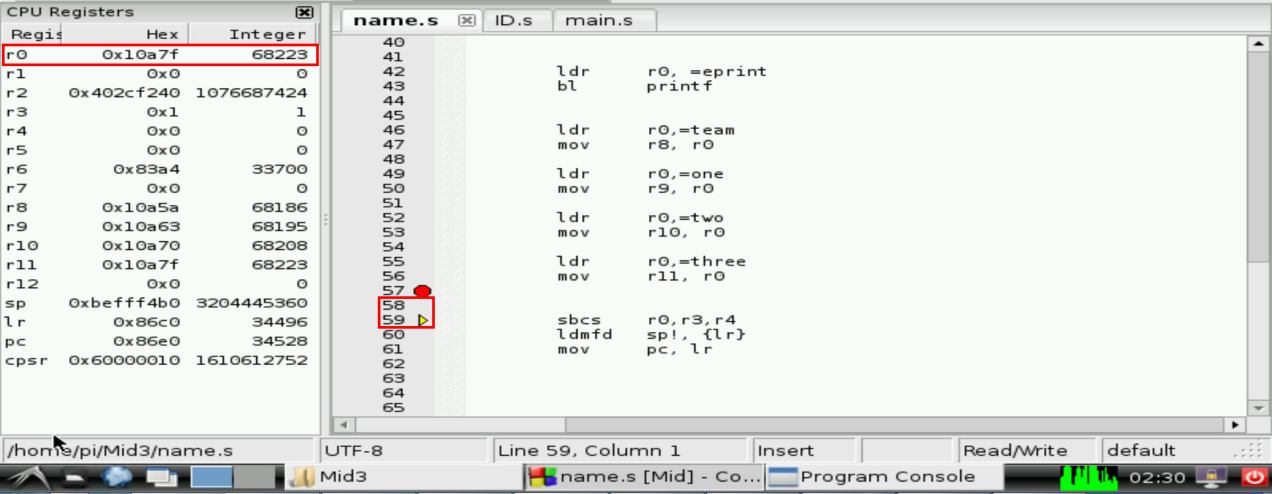
>>輸入學號10727124、10727131、10727153，輸入Q。

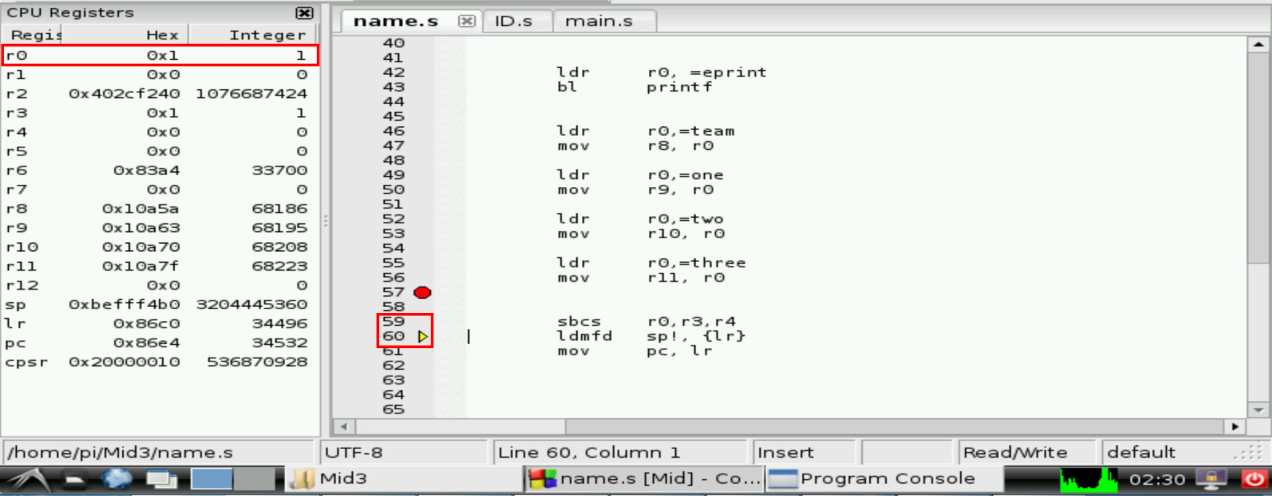
>>輸出"error command"。

>>分行輸出組別、學號搭配姓名，但學號總和為0，代表輸入錯誤Command。

‧指令說明：

name.s中指令：sbc s r0, r 3 , r 4

>>目的在使r0初始化。



定址模式選擇：[Rn]、[Rn, #±<offset\_12>]、[Rn], #±<offset\_12>

>>107：ldr r5, [r5]

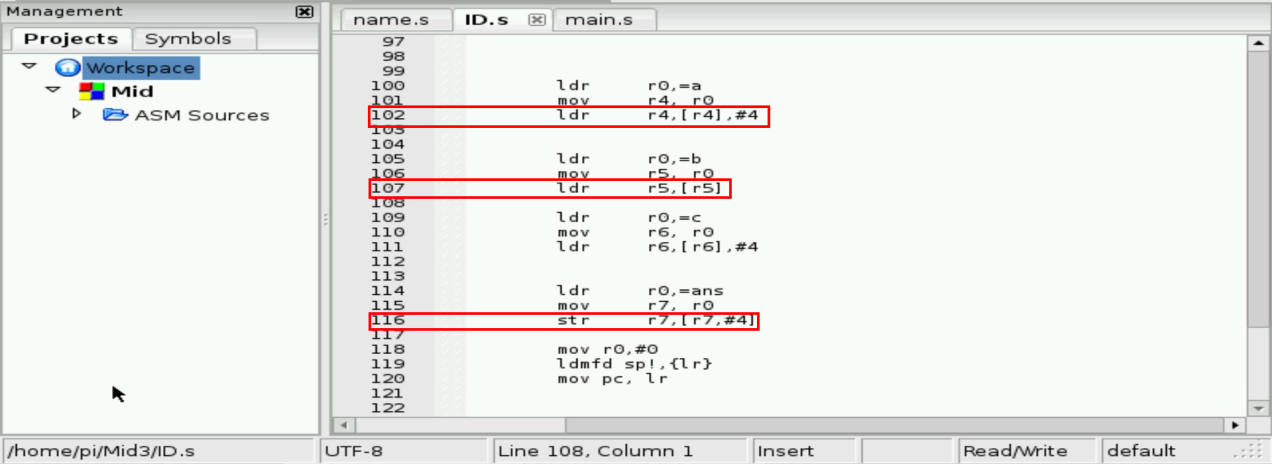
@Load r5 at the address in r5

>>102：ldr r4, [r4], #4

@Load r4 at the r4 address in r4 then increment r4 by 4(r4 = r4 + 4)

>>116：str r7, [r7, #4]

@Store r7 at the address (r7+4)



Operand2 格式選擇：

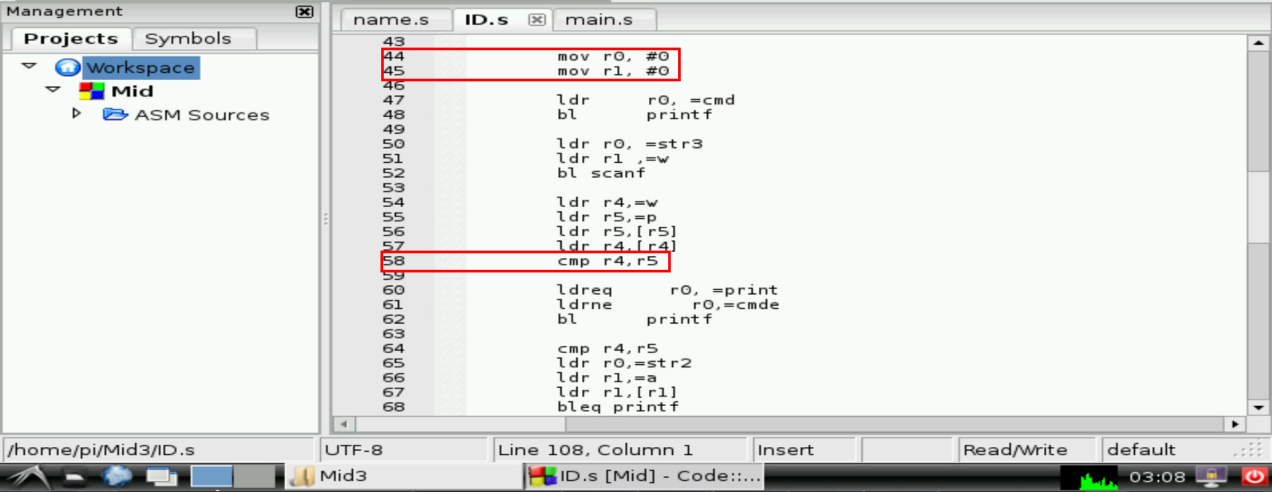
>>44：mov r0, #0

>>45：mov r1, #0

@分別將r0、r1初始化，方便接下來執行printf的工作。

>>58：cmp r4, r5

@r4存放讀進來的字元，r5存放字元p，比較是否相同。



非Branch指令的Contional Execution：

>>60：ldreq r0, =print

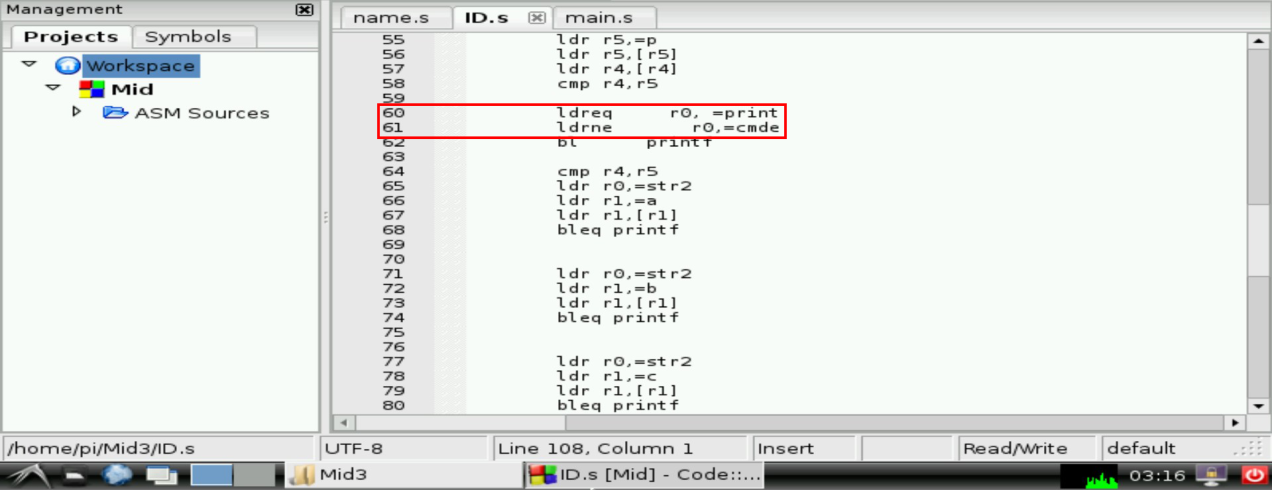
>>61：ldrne r0, =cmde

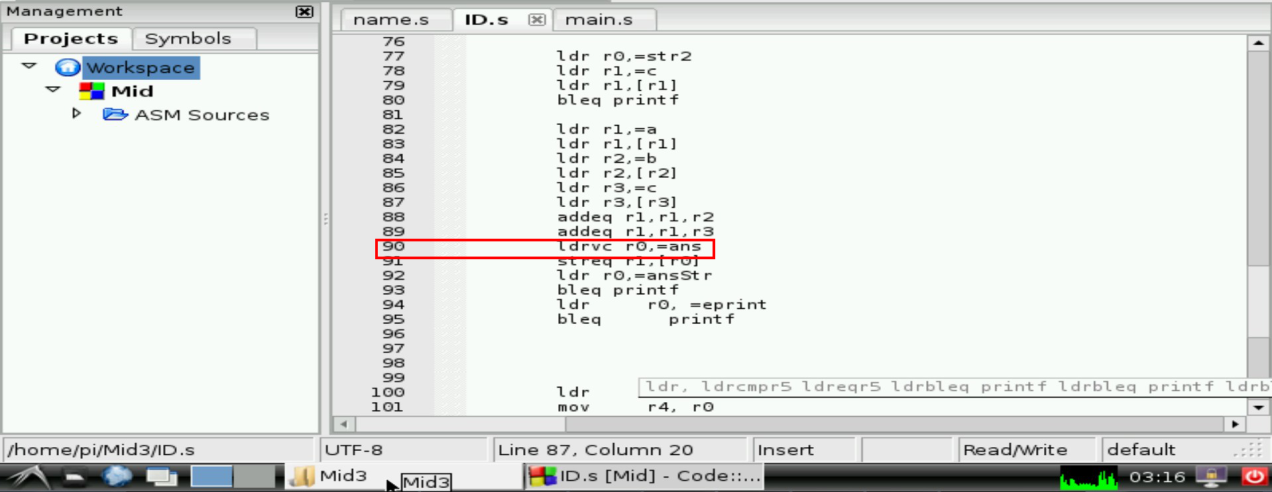
@第59行比較完讀入指令與字元p是否相同後，如果相同則繼續印出，

反之則印出"error command"

>>90：ldrvc r0, =ans

@將學號加總起來後，判斷有沒有發生overflow。





六、未來展望

‧心得：

這一次的project讓我們藉由實際操作學習到了如何使用組合語言撰寫程

式，再一剛開始因為不熟悉code block和組合語言的指令，所以遇到了不少

的困難，像是不知道暫存器中的值或是位址，經常會產生錯誤，或是印出的

東西不是我們要的，但也能藉由debugger來找出有問題的地方，和查看暫

存器的位址和值。這一次所要求的作業也讓我們體驗到，如果是用C或是

其他語言來寫是很簡單的，但換成組合語言就不一樣了，光是scanf就需要

三行程式碼，而其他地方也是如此，讓我們成功了解到程式是如何運作的。

‧分工方式與項目：

本次Project整體架構、構思、程式撰寫都是三人共同完成，包括定址模式

的選擇、Operands2格式的使用，還有Conditional Execution功能的操作，皆

是大家通力合作完成，至於報告的部分10727124負責解釋程式驗證結果、

Debug功能所顯示Memory dump、CPU Register的螢幕截圖，10727131負

責程式說明、設計重點說明，10727153負責心得感想、各組員分工方式與

負責項目以及最終報告的整合。