組合語言Final 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

使用Midterm所撰寫之name與ID兩個函數，列印組別、組員姓名、學號與學號總和，並參考範例程式檔案main.c、drawJuliaSet.c，以ARM組合語言設計drawJuliaSet函數，並依題目需求修改main.c。

二、方法

‧程式說明

1.name.s：印出組別及組員姓名。

2.ID.s：輸入組員學號再輸入p，印出組員學號及學號總和。

3.drawJulia.s：透過公式計算決定顏色，再將該顏色放入宣告的frame二

維陣列中。

4.main.c：整合name、ID和drawJulia函數，呼叫name和ID函數並將回

傳的資料印出來，再按下p鍵後呼叫drawJulia函數進行運算，最後

利用Write函數將畫面資料寫入Frame Buffer中。

‧設計重點說明

>>name：

1.將main傳進來的四個位址(r0-r3)進行備份。

2.配置記憶體給team、one、two、three四個變數。

3.分別存入組別、組員一、組員二、組員三。

4.將變數存入r0，並使用printf印出來，從team開始重複以上步驟直到將

three印出來。

5.分別將team、one、two、three存入一開始備份的四個位址，以便能在

main裡使用資料印出組別及組員姓名。

>>ID：

1.將從main傳進來的四個位址(r0-r3)進行備份。

2.配置記憶體給a、b、c、w、ans五個變數。

3.將格式字串"%d"存入r0，並將a、b、c分別存入r1，再利用scanf將

三個組員的學號分別讀入a、b、c。

4.將格式字串"%s"存入r0，並將w存入r1一樣利用scanf將指令讀入，

再利用cmp比較w及字元p，如果(eq)成立就執行4、5、6、7步驟，

反之如果(ne)成立就將字串"error command"存入r0並列印出來。

5.將格式字串"%d\n"存入r0，並將變數a存入r1列印出來，再將變數b、

c同樣執行以上動作。

6.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。

7.分別將a、b、c、ans存入一開始備份的四個位址，以便能在main裡使

用印出學號及總和。

>>drawJulia：

1.宣告cX、cY、width、height、frame五個變數。

2.將main傳入的資料從r0-r3、sp依序放入以上五變數中。

3.宣告x、y、zx、zy、tmp、color、maxIter、i變數，並將maxIter設為255、

其餘設為0。

4.r0、r1分別讀入x、width並進行比較，如果(ge)成立就移動到end結束

drawJulia，反之進入第一個迴圈forX並將y設為0進行以下步驟。

5.r0、r1分別讀入y、height進行比較，如果(ge)成立就將x值+1並回到第

一個迴圈forX執行步驟4，反之則進入第二個迴圈forY進行以下步驟。

6.將r0、r1、r2存入x、width lsr 1、1500，使r0 = r0 - r1再讓r0 = r0 \* r2

再將r0 = r0 / r1，所得之值存入zx來達成zx = 1500 \* (x-(width>>1)) /

(width>>1)。

7.將r0、r1、r2存入y、height lsr 1、1000，讓r0 = r0 - r1再讓r0 = r0 \* r2

再將 r0 = r0 / r1，所得之值存入zy來達成zy = 1000 \* (y-(height>>1)) /

(height>>1)。

8.r0、r1分別讀入I、maxIter，將r1的值存入r0的位址來達成I = maxIter。

9.r0、r1、r2、r3存入zx、zy、I、4000000，讓r0 = r0 \* r0、r1 = r1 \* r1再

讓r0 = r0 + r1來達成zx \* zx + zy \* zy，將r0和r3作比較，如果(ge)成

立就進行步驟13，反之將r2和0作比較，如果(le)成立就進行步驟13，

反之進入第三個迴圈while進行以下步驟。

10.r0、r1、r2、r3、r4分別讀入zx、zy、1000、cX、tmp，將r0 = r0 \* r0、

r1 = r1 \* r1，再將((r0-r1) / r2) + r3的值存入r4的位址來達成tmp =

(zx\*zx-zy\*zy) / 1000 + cX。

11.r0、r1、r2、r3分別讀入zx、zy、1000、cY，將r0 = 2 \* r0 \* r1，再將

(r0/r2) + r3的值存入r1的位址來達成zy = (2\*zx\*zy) / 1000 + cY。

12.r0、r1、r2分別讀入zx、tmp、i，將tmp的值放入zx的位址來達成zx

= tmp，將r2 = r2 - 1來達成I = I – 1，完成後回到第三個迴圈while進

行步驟9。

13.r1、r3分別讀入I、color，r1 = r1 & 0xff後將r2 = r1 lsl 8，再將r0 = r2 |

r1的值存入r3的位址來達成color = ((i&0xff)<<8) | (i&0xff)。

14.r0讀入color，利用"mvn"讓r1 = ~color，再將r2 = r1 & 0xffff的值放入

r0的位址來達成color = (~color) & 0xffff。

15.r0、r1、r2、r3分別讀入x、y、width、frame，讓r0 = r3 + (2\*r1\*r2+2\*r0)

來達成frame + (2\*y\*width+2x)的計算，藉此找到frame[y][x]之位址，r1

存入color後將r1的值放入r0的位址來達成frame[y][x] = color。

16.r0讀入y，讓r0 = r0 + 1來達成y = y + 1後回到第二個迴圈forY進行

步驟5。

>>main：

1.Define frame的寬和高為640及480。

2.宣告四個變數分別為name1、name2、name3、team。

3.呼叫name函數並將name1、name2、name3、team四個變數的位址傳入

name函數中。

4.宣告四變數id1、id2、id3、ans四變數。

5.呼叫ID函數並將id1、id2、id3、ans四變數的位址傳入ID函數中。

6.將name函數和ID函數回傳的組別、組員姓名及學號列印出來。

7.宣告tempteam、tempname1、tempname2、tempname3、tempid1、tempid2、

tempid3、tempans八個變數，並將組別、組員姓名、學號存入此八個變

數中。

8.宣告二維陣列frame[HEIGHT][WIDTH]、cX、cY、max\_cX、min\_cY、

cY\_step、變數。

9.將cY設為400並依據公式cY = cY + cY\_step更改cY的值，在cY <

min\_cY為止前執行迴圈持續呼叫drawJulia函數，並將cX、cY、frame

的寬和高及frame的位址傳入drawJulia函數。

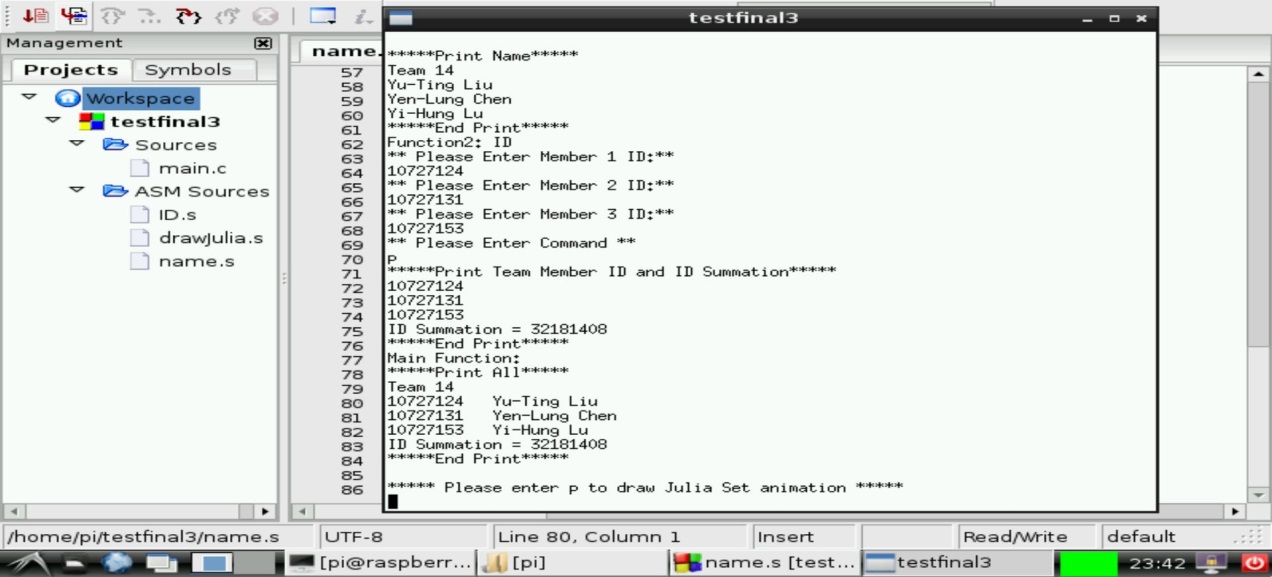
10.動畫執行完畢後利用步驟7所宣告的八個變數，將組別、組員姓

名及學號列印出來。

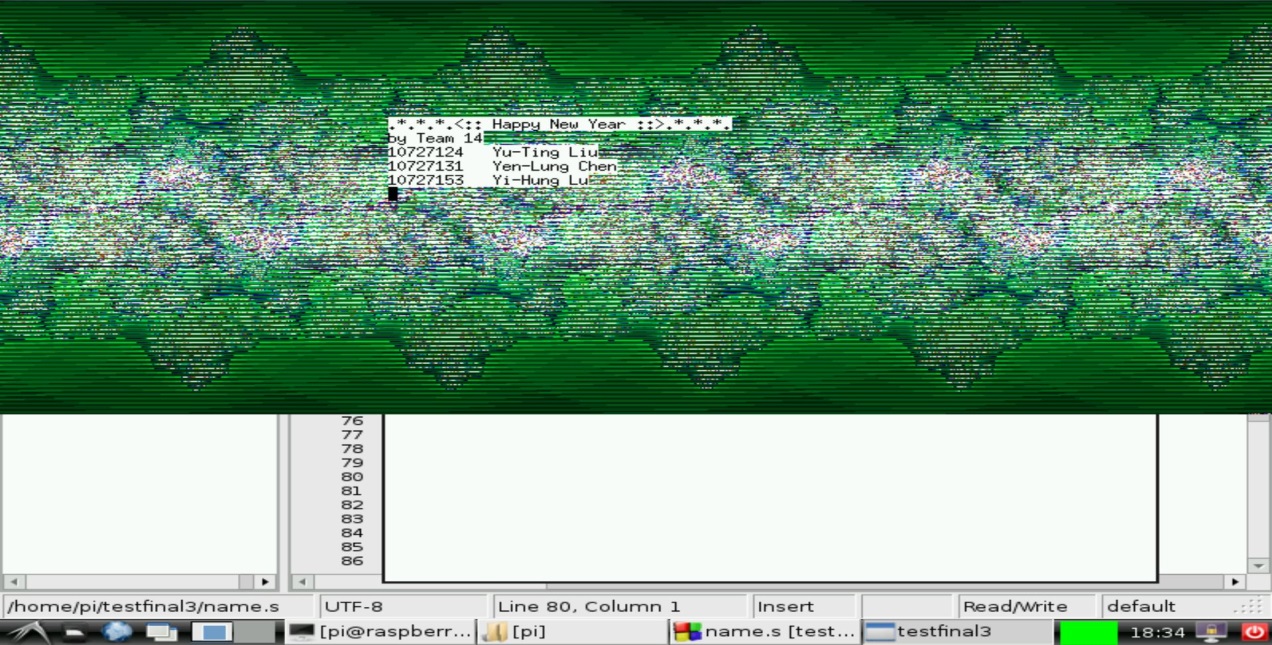
三、結果

‧程式執行結果

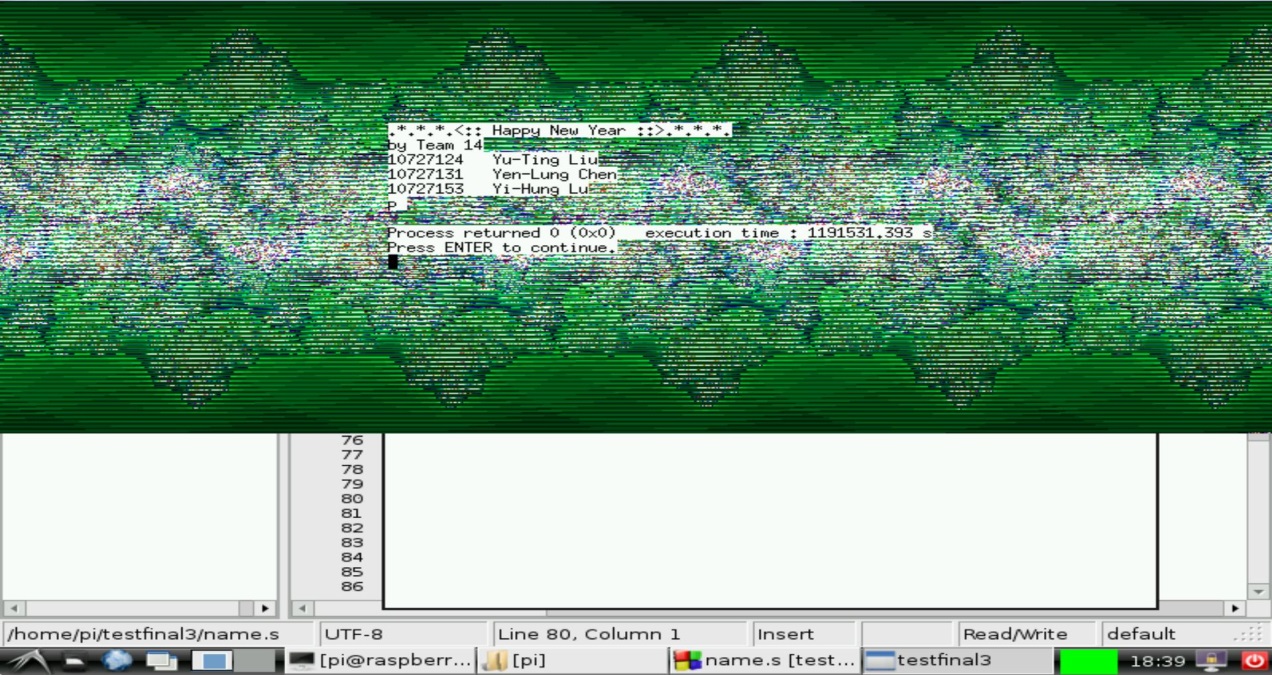
>>輸入學號、輸入p。



>>再次輸入p(執行JuliaSet動畫)。



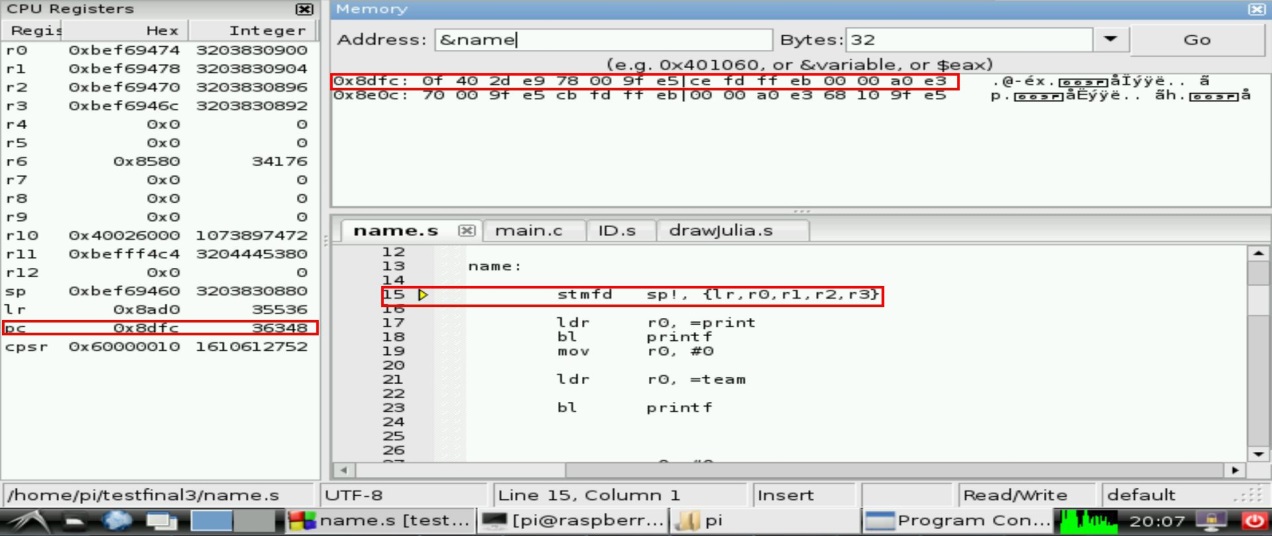
>>讀入p程式執行結束。



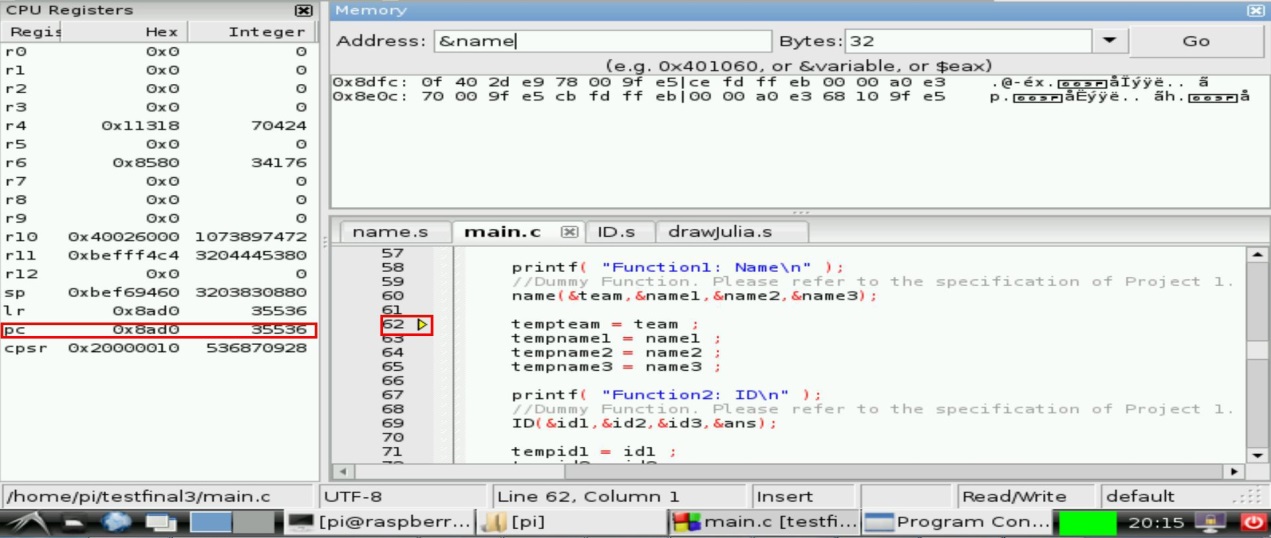
四、討論

‧name函數之記憶體位址及內容

>>name的所在位址(指向程式中第一道指令)。

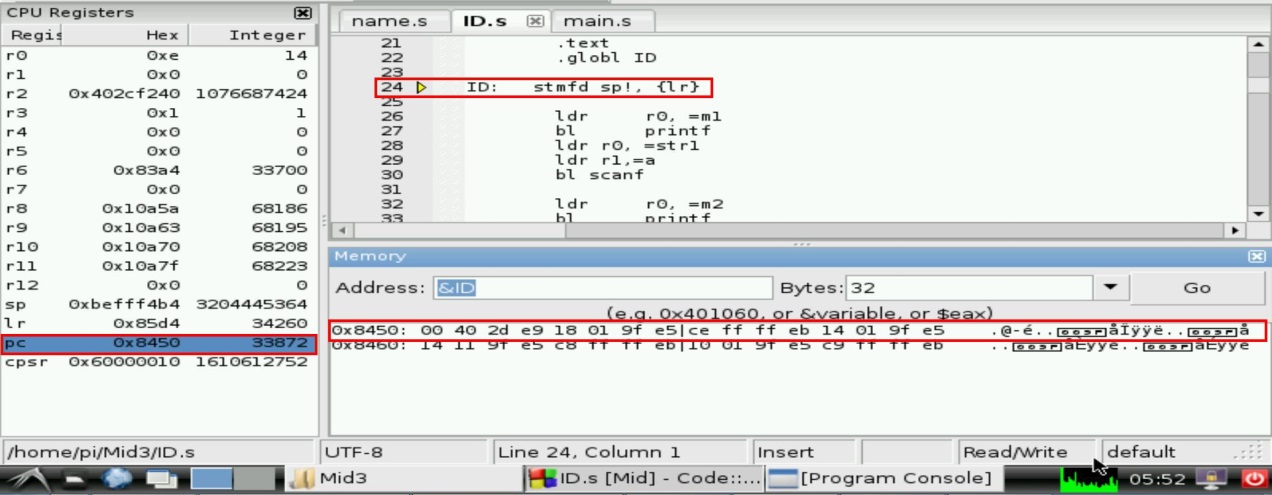


>>name的返回位址(return address)。

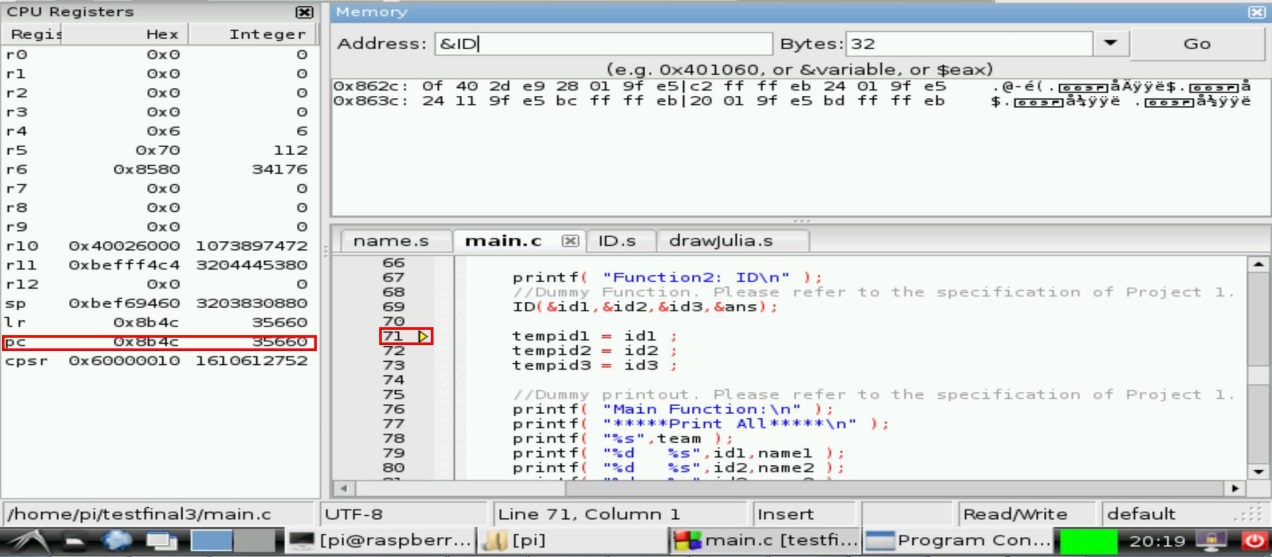


‧ID函數之記憶體位址及內容

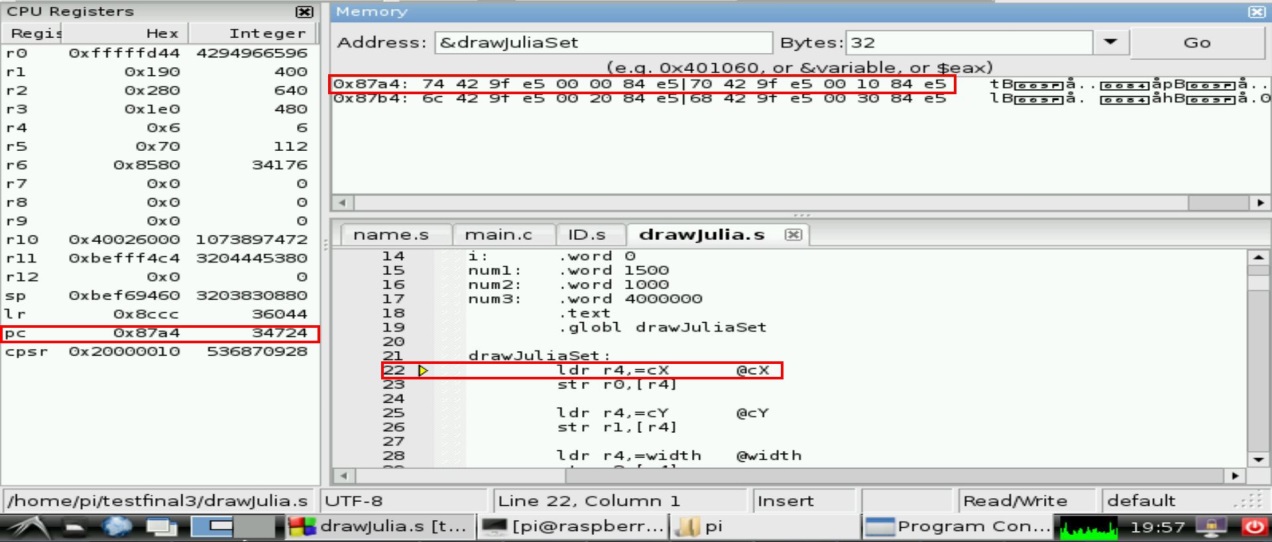
>>ID的所在位址(指向程式中第一道指令)。



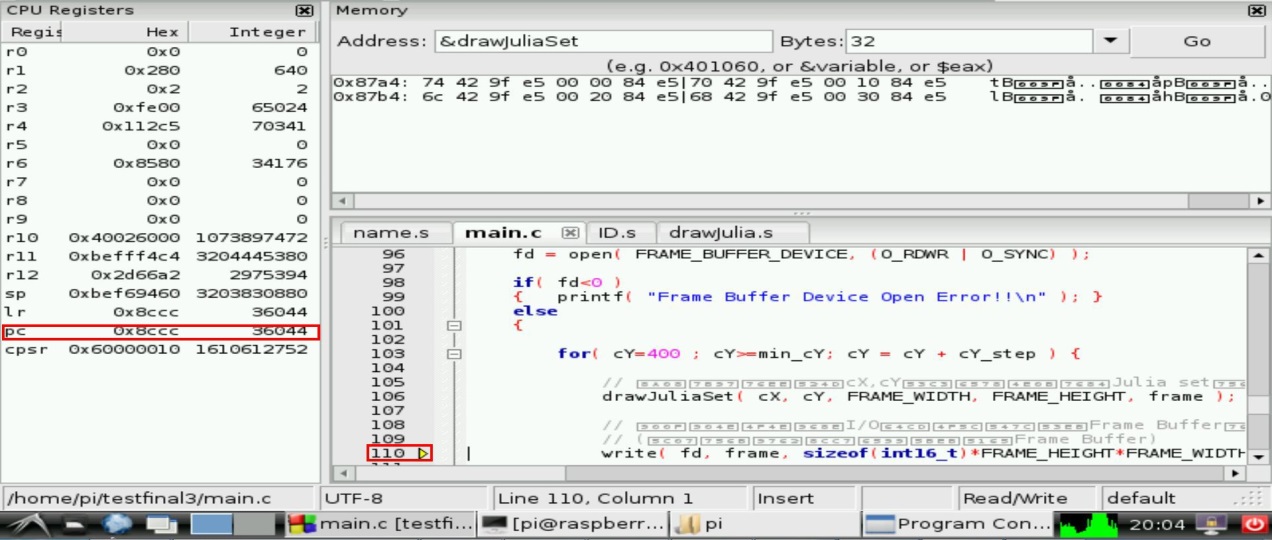
>>ID的返回位址(return address)。



‧drawJuliaSet函數之記憶體位址及內容

>>drawJuliaSet的所在位址(指向程式中第一道指令)。

>>drawJuliaSet的返回位址(return address)。

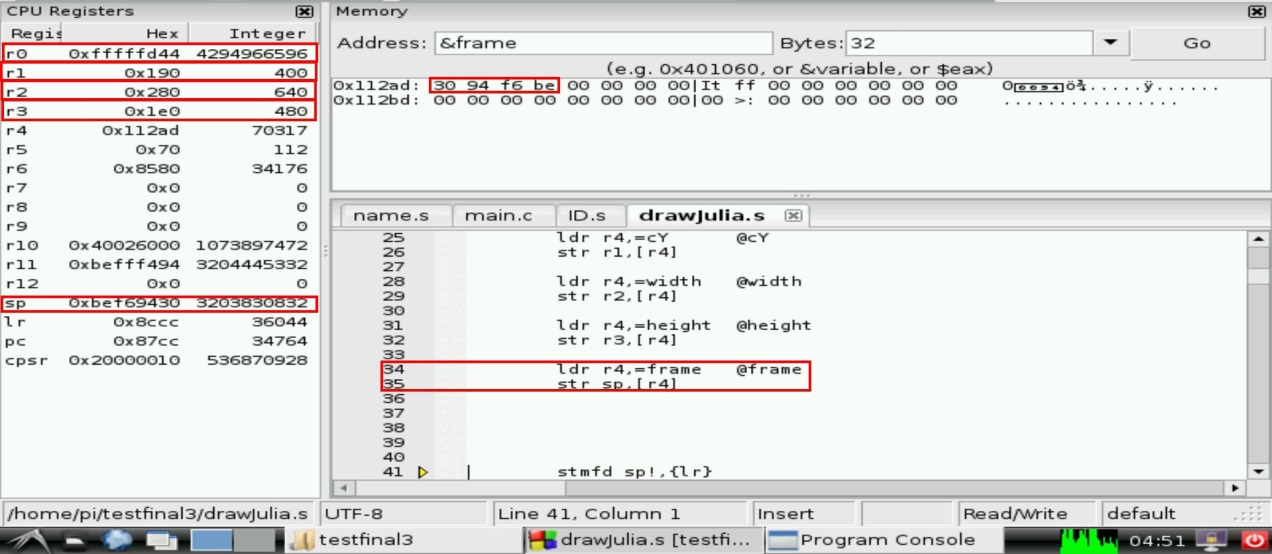


‧frame陣列之記憶體位址及內容

>>frame陣列的起始位址：

剛進入drawJulia函數時，r0 - r3分別存放cX、cY、WIDTH、HEIGHT，

而sp存放frame陣列的起始位址0xbef69430。

(配置一記憶體frame儲存frame陣列的起始位址。)

>>計算frame陣列的終止位址：

frame陣列的大小為frame[480][640]，因此陣列中最後一個index是

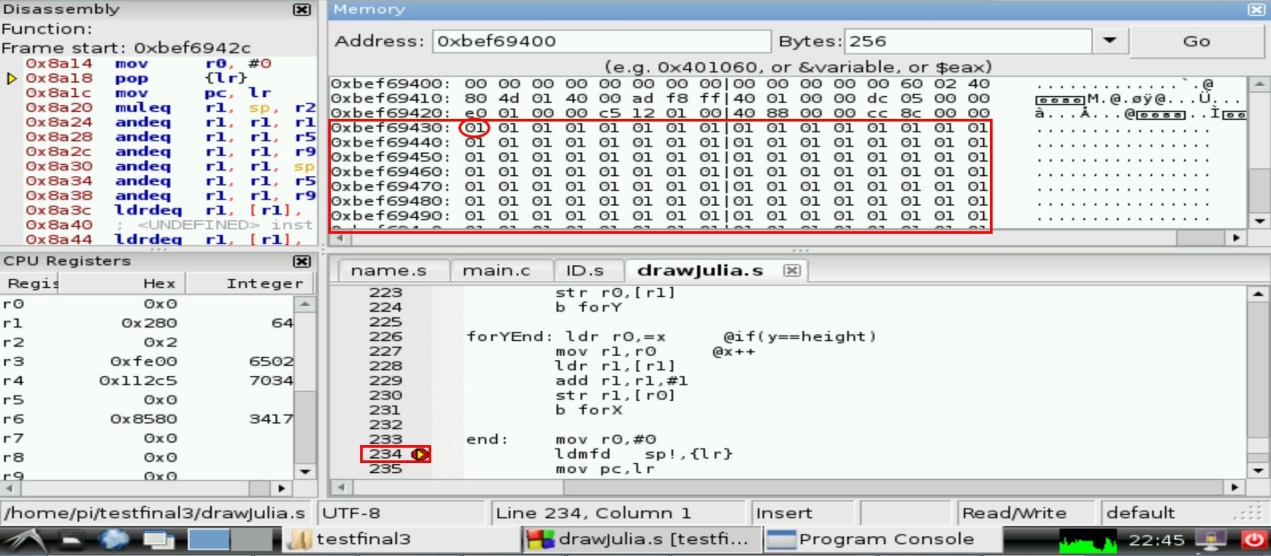
frame[479][639]，利用(479\*640+639) \* 2算出間距並將值轉換成16進制，

加上frame陣列的起始位址最後等於0xbefff42e，由於每單位存放2byte，

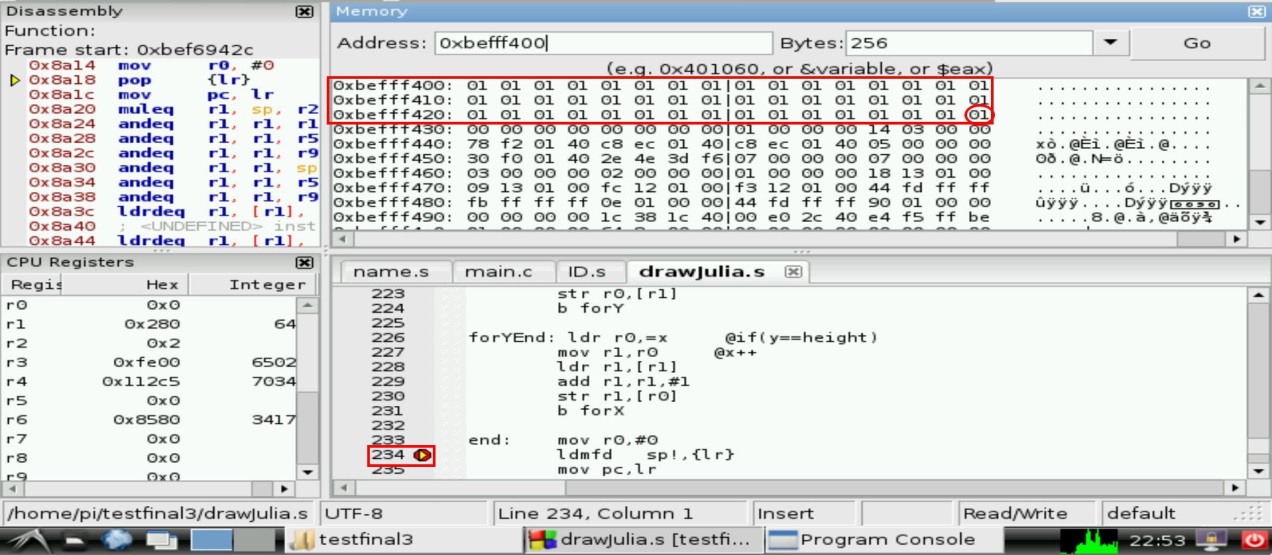
因此0xbefff42e +1 = 0xbefff42f 即是frame陣列的終止位址。

>>frame陣列的記憶體區塊部份內容：

(0xbef69430為起始位址)



(0xbefff42f為終止位址)



五、結論

‧完成目標

>>呼叫name與ID函數列印組別、組員名字、學號與學號總和。

>>讀入p執行Julia Set，並透過函數回傳的資料，分行輸出組別、學號

搭配姓名及學號總和於最後一個畫面。

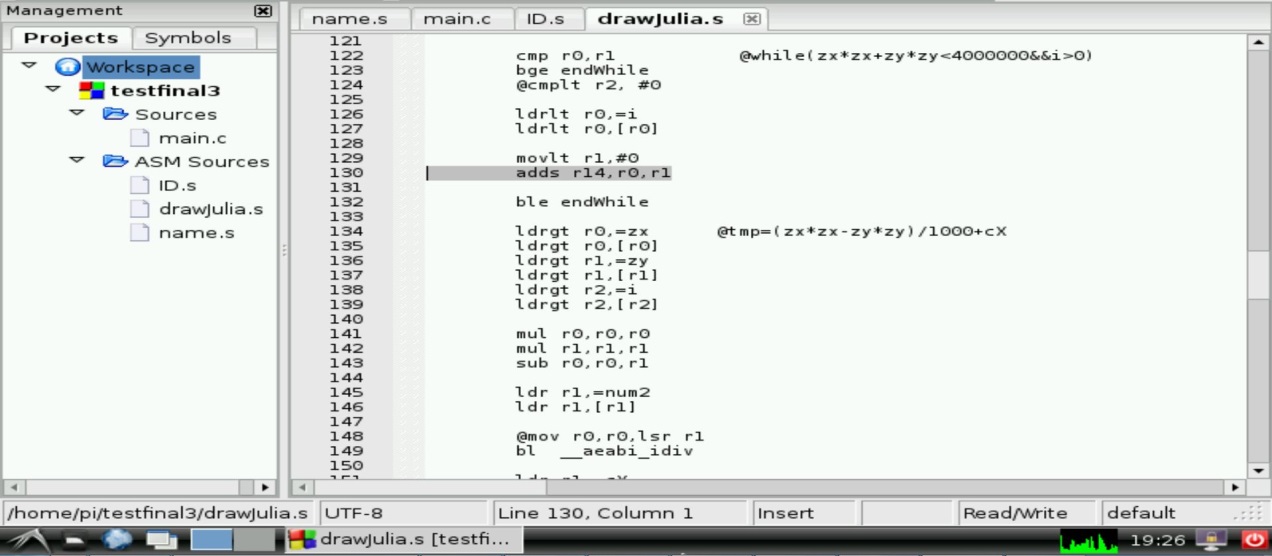
>>讀入p程式執行結束。

‧指令說明

指定指令：adds r14, r0, r1

>>130：判斷i是否大於0。

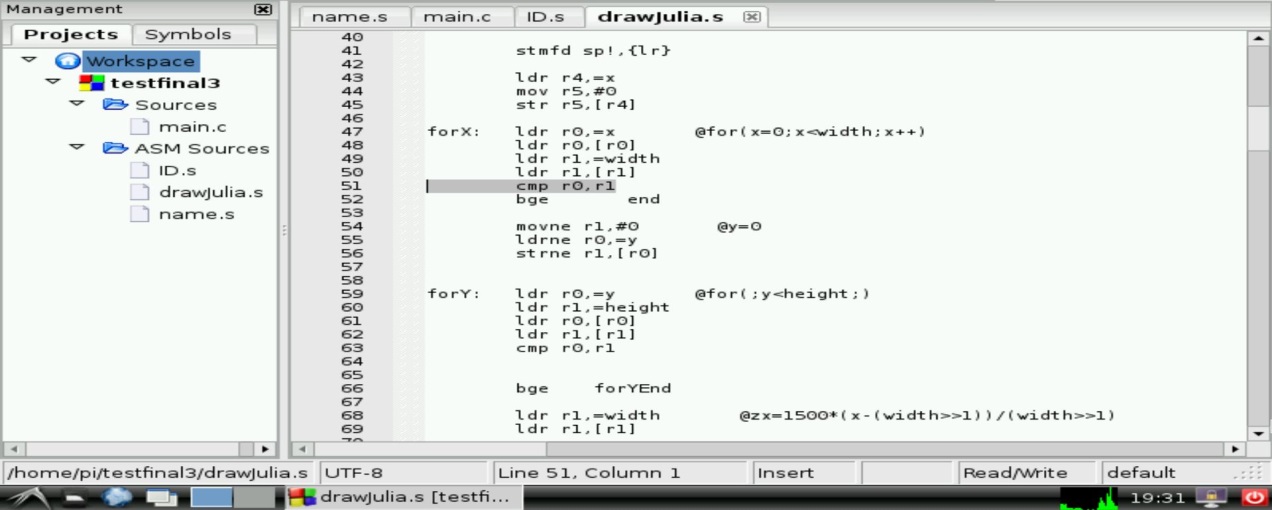
@先將r0設定為i的值、r1設定為0，r14 = r0 + r1 and set CPSR。



Operand2 格式選擇：

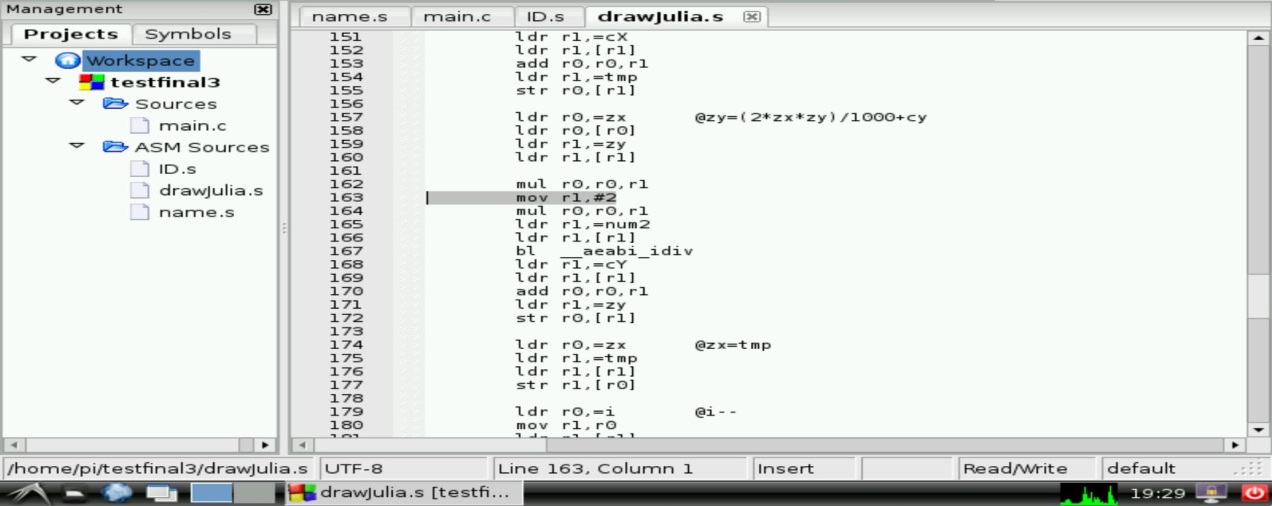
>>51：cmp r0, r1

@ r0存放x的值，r1存放width的值，比較r0、r1。



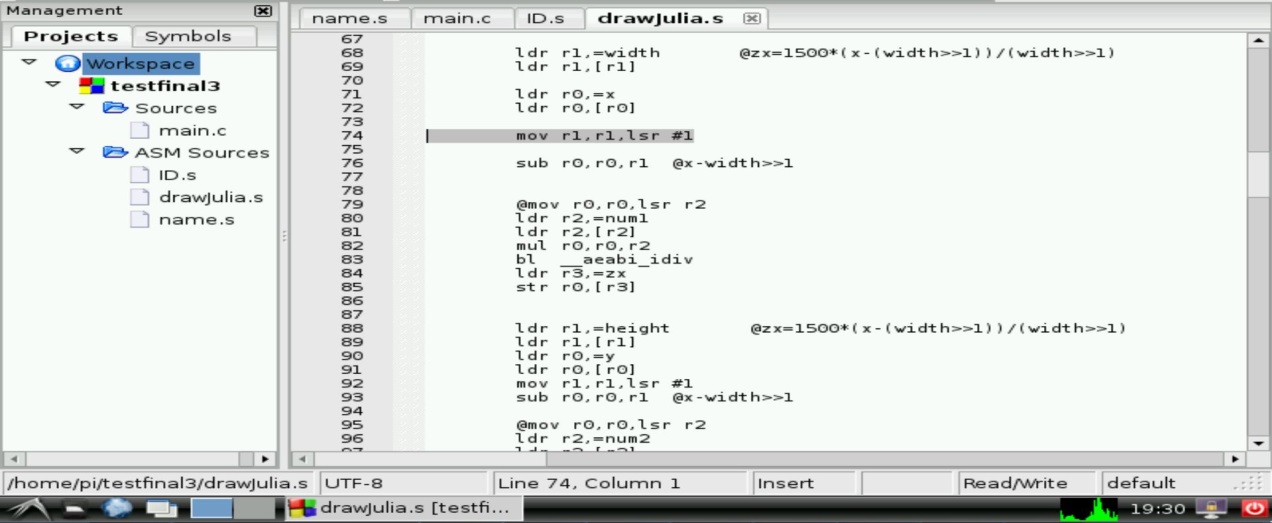
>>74：mov r1, #2

@將2 aasign 到r1，滿足下一行指令乘以2的需求。



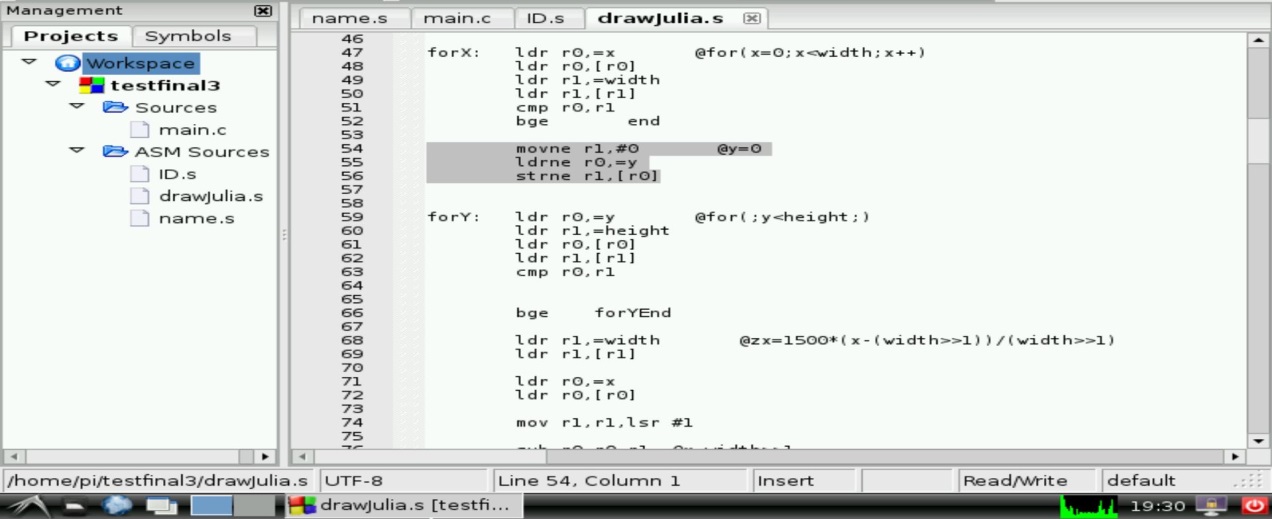
>>163：mov r1, r1, lsr #1

@將r1向右移動1 bit，並把值存放到r1。



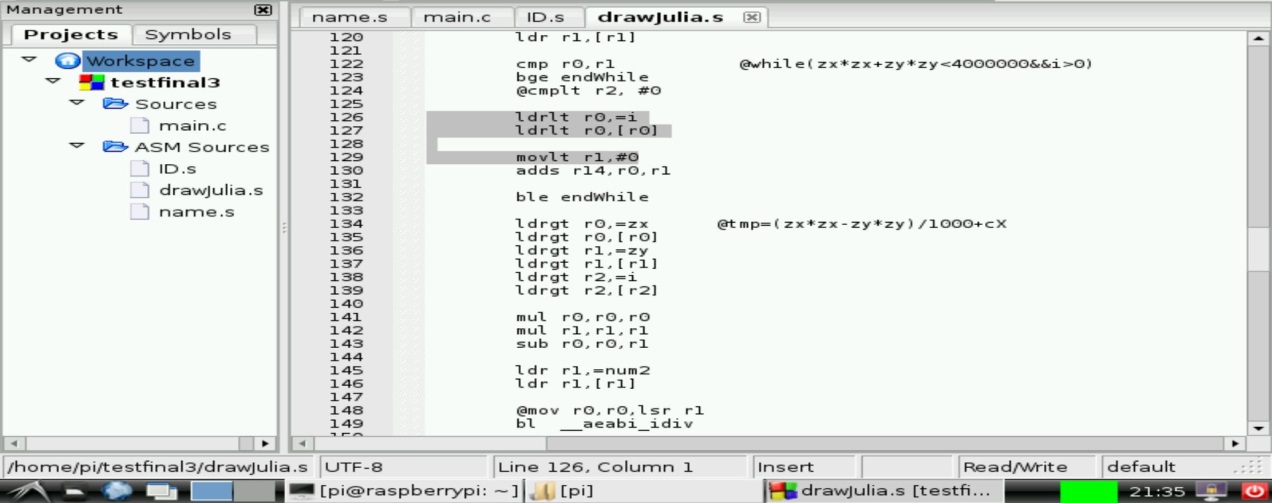
非Branch指令的Contional Execution：

>>54：movne r1, #0

@在51行比較完後，處理r0不等於r1的情形。

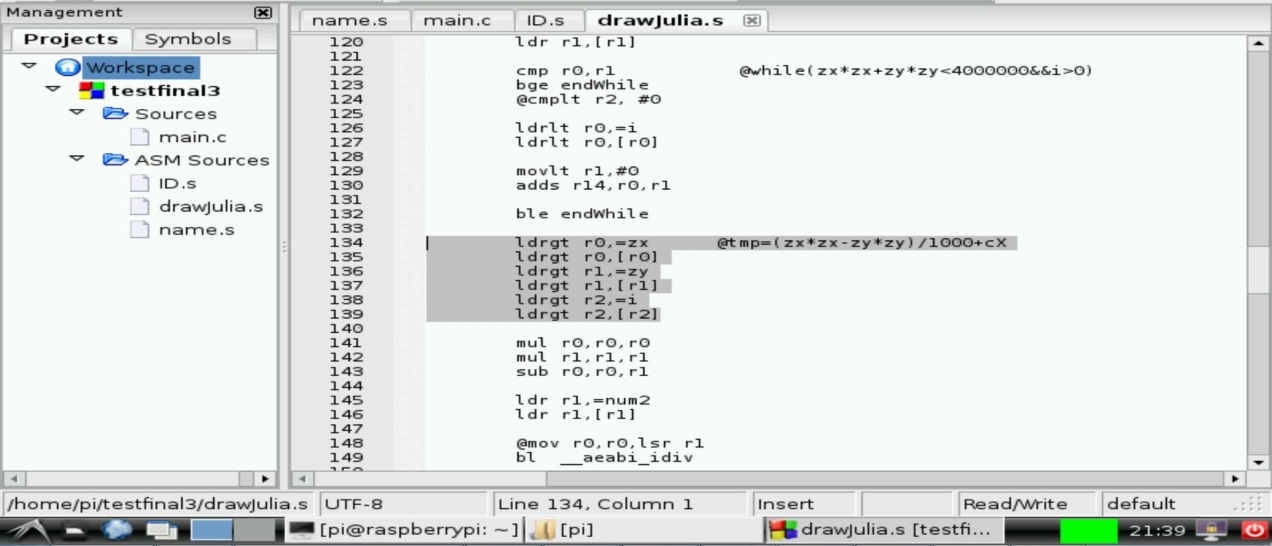
>>129：movlt r1, #0

@在122行比較完後，處理r0小於r1的情形。



>>134：ldrgt r0, =zx

@在130行執行完後，處理r0大於r1的情形。



六、未來展望

‧心得

這次的Final Project比起上次的Midterm Project難度明顯提升不少，首先要先將老師提供的Julia Set程式碼轉換為Assembly Language，由於Julia Set的程式碼中有多個迴圈，因此在Branch指令的使用上必須謹慎處理，另外其中也包含許多需要判斷的地方，故條件執行的指令也需要多著墨，還有一項重點是除法的應用，因為ARM處理器版本的關係不能使用"DIV"指令，後來上網搜尋發現可以使用"\_\_aeabi\_idiv"指令才得以解決，雖然這次的project真的費了不少功夫，也花了不少時間，但相對的也學習到很多相關的知識，並能有效的將老師上課所傳授的概念，應證在實際的操作上。

‧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Operands

格式的使用，還有Conditional Execution功能的操作，皆是大家通力合作完成，特別是在Debug的時候，大家絞盡腦汁的試了各種方法，才解決了Segmentation Fault造成的問題，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。