

Arm Assembly Final project

老師：朱守禮

組別：7

組員：

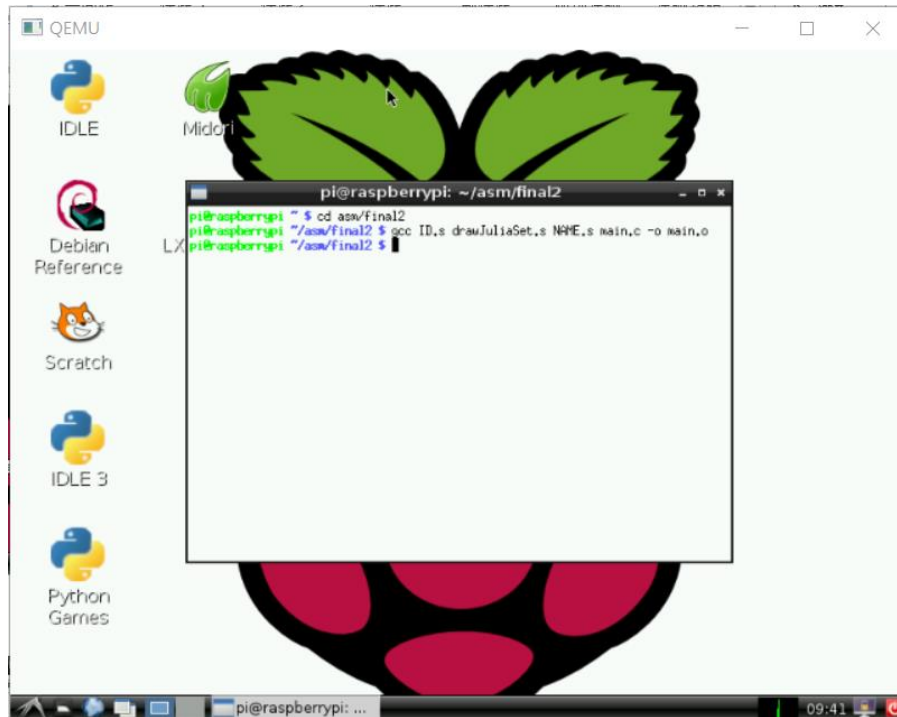
10520127 電資三 戴承浩

10520104 電資三 莊東翰

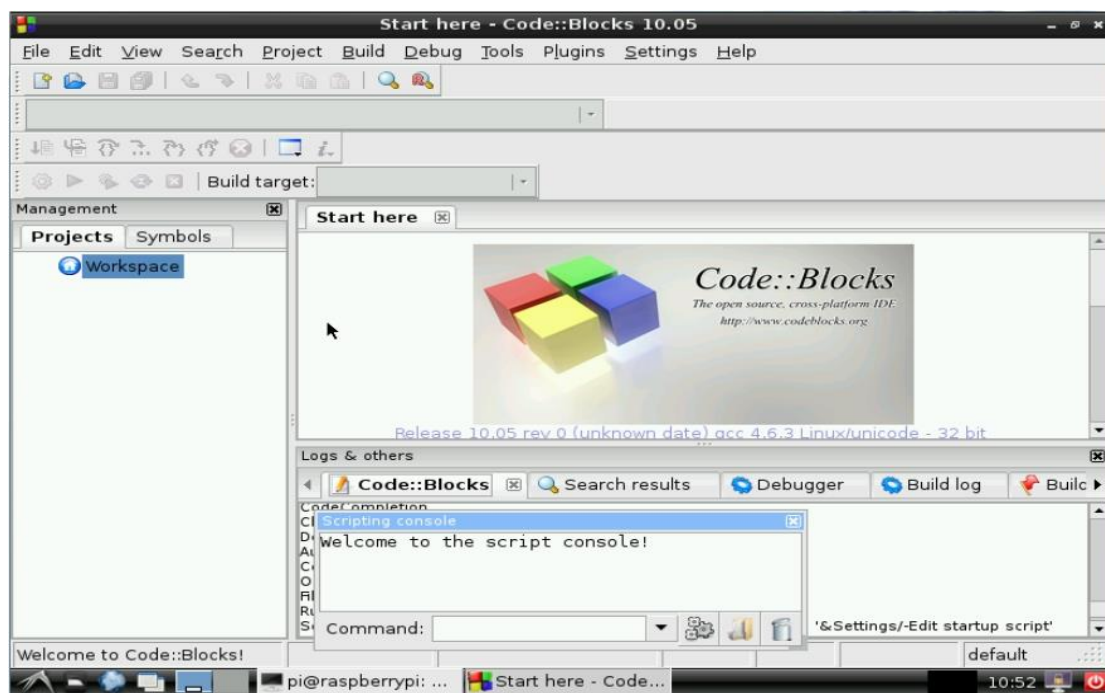
1. 背景

ARM Assembly Language

在 raspberrypi 模擬器下運行



Programing >> Code::Blocks



2. 方法

分四個檔 ID.s NAME.s drawJuliaSet.s main.c

先做介面需要的記憶體空間規劃，在 main.c 中依序呼叫 NAME.s、ID.s 和

drawJuliaSet.s。並且利用 drawJuliaSet.s 指定座標像素點的颜色，利用

framebuffer 做出圖形的處理，顯示碎形。

```
29 int main()
30 {
31     //RGB16
32     int16_t frame[FRAME_HEIGHT][FRAME_WIDTH];
33     int max_cX = -700;
34     int min_cY = 270;
35
36     int cY_step = -5;
37     int cX = -700; // x = -700~-700
38     int cY;        // y = 400~270
39     int fd;
40     int * id[4];
41     char name[4][50];
42
43     id[0] = (int*) malloc( sizeof( int ) );
44     id[1] = (int*) malloc( sizeof( int ) );
45     id[2] = (int*) malloc( sizeof( int ) );
46     id[3] = (int*) malloc( sizeof( int ) );
47
48
49     printf( "Function1: Name\n" );
50
51     //Dummy Function. Please refer to the specification of Project 1.
52     NAME( name[0], name[1], name[2], name[3] );
53
54     printf( "Function2: ID\n" );
55
56     //Dummy Function. Please refer to the specification of Project 1.
57     ID( id[0], id[1], id[2], id[3] );
58
59     //Dummy printout. Please refer to the specification of Project 1.
60     printf( "Main Function:\n" );
61     printf( "*****Print All*****\n" );
62
63     printf( "%s", name[0] );
64     printf( "%d %s", *id[0], name[1] );
65     printf( "%d %s", *id[1], name[2] );
66     printf( "%d %s", *id[2], name[3] );
67     printf( "ID Summation = %d\n", *id[3] );
68     printf( "*****End Print*****\n" );
69
70     printf( "\n***** Please enter p to draw Julia Set animation *****\n" );
71     // 等待使用者輸入正確指令
72     while( getchar() != 'p' ) {}
73
74     // 清除畫面
75     system( "clear" );
76
77     // 打開 Frame Buffer 硬體裝置的Device Node，準備之後的驅動程式呼叫
78     fd = open( FRAME_BUFFER_DEVICE, (O_RDWR | O_SYNC) );
79
80     if( fd<0 )
81     { printf( "Frame Buffer Device Open Error!!\n" ); }
82     else
83     {
84         for( cY=400 ; cY>=min_cY; cY = cY + cY_step ) {
85
86             // 計算目前cX,cY參數下的Julia set畫面
87             drawJuliaSet( cX, cY, FRAME_WIDTH, FRAME_HEIGHT, frame );
88
89             // 透過低階I/O操作呼叫Frame Buffer的驅動程式
90             // (將畫面資料寫入Frame Buffer)
91             write( fd, frame, sizeof(int16_t)*FRAME_HEIGHT*FRAME_WIDTH );
92         }
93     }
94 }
```

各別的函數內部運作

程式設計：

NAME 函數功能：

說明：先在 NAME 函數裡定義好想印出的字串，並利用 strcpy 傳回 main，以方便未來能夠直接在 main 再次印出

```
.data
msg1:  .asciz  "Team07\n"
msg2:  .asciz  "ZHUANG DONG HAN\n"
msg3:  .asciz  "DAI CHENG HAO\n"
msg4:  .asciz  "ZHUANG DONG HAN\n"

.text

.globl NAME

store:
    mov r0, r8
    ldr r1, =msg1
    bl strcpy
    mov r0, r9
    ldr r1, =msg2
    bl strcpy
    mov r0, r10
    ldr r1, =msg3
    bl strcpy
    mov r0, r11
    ldr r1, =msg4
    bl strcpy

    ldmfd sp!, { r8-r11,pc }
```

ID 函數功能：

說明：先在 main 中定義出四個指標陣列，用來記錄儲存學號的變數的位置和相加結果。在 ID.s 中輸入三筆學號資料，在最後依序沿著 r0, r1, r2, r3 紀錄的位置擺入學號跟相加結果。

```
41  .globl ID
42
43  ID:
44      stmfd  sp!, {r8-r11,lr}    @ push lr onto stack
45      mov r8, r0
46      mov r9, r1
47      mov r10, r2
48      mov r11, r3
```

```

149      ldr    r3, =b
150      ldr    r0, =n
151      ldr    r1, =m
152      ldr    r2, =a
153      ldr    r3, [r3]
154      ldr    r0, [r0]
155      ldr    r1, [r1]
156      ldr    r2, [r2]
157      str    r0, [r8]
158      str    r1, [r9]
159      str    r2, [r10]
160      str    r3, [r11]

```

DrawJuliaSet 函數功能：

說明：設定大小 640 X 480 的 frame 陣列中計算每一格對應的值，這些值影響顏色的輸出。

Main 函數功能：

說明：呼叫 ID、NAME、DrawJuliaSet 等功能，最後輸入 'p' 印出碎形。

設計重點：負責整合的 main 可以使用高階語言的 c 檔，利用 c 檔來輸入 framebuffer 指令，開啟 framebuffer，運算的動作可以由 s 檔來撰寫。

3. 結果

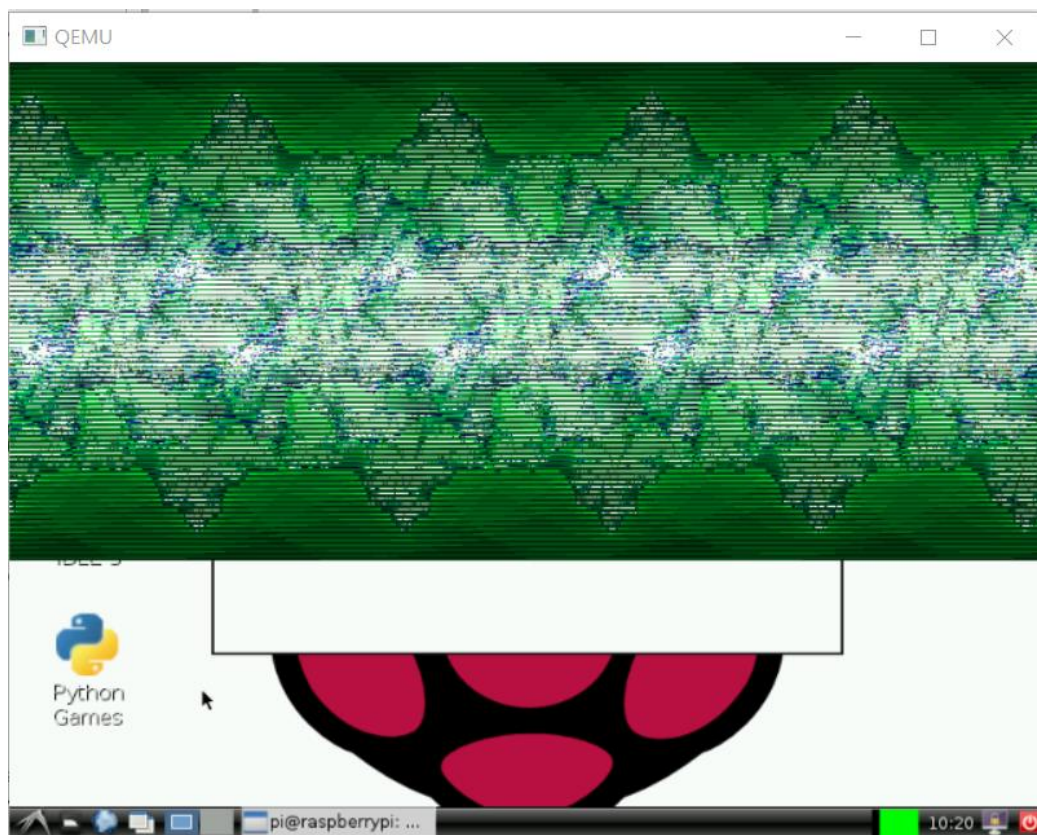
```

pi@raspberrypi: ~/asm/final2
pi@raspberrypi ~ $ cd asm/final2
pi@raspberrypi ~/asm/final2 $ gcc ID.s drawJuliaSet.s NAME.s main.c -o main.o
pi@raspberrypi ~/asm/final2 $ ./main.o
Function1: Name
Team07
ZHUANG DONG HAN
DAI CHENG HAO
ZHUANG DONG HAN
Function2: ID
Enter the first sid:
10
Enter the second sid:
20
Enter the third sid:
30
Please enter the command:
p

```

```
pi@raspberrypi: ~/asm/final2
Enter the first sid:
10
Enter the second sid:
20
Enter the third sid:
30
Please enter the command:
p
****Print Team Member ID and ID Summation****
10
20
30
The sum is: 60
Main Function:
****Print All****
Team07
10  ZHUANG DONG HAN
20  DAI CHENG HAO
30  ZHUANG DONG HAN
ID Summation = 60
****End Print****

**** Please enter p to draw Julia Set animation ****
p
```



4. 討論

這次為了寫出條件式和迴圈使用了更多 label 做判斷的動作。因為這次使用了許多 label，所以一開始讀入的 5 個值有 4 個被放入 stack 中做保存(第五個原就在 stack 中)，以方便後續要使用的時候直接控制 sp 就能取到需要的值。

5. 結論

這次碰到有幾個問題：這次使用的 ARM 版本無法計算除法、如何在迴圈裡的 `cmp` 中判斷的兩個暫存器內容是正確的、如何把東西正確地放入在 `main` 已經宣告的陣列中。解決這些問題的方法就是要會使用 `stack` 跟學會從 `main` 傳值。

《 心得 》

10520127 電資三 戴承浩

這次最大的問題是要釐清 `sp` 的運用，因為利用堆疊需要理解裡面的 `byte` 的配置，以作為 `main.c` 裡 `function` 的一些參數回傳，比較麻煩的是有三個 `function` 需要處理，因為都利用 `sp` 堆疊的做代數的傳遞，所以需要釐清每個參數放在 `sp` 的配置分別占用幾個 `byte`

10520104 電資三 莊東翰

期末作業讓我認識到了 `c` 檔與 `s` 檔整合的寫法，認識到了如何從 `c` 檔傳值到 `s` 檔中、迴圈的寫法以及將值放入 `stack` 中儲存的作法。經過這次作業讓我更熟悉在硬體中資料的處理運作方式，在 `C` 中能簡單完成的方法在 `ARM` 組合語言中並沒有那麼容易完成。

組員分工：

10520104 電資三 莊東翰

程式撰寫(主要 傳值、stack、迴圈條件判斷)與 Debug

資料查尋

討論

書面資料

10520104 電資三 戴承浩

程式撰寫(主要 除法、陣列操作) 與 Debug

資料查尋

討論