期末Project

114學年度第1學期

老師：朱守禮　老師

組別:18

  學生：

11327144 莊有隆

                    11327137 林宇宸

一、背景

目的：

熟悉 ARM 組合語言語法、大數據量記憶體存取（Frame Buffer）與數學演算法實作。

核心任務：

在樹莓派模擬器的 Console 模式下，透過組合語言計算並繪製動態的 Julia Set 分形圖形。

二、方法

工具：

            Notepad：撰寫程式。

           WinSCP：下載專案檔案。

           QEMU模擬器：使用 Code::Blocks 執行組合語言檔及檢索暫存器等資料。

     1. 程式說明

      name.s ：印出組別及三名組員的英文名字。

      id.s   ：讀取三名成員的學號，並根據讀入的控制指令判斷是否輸出出， 隨後依序輸出學號，並將學號加總後輸出。

     main.s ：負責控制動畫循環、開啟/dev/fb0設備節點、並將

                 frame陣列寫入顯示記憶體。

drawJuliasSet.s：主要用來計算julia set動畫的每個pixel顏色，並將結果存入frame二維陣列，最後由main.c寫入Frame Buffer顯示。

     2. 設計重點說明

            應避免使用 r0 至 r3 暫存器來存取欲存放的資料，因為 printf

及scanf 等函式常使用此四個暫存器運算，故存放內容極易發生變 化。

* name.s:

              需分配三段記憶體空間存放名字字串內容，並靈活運用 bl

          printf指令。首先使用 ldr 將 r0 暫存器載入欲輸出內容，並執

          行 printf。bl 指令中，b 表示進入 branch (分支)，l 表示將 PC

(執行下一道指令的位址) 存進 LR (函式返回地址)，以便返回主程式繼續執行，避免位址錯亂。

* id.s :

             除了輸出基本字串格式外，則需要使用 bl scanf 指令進行資料

讀入。但在讀入前需做好前置工作：準備 r0 暫存器作為讀入資料格式，準備 r1 暫存器作為存放欲讀入資料的記憶體段位址。

* enter\_student\_number 讀入段：

             共 3 筆資料，r11 暫存器用於計算次數。若次數大於 3，則跳出

          label load\_number。後續進行 scanf 前置動作，並使用

          strle r4, [r7],  #4 post-index 將整數資料存進去，並位移 4 位

          元組 (byte) 偏移量，以存取另外 2 筆資料，持續迴圈。

* 輸出 student\_number 段：

            再將 r7 暫存器內的三筆資料取出，進行輸出及總和運算。

* Id.s特殊要求：

             mov r12, sp：備份堆疊指標 (SP) 至 r12。

              rsbs sp, lr, pc：透過逆減法擾亂 SP 記憶體。

              mov sp, r12：將備份資料放回 SP。

          運用 mov 將 SP 暫存器內資料直接備份至 r12 暫存器，便無需

擔心後續指令是否會擾亂 SP，只需最後將 r12 備份資料放回 SP 即可。

* main.c :

程式主要透過呼叫name()、id()與 drawJuliaSet()三個函式來整合組員資料與julia Set動畫功能。name()會印出組員姓名及組別名稱，id() 則讀入三個學號並將其寫入全域陣列id\_store，同時計算總和存入id\_sum\_val。由於資料已存入全域變數，main.c 僅需載入所需資料即可進行顯示，而無須再次計算。

使用者輸入學號後，程式即時計算總和，並透過 printf 將完整組員資訊與總和分行印出。為避免影響 printf 與 scanf 所使用的r0–r3暫存器，資料皆透過全域變數或其他暫存器存取，確保計算與顯示正確。

動畫部分，程式直接操作/dev/fb0的 Frame Buffer，將 drawJuliaSet() 計算出的畫面逐幀寫入硬體，並使用雙層迴圈控制cY變化，實現 Julia Set動態畫面。透過按鍵控制動畫開始與結束，提供使用者直覺操作介面。

* drawJuliaSet.s:  
    
  核心功能是計算JuliaSet，並將每個pixel的顏色寫入frame陣列中。程式使用兩層迴圈分別控制 x 與 y 座標，對應畫面每個像素的位置。

數值運算方面，將畫面座標轉換為JuliaSet的複數值 zx 與 zy，透過整數運算方式（如乘以1000再除以常數)來模擬浮點運算，兼顧精度與效能。\_\_aeabi\_idiv函式被用於整數除法，確保運算正確。

在函式開頭，透過 stmfd sp!, {r4-r11, lr} 保存暫存器內容，並在結尾使用 ldmfd sp!, {r4-r11, pc} 恢復，確保函式呼叫返回後暫存器值不受影響。整體設計兼顧暫存器管理、計算精度與記憶體存取，能正確產生 Julia Set 動態畫面。

* drawJuliaSet.s特殊要求：

指定: adds lr, sp, pc

解決: mov r0, lr 備份再存回

* 除法實作：

一張含有 綠色, 對稱 的圖片

自動產生的描述專題中使用了除法函式庫(\_\_aeabi\_idiv)來完成整數除法運算，以符合計算JuliaSet所需的精度與效率。在使用除法函式庫時，仍需注意負數處理與運算順序，確保結果正確，避免畫面出現錯誤或破碎的像素。透過觀察結果，理解到即便使用函式庫，組合語言中暫存器保存與參數傳遞也很關鍵，錯誤的暫存器管理會導致除法結果不如預期，就如同下面的圖片。

四、討論

一張含有 螢幕擷取畫面, 地圖, 鮮豔 的圖片

自動產生的描述一張含有 螢幕擷取畫面, 鮮豔 的圖片

自動產生的描述成果圖:

一張含有 螢幕擷取畫面, 鮮豔, 地圖 的圖片

自動產生的描述一張含有 地圖, 螢幕擷取畫面 的圖片

自動產生的描述一張含有 螢幕擷取畫面, 地圖, 文字 的圖片

自動產生的描述

執行結束並成功輸出。

* 一張含有 文字, 螢幕擷取畫面, 軟體, 電腦圖示 的圖片

  自動產生的描述Frame陣列記憶體區塊:

#起始位置:0xbef69468

#結束位置:0xbefff468

運算方式:

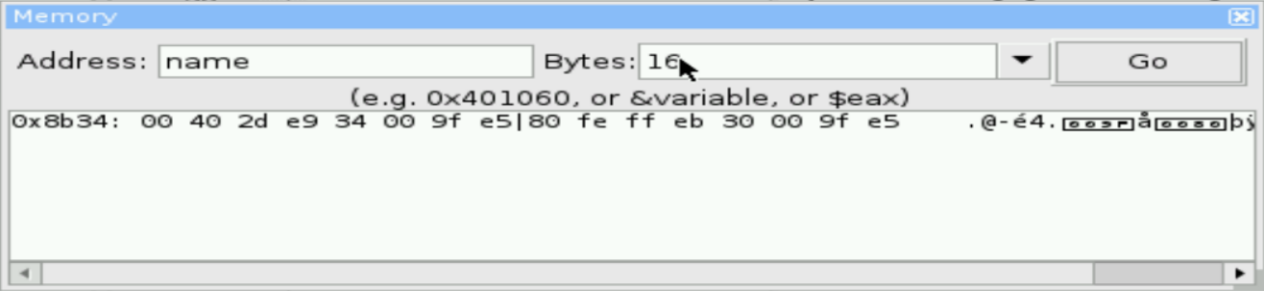
640(weight) \* 480(height) = 307200

#256色中: 1 像素 = 1 byte, 但本次宣告int16\_t = 2 byte

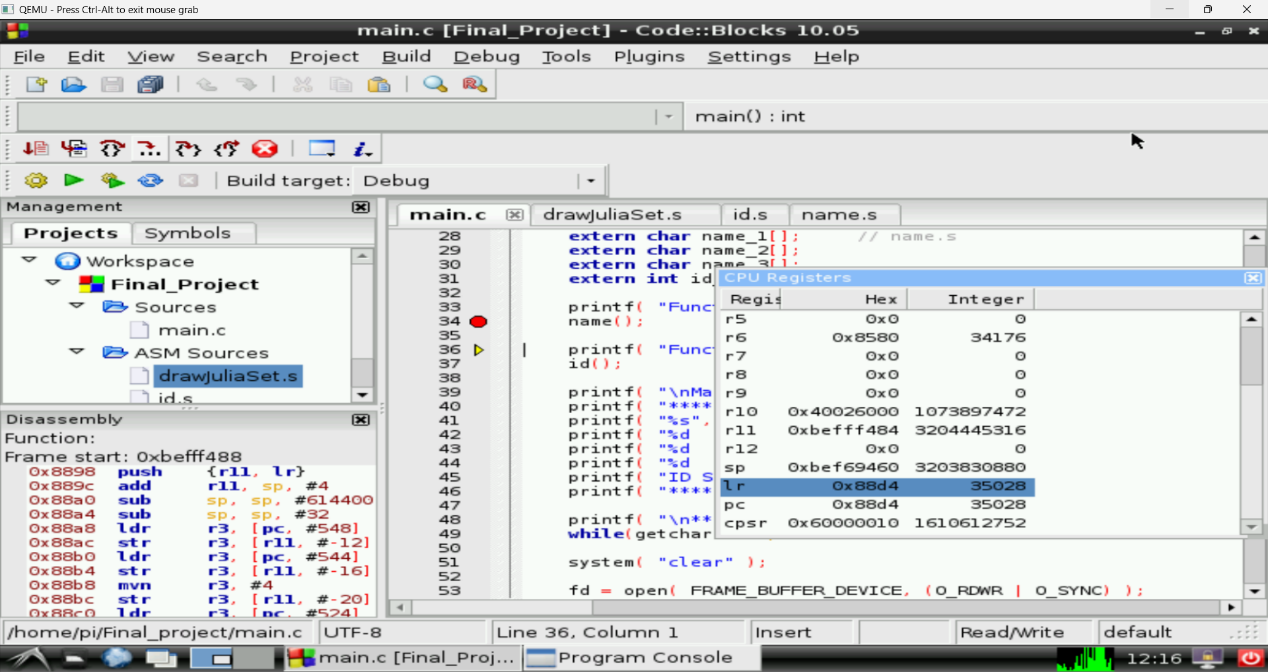
30720 \* 2 = 614400

614400轉16進位 = 0x96000

0xbef69469 + 0x96000 = 0xbefff468(結束位置)

* Name函式的所在位置

Name函式的所在位置: 0x8b34

* Name函式的Return Address

Name函式的Return Address: 0x88d4

* Id函式的所在位置

一張含有 文字, 行, 字型, 螢幕擷取畫面 的圖片

自動產生的描述

Id函式的所在位置: 0x8760

* Id函式的Return Address

一張含有 文字, 軟體, 電腦圖示, 多媒體軟體 的圖片

自動產生的描述

* Id函式的Return Address: 0x8854
* drawJuliaSet函式的所在位置

一張含有 文字, 行, 字型, 螢幕擷取畫面 的圖片

自動產生的描述

drawJuliaSet函式的所在位置: 0x862c

* drawJuliaSet函式的Return Address

一張含有 文字, 軟體, 電腦圖示, 多媒體軟體 的圖片

自動產生的描述

drawJuliaSet函式的Return Address: 0x86f0

五、結論

1.整合 C 與 ARM 組合語言能力提升:

成功將Midterm Project的name與id函數整合到期末專題中，並以組語實作drawJuliaSet函數，熟悉函式呼叫、暫存器管理與 stack 使用。

2.掌握低階運算與暫存器管理:

透過 Julia Set 的雙層迴圈與 while 迴圈運算，實際操作資料處理指令與 conditional execution，理解暫存器保存、stack offset 計算的重要性。

3.除錯與實作經驗:

在撰寫過程中解決了暫存器覆蓋、記憶體位址錯誤與除法運算結果不正確等問題，加深對組合語言程式流程的理解。

4.專題設計與系統整合能力:

經歷midterm project到final project，理解大型專題的程式模組化設計、資料傳遞、與多函式整合的重要性。

六、未來展望

* 精準度提升：

目前為了運算效率多採用定點數運算，這在某些縮放層級下會出現失真。未來展望能導入高精度的浮點數運算邏輯，解決在極細微觀察下的圖形破碎問題，使分形圖形的細節更加精緻。

* 互動式介面開發：

預計加入中斷處理機制，讓使用者能在程式執行期間透過鍵盤即時輸入不同的參數值，動態觀查 Julia Set 在不同複數平面上的連續演化過程，提升程式的教育與互動價值。

七、心得感想

       在這次期末專題中，要實作一個具有實際圖形輸出的程式。與期中只在組合語言中撰寫小型功能不同，這次的專題需要考慮整體程式架構、函式介接方式，以及與作業系統硬體裝置的互動，也是一個相當具有挑戰性的挑戰。

在實作過程中，我最大的收穫是對ARM EABI呼叫慣例有更實際的理解。過去只是在課堂上知道參數會放在 r0 到 r3，其餘放在 stack，但在實際撰寫 drawJuliaSet時，必須自己正確地保存暫存器、計算stack offset來取得 frame buffer的位址，否則畫面就會顯示錯誤甚至造成程式當掉。這讓我們真正體會到暫存器管理與stack操作在組合語言中的重要性。

在除錯方面，也遇到不少問題，例如暫存器值被覆蓋、stack 位址計算錯誤，或是除法運算結果不如預期。這些錯誤在C語言中較容易被編譯器保護，但在組合語言中必須完全由程式設計者自行負責。透過一次次的測試與修正，我對程式實際在CPU中執行的流程有了更清楚的認識。

八、各組員分工方式與負責項目

11327137 林宇宸:

報告撰寫、drawJuliaSet.s校正、main.c實作

11327144 莊有隆:

報告撰寫、drawJuliaSet.s實作、main.c校正