DrawJuliaSet 報告

負責部分: main.c、drawJuliaSet.s、報告文件

一、背景

1.Name: 規劃四個記憶體區塊，分別存放組別與組員的英文姓名，並印出。

2.ID: 規劃四個記憶體位址，作為輸入學號與其總和的緩衝區，以輸入的方式，輸入三個學號，最後按下 p 鍵，即分行印出完整的組員學號與學號總和。

3.drawJuliaSet:計算並儲存 frame 二維陣列每個元素的值，由此決定投影至畫面上的 Pixel 顏色。

4.main:在 MAIN 中呼叫 NAME 和 ID 函數，分別達成前兩支函數的分項功能，按下 p 鍵，使用迴圈呼叫 drawJuliaSet 函數計算並畫出 JuliaSet 的動態畫面，最後輸出完整的組別、組員資訊及組員學號。

二、方法

1.Name:
程式說明:

(1)宣告兩個 label—name1, name2(配置記憶體空間)，分別存放組員英文姓名。將 name1 load 到 r0, 並直接印出，再將 name2 load 到 r0, 直接印出，最後是將 name3 load 到 r0, 直接印出。
Load r3 = namemsg2(字串”End Print”的地址)，令 r4 = -1， 然後執行指定指令(sbcs r0, r3, r4)，之後判斷 N 是否為 0，若是，則印出字串”End Print”。

(2)利用 sbc r1, r3, #-1 將 r1 的值設為字串”End Print”的地址，再執行 subs r1,r1,r0，設定 CPSR flag，利用條件執行判斷，若為 eq(r1 = 0)則將 r0 設為 r1，若為 ne(r1 != 0)，則將 r0 設為 0。

設計重點:
利用條件執行來檢查是否有印過字串”End Print”，若最後 return 0，則代表有印出，若不是 return 0 則沒有印出。
寫四個 function: Getname1,Getname2,Getname3,Getteam 負責回傳姓名及組別，在 main.c 中，可以使用這些 function 來取得組員與組別資訊。

2.ID:
程式說明:

用 scanf 將輸入的組員學號存進 label—id1 id2 與 id3 中，將輸入的 command

存入 label--command 中，設一個 label—p，裡面放的是字元 p，把這兩個 label 的值拿去比較，若相同則利用 r0 跟 r1 以及 r4 使用跟之前不一樣的定址方式將所有 id 加總起來，再用 str 將算出來的加總值存到 label—sum 裡，再依序印出 id1, id2 與 id3 以及總和 sum。

設計重點:
設四個 label，分別寫了四個 function 負責回傳 id 及總和所在的記憶體位置，在 main.c 中就能得到 id 及總和。

3.drawJuliaSet:
程式說明:
(1)使用 sp 先配置記憶體空間給參數以及區域變數，以 r4 作為 i，r5 作為 x，r6 作為 y。

(2)for1:進入 for1，將 x 與 width 比較，若 x 大於 width 則跳至 end，若小於，則跳至 for2。

(3)for2:將 y 與 height 做比較，若 y 大於 height，則將 x 加一，並跳出 for2 至 for1；若小於則將 r0 設為 1500*(x-(width>>1))，再將 r1 設為 (width>>1)，使用外部函式將 r0/r1 的值放入 r0，此時將 r0 存入 sp 中。將 r0 設為 1000*(y-(height>>1))，r1 設為(height>>1)，一樣使用除法將 r0/r1 的值存入 r0，最後將 i 設為 maxIter 的值，跳至 while。

(4)while:先判斷 i 是否大於零，若否則跳至 afterwhile，是則再判斷 zx*zx+zy*zy 是否小於 4000000，若否則跳至 afterwhile，是則先將 ip 設為(zx*zx-zy*zy)/1000+Cx，存入 sp 中(此為 zx)，將 ip 設為 (2*zx*zy)/1000+Cy，存入 sp 中(此為 zy)，最後將 i 減一，跳回 while。

(5)afterwhile:將 i 以 halfword 存入 sp 中，再取出 halfword 的 i，將 ((i&0xff)<<8)|(i&0xff)的值存入 r0,再取 1 補數存入 r1，將 r1&0xffff 的結果存入 r0，最後將此結果放入陣列中。執行完後跳回 for2。

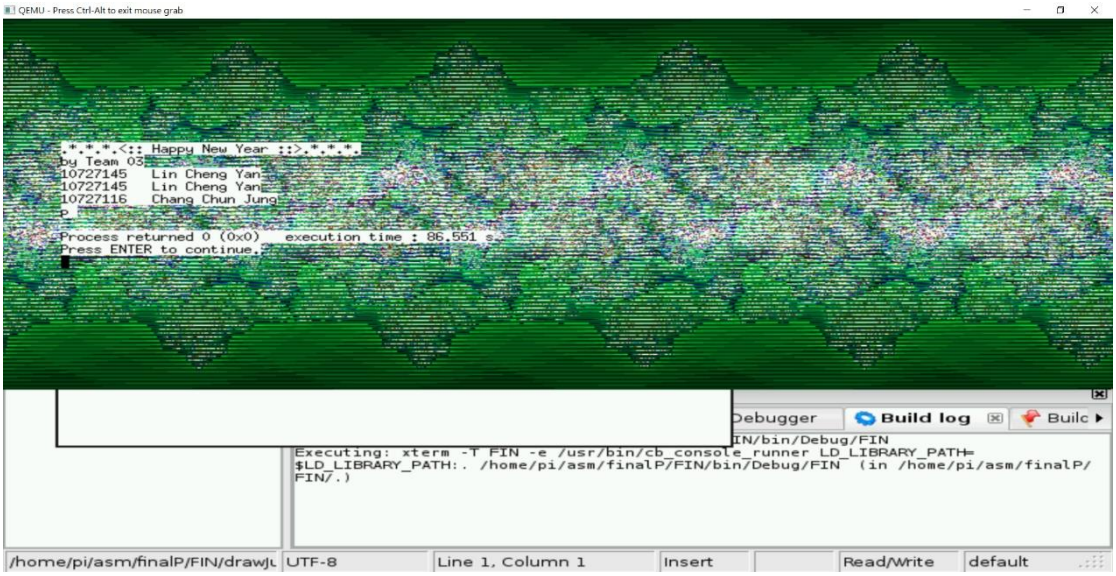
設計重點:
將 zx, zy, Cx, Cy, width, height 等資料都存入 sp，這樣在計算的時候就不容易出錯，也可以隨時存取資料，並使用外部函式來做除法，且除了呼叫除法使用 bl 以外，其他迴圈皆使用 b 來跳迴圈，可避免改到 lr 而回不到 main 裡。

4.main:
程式說明:

呼叫 NAME 與 ID 函數，分別執行完之後，再依序呼叫 Getid 及 Getname 等 function，得到組別、學號、姓名及學號總和等資訊，將這些資訊印出，再使用迴圈呼叫 drawJuliaSet 函數，計算並儲存 frame 二維陣列每個元素的值，最後印出 JuliaSet 動態畫面以及組別學號姓名。

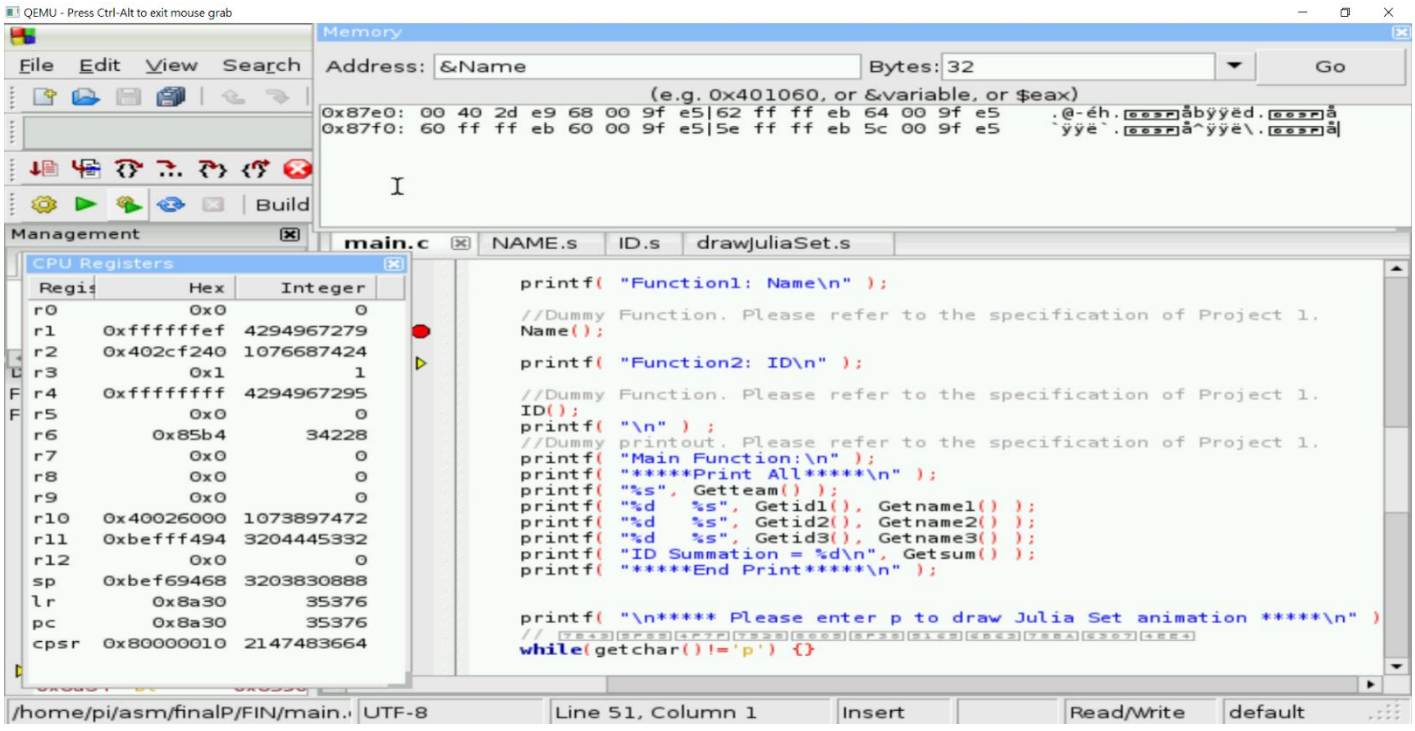
設計重點:
以上所需資訊皆使用呼叫函數的方式得到，故相關 function 皆設為 global。

三、結果

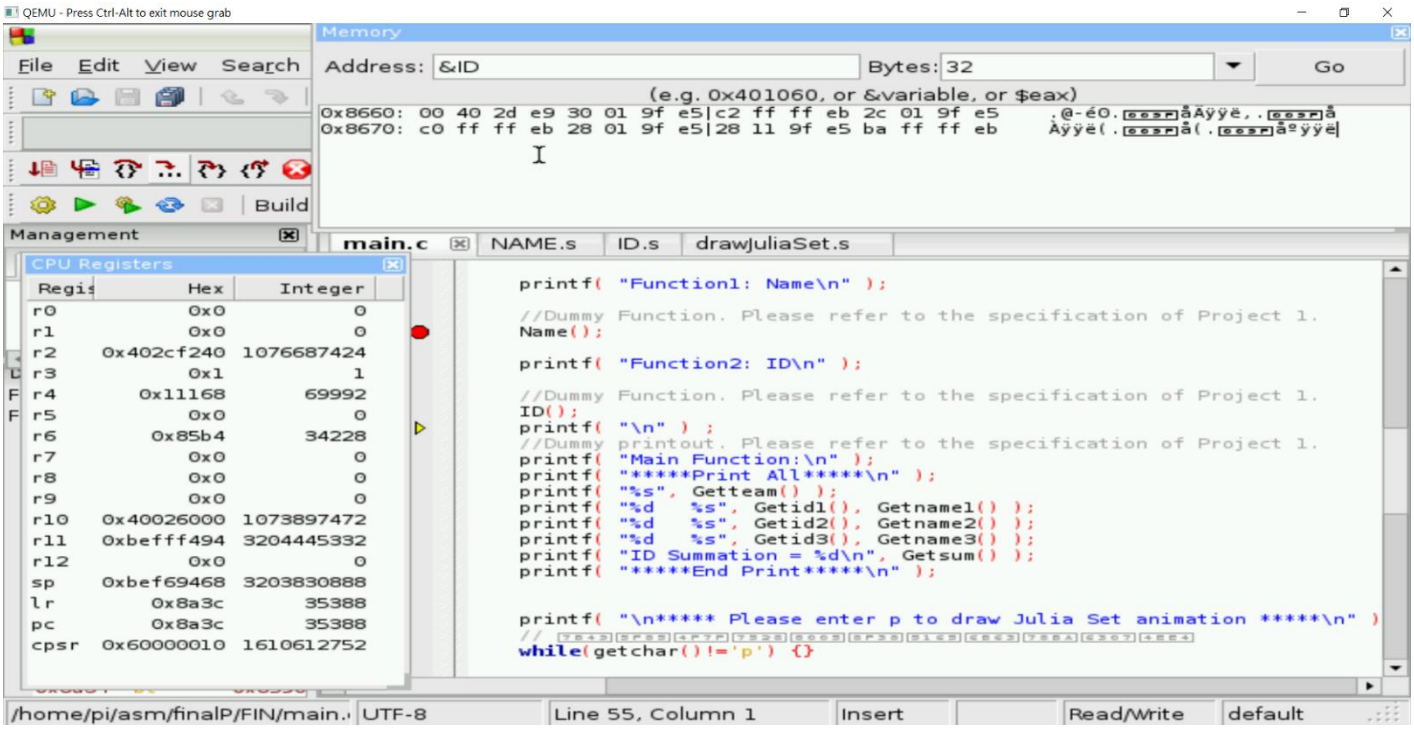


四、討論

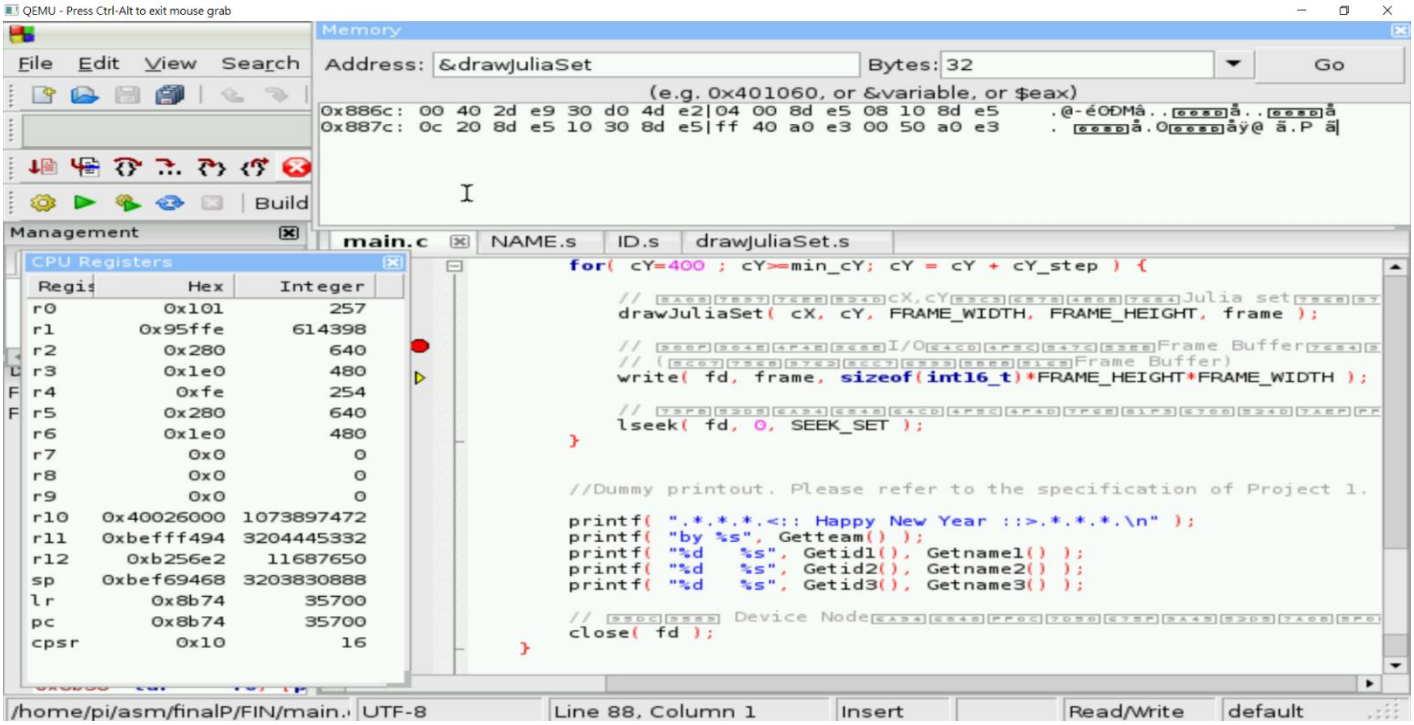
Name 所在記憶體位址:0x87e0,返回位址:0x8a30



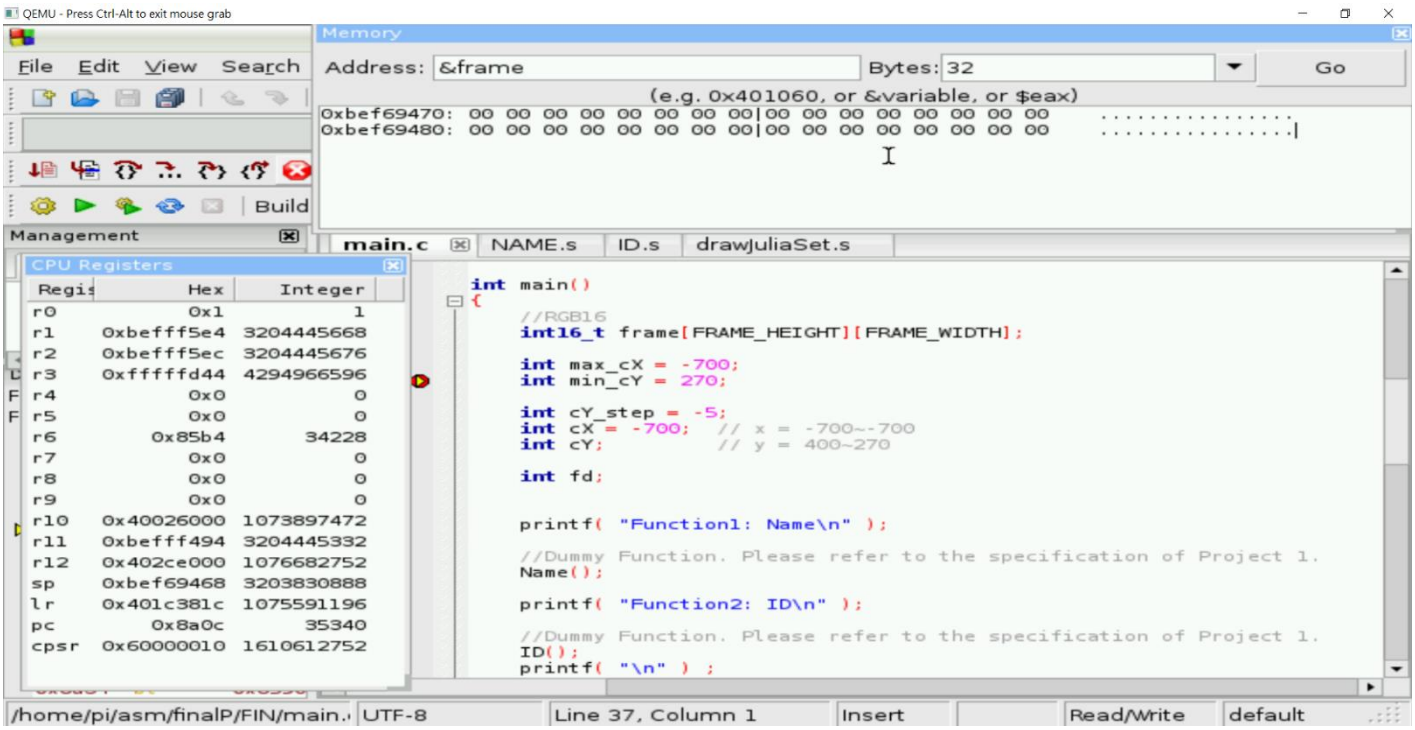
ID 所在記憶體位置: 0x8660, 返回位址:0x8a3c



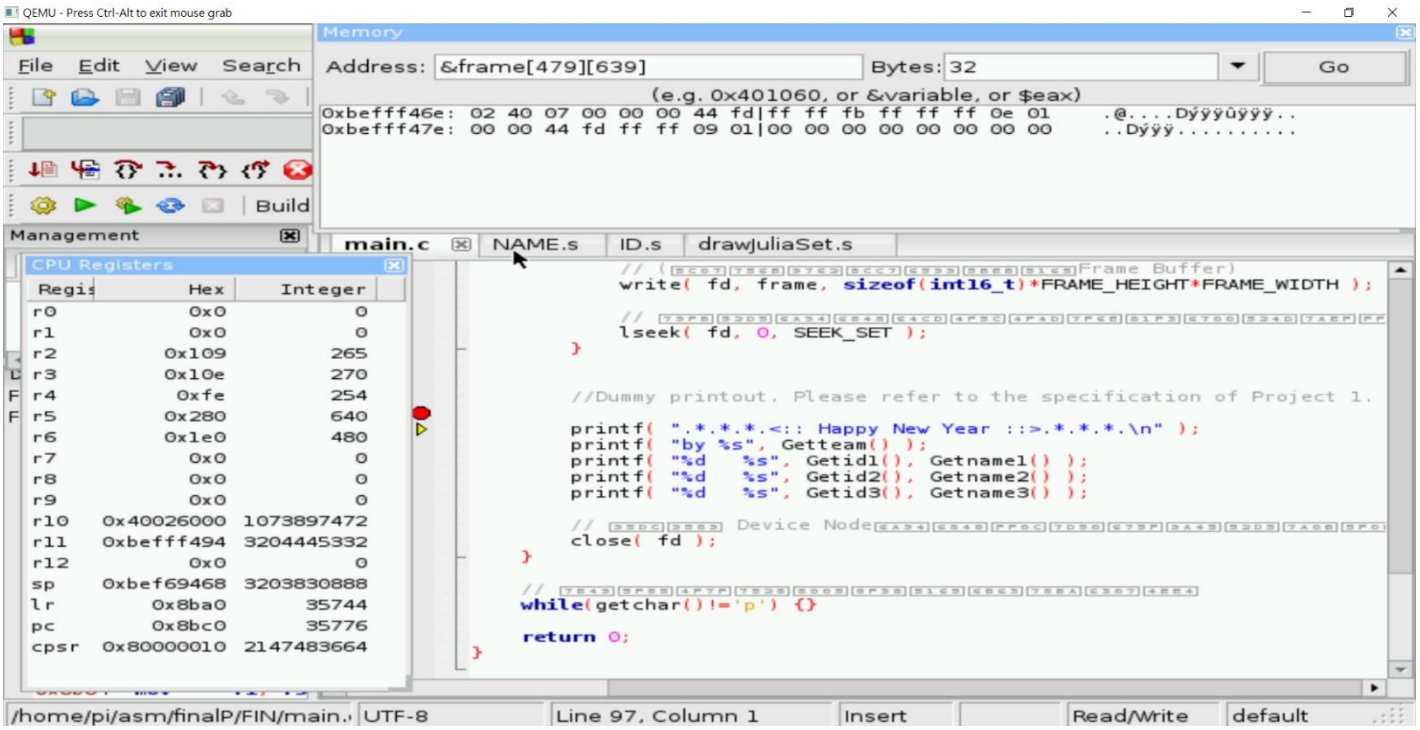
drawJuliaSet 所在記憶體位址:0x886c, 返回位址:0x8b74



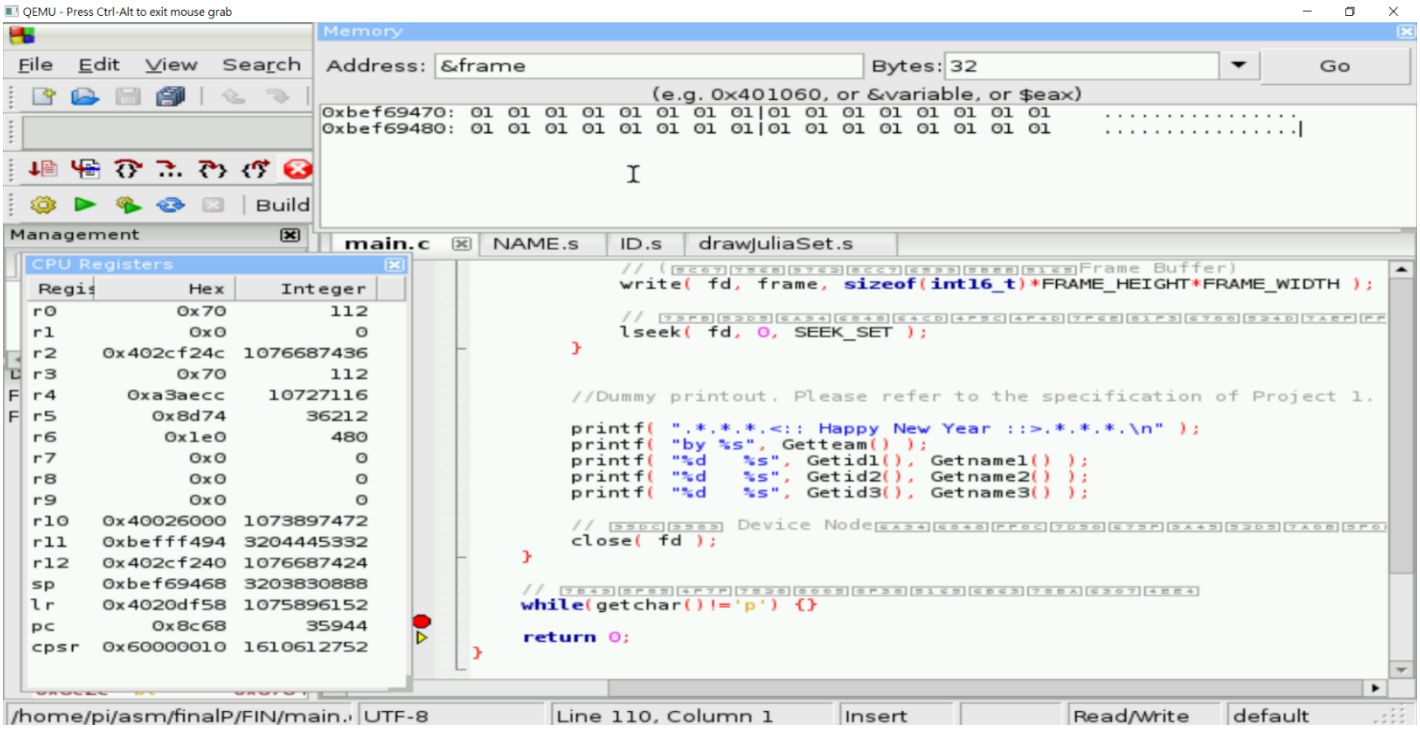
frame 陣列初始記憶體位置: 0xbef69470



frame 陣列結束記憶體位址:0xbefff46e



frame 記憶體區塊內容: 其內容為 Pixel 顏色



五、結論

我們在寫 drawJuliaSet 之前，先看了機轉後的程式碼，發現其實機轉的程式碼沒有比較簡單，因為一開始 r0 應該為 Cx，也就是-700，但是機轉卻將 r0 設為 700，到目前為止我們還是不清楚為什麼要這樣做。我們直接照著 C code 寫，本來我們有自己寫了除法的 function，但是發現跑起來會跑太久，所以最後才參考機轉程式碼，使用了外部除法函式。再來因為我們之前寫程式時，不常使用邏輯運算，所以在計算 color 的時候花了一些時間先理解 C code 在幹嘛，剩下的就比較沒有什麼困難。