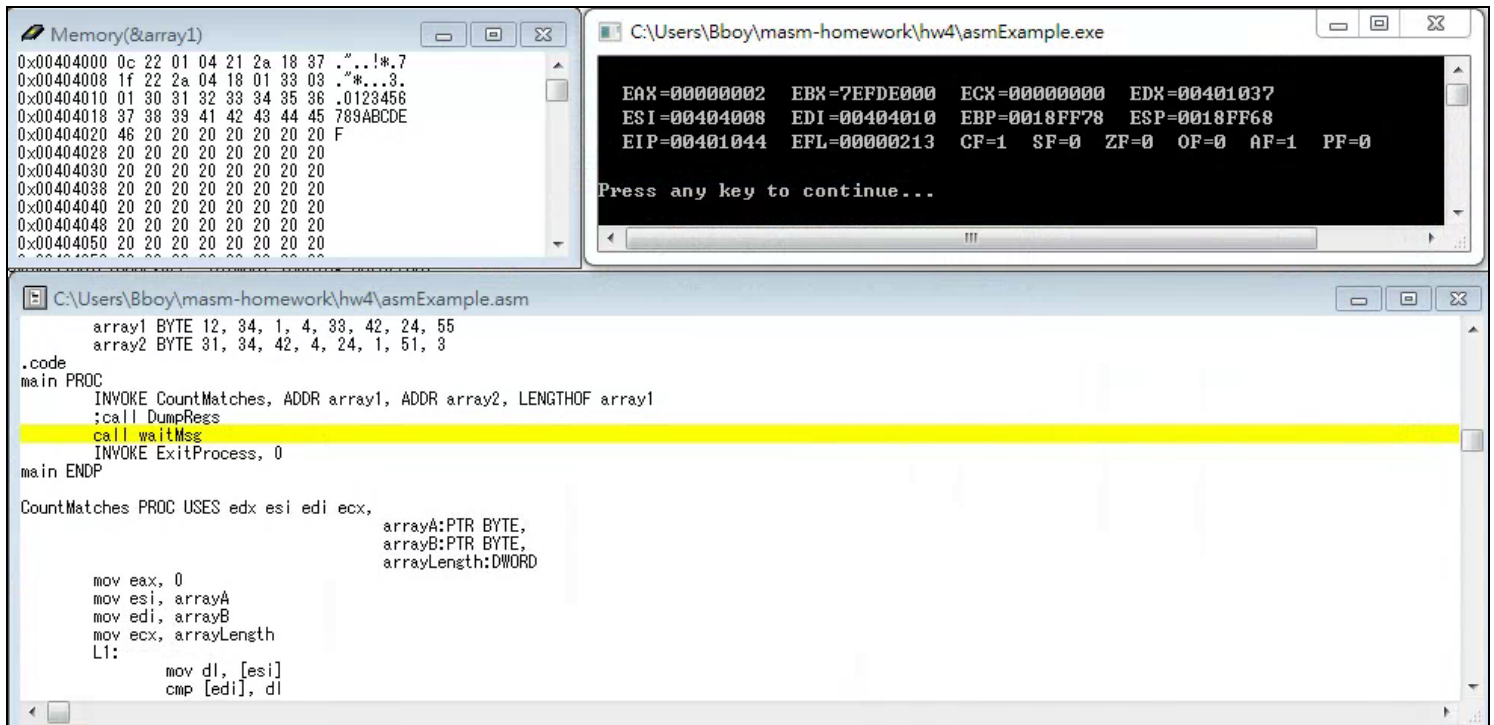


組合語言與系統程式 作業四

資工2B 102502559 吳承霖

程式結果截圖：



由截圖我們可以看到，array1以及array2的開頭分別位於記憶體位置0x00404000以及0x00404008，程式執行到最後時，我有將Registers的值Dump出來，圖右上角即是各個暫存器的值，我們可以看到EAX的值為2，符合題目要求。

程式碼相關解釋：

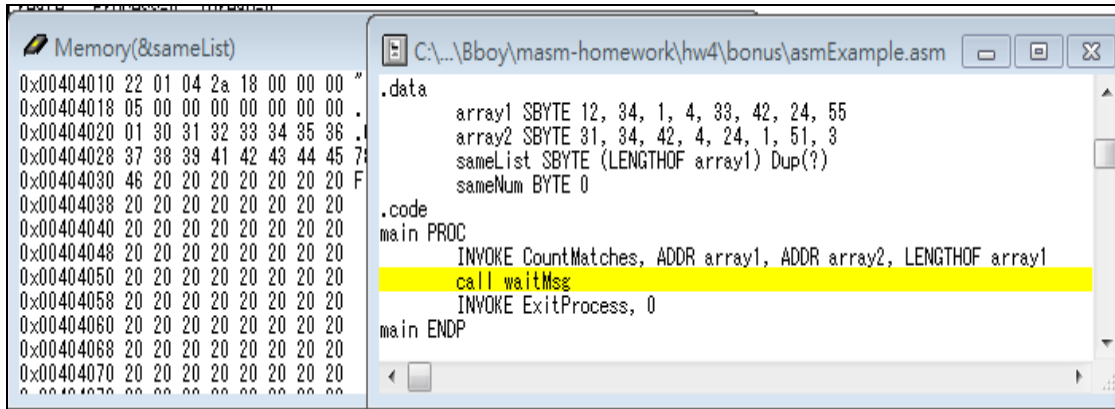
這是程式的核心邏輯部分，CountMatches這個Procedure呼叫時會帶入三個參數，分別為兩個array的開頭位址，還有array的長度。運作原理為將兩個指標暫存器指向的值進行compare，判斷compare動作產生的旗標。如果ZF為1，則代表兩數相等，利用Jump動作進行條件判斷，每一次發現相等的兩數時，則把EAX暫存器的值+1。

```
24 CountMatches PROC USES edx esi edi ecx,
25                                     arrayA:PTR SBYTE,
26                                     arrayB:PTR SBYTE,
27                                     arrayLength:DWORD
28
29     mov eax, 0
30     mov esi, arrayA
31     mov edi, arrayB
32     mov ecx, arrayLength
33     L1:
34         mov dl, [esi]
35         cmp [edi], dl
36         jz EQUAL
37         jmp FORWARD
38     EQUAL:
39         inc eax
40     FORWARD:
41         inc esi
42         inc edi
43         loop L1
44     call DumpRegs
45     ret
46 CountMatches ENDP
```

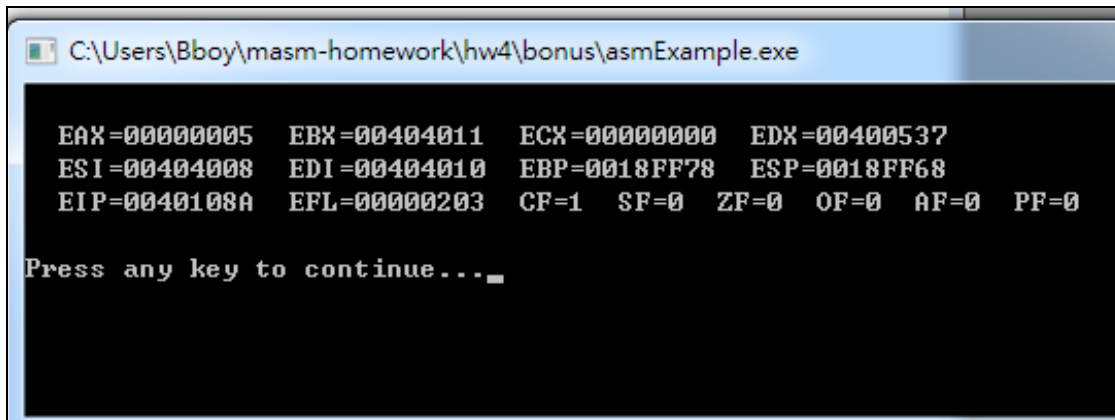
Procedure的最後把Register的值倒出來顯示在Console上，然後return回main procedure。

Bonus 加分題

程式結果截圖



The screenshot shows a debugger window with two panes. The left pane, titled 'Memory(&sameList)', displays a memory dump starting at address 0x00404010. The right pane shows the assembly code for 'asmExample.asm'. The code includes data definitions for 'array1', 'array2', 'sameList', and 'sameNum', followed by a code section with a 'main' procedure. The 'main' procedure calls 'CountMatches', 'waitMsg', and 'ExitProcess'. The 'call waitMsg' instruction is highlighted in yellow.



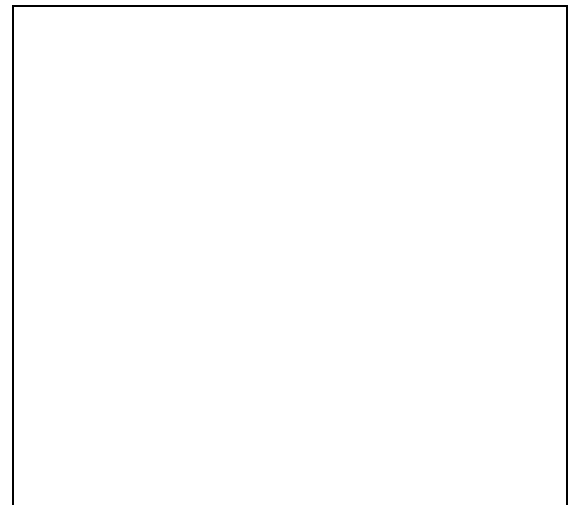
The screenshot shows a console window titled 'C:\Users\Bboy\masm-homework\hw4\bonus\asmExample.exe'. It displays the values of various registers: EAX=00000005, EBX=00404011, ECX=00000000, EDX=00400537, ESI=00404008, EDI=00404010, EBP=0018FF78, ESP=0018FF68, EIP=0040108A, EFL=00000203, CF=1, SF=0, ZF=0, OF=0, AF=0, PF=0. Below the register values, it says 'Press any key to continue...'. The value of EAX (00000005) matches the requirement mentioned in the text.

記憶體內容的部分，這裏顯示的是變數sameList和變數sameNum的內容，sameList為一個跟array1長度相同的陣列，負責存放題目的兩個陣列中有哪些相同的數，在此圖中sameList為8的Byte的陣列，記憶體位置由0x00404010到0x00404017；至於sameNum，則儲存有幾種數在兩陣列中都有出現，也就是題目的要求，在此圖中，sameNum的記憶體位址為00404018。

另外在Console有把記憶體內容印出來，我們可以看到EAX=00000005，符合題意。

程式執行原理

在主要邏輯Procedure CountMatches中，ESI暫存器會依序指向每一個array1的Element，不過再跟array2比較是否有相同數之前，我先檢查目前ESI暫存器指向的值是否之前已經比較過，如右圖，DL暫存器是存ESI所指向的值，DL會依序跟sameList裡面的每一個Element做比較，如果發現已經比較



過，則ESI加一，用array1的下一個元素繼續進行跟array2做比較的動作。

而每一個array1元素跟array2元素比較的程式碼，如右圖所示，仍然利用compare指令對array2的每一個元素進行比較，跟上一個程式的不同在於：如果發現有相同的值，則把這個值存進sameList記錄起來(以免下一次又重新比對一次)，另外也把sameNum加一，代表多一個數值在兩陣列中都有出現。

```
L2:
    mov ebx, OFFSET sameList
    cmp [ebx], dl
    jz  DUPLICATE
    jmp CONTINUE
DUPLICATE:
    pop ecx
    jmp FORWARD
CONTINUE:
    inc ebx
LOOP L2
```

心得

這次作業有兩支程式，第一支程式很快就做完了，用的是以前教過的Procedure概念和邏輯處理。不過在寫加分題的時候卻卡了很久，原因是因為不知道如何可以設計出非常快且有效率的比對搜尋方法，最後萬不得已之下，只好多花費一些記憶體位置來做比對的紀錄。透過撰寫這次加分題，在腦內不停地思考稍微複雜的結構如何設計，有較多的Jump和巢狀迴圈，實在是一大挑戰，利用這次作業磨練出這麼多，覺得非常有成就感！