

組合語言與系統程式第八週上機實習報告

組別16 102502559吳承霖、102502557林唐正

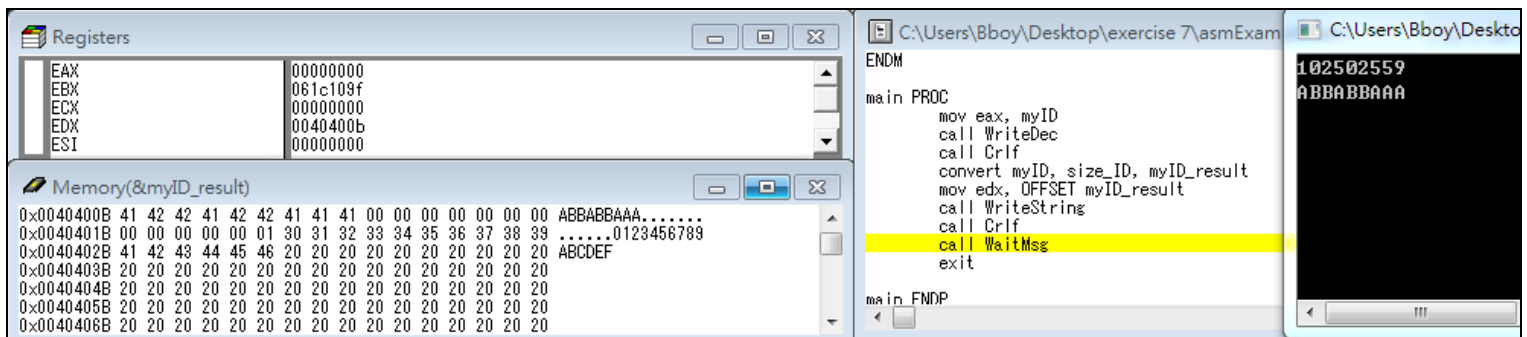
程式執行流程以及原理

這次上機實習主要是練習Macro的應用，Macro與先前教過的PROC稍有不同，Macro僅僅只是把一段程式碼在呼叫處展開並執行，因此並沒有類似return的動作。右圖為此次程式的Macro程式碼片段。

- ❑ 在14行的位置，宣告了這個Macro的名子以及要傳入的參數，並且在第44行的位置ENDM表示Macro的結束。
- ❑ 15行到18行、40行到43行有一連串的push/pop，目的是為了不要在執行這段Macro時，更動到其他暫存器的值，所以先將暫存器的狀態存進stack中，等到Macro執行完後，再將暫存器的值從stack取出。
- ❑ 19行到39行則為本次程式的主要邏輯，esi暫存器指到的是結果變數的記憶體位置，判斷奇偶的功能是利用TEST指令，利用TEST更動的Zero Flag作條件判斷，如果ZF不為1(奇數)則跳至ODD Label處把'A'字元存進esi暫存器指向的記憶體位址；若ZF為1(偶數)則繼續執行接續的程式碼，將'B'字元存進esi指向的記憶體位址。
- ❑ TEST偵測的bit為eax內存的值的最後一位，但以題目的要求我們應該判斷學號(十進位)的每一位，所以在程式內做了一個迴圈L1讓學號每次迴圈都除以十，然後再TEST最後一碼，達到各位元檢查奇偶的效果。

```
14  convert MACRO origin, len, result
15      push eax
16      push ecx
17      push ebx
18      push esi
19      MOV ecx, len
20      MOV ebx, 10d
21      MOV esi, OFFSET result
22      ADD esi, 8
23      L1:
24          MOV eax, origin
25          TEST eax, 1b
26          JNZ ODD
27          MOV byte ptr[esi], 'B'
28          DEC esi
29          MOV edx, 0
30          DIV ebx
31          MOV origin, eax
32          LOOP L1
33      ODD:
34          MOV byte ptr[esi], 'A'
35          DEC esi
36          MOV edx, 0
37          DIV ebx
38          MOV origin, eax
39          LOOP L1
40      pop esi
41      pop ebx
42      pop ecx
43      pop ebx
44  ENDM
```

程式運行結果截圖



以上是程式執行到最後時的暫存器以及記憶體狀態圖，途中最右邊附上Console內印出102502559以及各個位數判斷奇偶產生的序列。

心得

想想其實程式邏輯並不複雜，不過我們這組卻意外地因為小bug卡住了。原因是在DIV ebx的時候，沒有將edx暫存器的值初始化成0，我們忽略了除數是ebx的時候，被除數是兩個ebx的長度，也就是我們除的是EDX:EAX，沒有將EDX暫存器初始化成0的話會產生Overflow的情況。從錯誤中汲取教訓，透過直這次上機實習總算把乘除的概念弄得更清楚，也對日後能練習越來越複雜的組合語言應用感到期待。