

組合語言與系統程式第11週上機實習

第16組 102502559吳承霖、102502557林唐正

程式一 流程原理解說

右圖為本程式的核心邏輯，在main 這個procedure中呼叫了這個procedure五次，分別傳入五組數字比較是否有任兩個數字相同。要執行比較，最少須比較v1和v2、v1和v3、v2和v3三次比較，而比較兩數字是否相等的作法為使用cmp指令，如果兩數相等，則經過cmp後換改變ZF為1，反之如果兩數不相等則ZF為0。比較完之後利用產生的ZF進行條件判斷，如果有任兩數相等，則直接給予eax為0，然後結束此Procedure。

至於呼叫此Procedure並帶入三個參數的呼叫方法為使用INVOKE指令，使用INVOKE指令也必須事先宣告這個Procedure的Prototype。

```
24 DifferentInputs proc,  
25     v1:dword, v2:dword, v3:dword  
26  
27     mov eax,v1      ; 取出 v1  
28     cmp v2,eax      ; 與v2比較  
29     je Label_Equal  ; 相同則跳到回傳0  
30     cmp v3, eax     ; 比較  
31     je Label_Equal  ; 跳  
32     mov eax,v2      ; 取出 v2  
33     cmp v3, eax     ; 比較  
34     je Label_Equal  ; 跳  
35     mov eax,1       ; 回傳1  
36     jmp exit_label  ; return true  
37  
38 Label_Equal:  
39     mov eax,0       ; return false  
40  
41 exit_label:  
42     call DumpRegs  
43     ret  
44 DifferentInputs endp
```

程式二 流程原理解說

附圖為程式的核心邏輯程式碼。31、32行為傳入procedure的參數，pstart為一個指標指在一個string的位置、nChars為要刪除的字元個數。

41到43行為movsb指令的前置準備，把source+nChars指向正確要複製的位置，把destination指向要被刪除的位置(目的地)。

45行的cld指令，意思為clear direction flag，esi和edi在movsb一次後就會加上1，最後47行在字串的最後插上結尾符號null。

```
30 Str_remove PROC,  
31     pStart:PTR BYTE,      ; points to first character to delete  
32     nChars:DWORD         ; number of characters to delete  
33  
34     INVOKE Str_length, pStart  
35     mov ecx,eax           ; 抓到字串長度存放在 ecx 給之後複製迴圈用  
36  
37     .IF nChars <= ecx    ; 檢查移除字元是否超過字串大小  
38         sub ecx,nChars   ; 如果沒有超過，就剪掉移除的字元數量  
39     .ENDIF  
40  
41     mov esi,pStart        ; 設定複製來源字串的起始點  
42     add esi,nChars        ; 將起始點移到正確要複製的位置  
43     mov edi,pStart        ; 將目的起始點設定好  
44  
45     cld                  ; clear direction flag (forward)  
46     rep movsb            ; do the move  
47     mov BYTE PTR [edi],0 ; insert new null byte  
48  
49 Exit_proc:  
50     ret  
51 Str_remove ENDP
```

本次上機心得

這次上機實驗學到了一些對字串的處理方式，字串說穿了也就是一個字元的陣列，所以關於指標和間接定址存取array元素的技巧是很重要的。此外還學到了rep這個指令，搭配direction flag作array的操作比起以前用for迴圈寫是一個方便許多的方法！