

Lab Homework Week 11 Report

Group 66: 102403015 程祥恩、102403016 邱威穎、102403020 曾子軒

目標：請利用本週課程所學內容，完成兩支程式。

第一支程式請完成，將三個輸入 `v1`, `v2`, `v3` 判別出三個數值(byte)是否相異，若三個彼此間完全相異則將 `eax` 設定成 1 否則設定成 0。

第二支程式請完成，將兩個輸入字串 `target1`、`target2` 利用 `movsb` 來移除字串前 `nChars` 個字元。

Different Inputs 程式碼：

exercise11_1.asm - 記事本

檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)

```
TITLE exercisell_1.asm[exercisell_1.asm]
INCLUDE Irvine32.inc

.386

.model flat,stdcall

.stack 4096

ExitProcess proto,dwExitCode:DWORD ;宣告 proto type

DifferentInputs proto, v1:DWORD, v2:DWORD, v3:DWORD ;宣告 proto type

.data

.code

start@0 proc

    invoke DifferentInputs,2,2,3
    invoke DifferentInputs,2,3,2
    invoke DifferentInputs,3,5,5
    invoke DifferentInputs,2,2,2
    invoke DifferentInputs,104522064,102403015,102403016 ; 填上組員學號

    call WaitMsg

    invoke ExitProcess,0

start@0 endp

DifferentInputs PROC, v1:DWORD, v2:DWORD, v3:DWORD

    mov eax,v1          ; 取出 v1
    cmp v2,eax          ; 與v2比較
    je Label_Equal      ; 若相等則跳到Label_Equal,回傳0

    cmp v3,eax          ; 與v3做比較
    je Label_Equal      ; 若相等則跳到Label_Equal,回傳0
    mov eax,v2          ; 取出 v2

    cmp v3,eax          ; 與v3做比較
    je Label_Equal      ; 若相等則跳到Label_Equal,回傳0
    mov eax,1           ; 回傳1

    jmp exit_label      ; return true

Label_Equal:

    mov eax,0           ; return false

exit_label:

    call DumpRegs

    ret

DifferentInputs endp
```

這支程式裡定義了 procedure DifferentInputs：呼叫時需要傳入三個 DWORD 參數 v1、v2 與 v3，首先先將 v1 的值放入 eax 裡，讓 v2 與 eax(此時等於 v1)做比較(cmp)，如果相等就直接跳到(je)Label_Equal 將 eax 的值設為 0 並回傳，不相等就繼續比較下一項 v3 與 eax(此時等於 v1)，也是一樣相等就設 eax 為 0 並回傳，不相等就繼續比下一項，將 v2 的值放入 eax 裡讓 v3 與 eax(此時等於 v2)比較，如果相等將 eax 設為 0、不相等將 eax 設為 1 並回傳。在主程式裡用 invoke 指令傳入參數呼叫 procedure DifferentInputs，看指令跑完後 eax 的值就能知道傳入的參數是否有相同的值。

Different Inputs 程式結果：

```
invoke DifferentInputs,2,2,3  
invoke DifferentInputs,2,3,2  
invoke DifferentInputs,3,5,5  
invoke DifferentInputs,2,2,2  
invoke DifferentInputs,104522064,102403015,102403016 ; 填上組員學號
```

C:\Users\user\Desktop\AssemblyLanguage\WINdbgFolder\exercise11_1.exe

```
EAX=00000000 EBX=00394000 ECX=0040100A EDX=0040100A
ESI=0040100A EDI=0040100A EBP=0019FF70 ESP=0019FF70
EIP=00401091 EFL=00000246 CF=0 SF=0 ZF=1 OF=0 AF=0 PF=1
```

```
EAX=00000000 EBX=00394000 ECX=0040100A EDX=0040100A
ESI=0040100A EDI=0040100A EBP=0019FF70 ESP=0019FF70
EIP=00401091 EFL=00000246 CF=0 SF=0 ZF=1 OF=0 AF=0 PF=1
```

```
EAX=00000000 EBX=00394000 ECX=0040100A EDX=0040100A
ESI=0040100A EDI=0040100A EBP=0019FF70 ESP=0019FF70
EIP=00401091 EFL=00000246 CF=0 SF=0 ZF=1 OF=0 AF=0 PF=1
```

```
EAX=00000000 EBX=00394000 ECX=0040100A EDX=0040100A
ESI=0040100A EDI=0040100A EBP=0019FF70 ESP=0019FF70
EIP=00401091 EFL=00000246 CF=0 SF=0 ZF=1 OF=0 AF=0 PF=1
```

```
EAX=00000001 EBX=00394000 ECX=0040100A EDX=0040100A
ESI=0040100A EDI=0040100A EBP=0019FF70 ESP=0019FF70
EIP=00401091 EFL=00000202 CF=0 SF=0 ZF=0 OF=0 AF=0 PF=0
```

呼叫五次 (五組測資)，前四組都有相同的數值 (2 · 2 · 5 · 2) 因此 eax 的值為 0，第五組因為學號都不同因此 eax 為 1。

String Remove 程式碼：

```
1  TITLE exercisel1_2[exercisel1_2]
2  INCLUDE Irvine32.inc
3
4  Str_remove PROTO,
5  pStart:PTR BYTE,
6  nChars:DWORD
7
8  .data
9  target1 BYTE "102403015",0 ;填入組員的學號
10 target2 BYTE "102403020",0 ;填入組員的學號
11 target3 BYTE "999999999",0
12
13 .code
14 start@0 PROC
15     INVOKE Str_remove, ADDR target1, 5 ;第一個字串移除5個字元
16     mov edx,OFFSET target1
17     call WriteString
18     call Crlf
19
20     mov ebx, OFFSET target2
21     INVOKE Str_remove, ebx, 2 ;第二個字串移除2個字元
22     mov edx,OFFSET target2
23     call WriteString
24     call Crlf
25
26     INVOKE Str_remove, ADDR [target2+1], 15 ; 第二個字串移除超過字串長度的字元
27     mov edx,OFFSET target2
28     call WriteString
29     call Crlf
30     call WaitMsg
31     exit
32 start@0 ENDP
33
34 Str_remove PROC,
35     pStart:PTR BYTE, ; points to first character to delete
36     nChars:DWORD ; number of characters to delete
37
38     INVOKE Str_length, pStart
39     mov ecx,eax ; 抓到字串長度存放在 ecx 給之後複製迴圈用
40     .IF nChars <= ecx ; 檢查移除字元是否超過字串大小
41     | sub ecx,nChars ; 如果沒有超過，就剪掉移除的字元數量
42     .ENDIF
43
44     mov esi, pStart ; 設定複製來源字串的起始點
45     add esi, nChars ; 將起始點移到正確要複製的位置
46     mov edi, pStart ; 將目的起始點設定好
47     cld ; clear direction flag (forward)
48     rep movsb ; do the move
49     mov BYTE PTR [edi],0 ; insert new null byte
50
51 Exit_proc:
52     ret
53 Str_remove ENDP
54 END start@0
```

此程式會在 Main Function 呼叫三次 Str_remove Procedure，每次都會傳入兩個參數，分別是字串指標以及欲刪除的字元數。

第一次呼叫 Str_remove 時以 target1 的位址和 5 作為參數，代表要把

target1 的前五個字元(byte)刪除。此時我們進入 Str_remove Procedure，首先判斷欲刪除的字串長度(nChars)是否大於輸入之字串長度(Str-length、eax->ecx)，若是，則最多只會刪除該字串長度的字元。第 44 行將 pStart，也就是字串的起始位置設給 esi，第 45 行會將 esi 指向「不會被刪除的區塊的起點」，第 46 行則把原先的 pStart 設給 edi，第 48 行呼叫 rep movsb，每次執行這行指令都會使 ecx 減一，並把 esi 設給 edi，直到 ecx 歸零。Ecx 歸零時，所有元素都會往前(nChars)個位址，把前面的元素蓋掉，如此就達到了刪除的效果。

第二次呼叫時，原理和第一次相同

第三次呼叫時，會傳入 target2+1 的位址，因為 target2 的前兩個元素已被刪除，所以傳入的參數是指向 102403020 中的“4”，另一個參數為刪除 15 個元素，但因為 15 大於字串長度 9，所以 Str_remove 會刪除包含 4 以及在其之後的 8 個元素，又因為記憶體位置是連續的，所以會借用 target3 的 9999。跑完程式後，答案即為 2999。

String Remove 程式結果：

A screenshot of a Windows command prompt window. The title bar shows the file path: C:\Users\user\Desktop\AssemblyLanguage\WINdbgFolder\exercise11_2.exe. The command prompt displays the following text: 3015, 2403020, 2999, and Press any key to continue... The text is in a monospaced font, with the first three lines being the output of the program and the last line being the standard Windows prompt instruction.

第一項測資 102403015 刪除前 5 個數字即為 3015，第二項測資

102403020 刪除前 2 個數字即為 2403020，第三項測資將第二項的結果從第二個數字開始因此保留了 2，又因記憶體配置連續因此取到了 target3 的 999，因此結果為 2999。

心得：

這次的作業又複習了如何宣告、定義 procedure 及傳入的參數，傳入的參數還可以使用 PTR 即為傳入一記憶體位址。第一小題還蠻簡單的利用 cmp 與 je 指令搭配實作 if else 的邏輯，call DumpRegs 是第一次用原來可以印出暫存器與 flags 的值。第二題用了今天新學到的指令與技巧相對較生疏，把字串刪減從原字串位址覆寫上新的值有點像在 C 語言做的把陣列縮減整理，把 esi 與 edi 設好搭配 rep movsb 來完成迴圈重複執行的動作還蠻方便的。