

In-Class Exercise # 12 - "String Copy"

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

Objective :

將第一位組員的學號字串複製到第二位組員學號字串の後方並印出結果。

Code :

```
1  TITLE exercise12.asm[exercise12.asm]
2  INCLUDE Irvine32.inc
3
4  Str_copyN PROTO,
5      source:PTR DWORD,    ; source string address
6      target:PTR DWORD,    ; target string address
7      maxChars:DWORD       ; maximum number of characters to copy
8
9  .data
10     string_1 BYTE "102403016" 0 ;填入組員1學號
11     string_2 BYTE "102403020" 0 ;填入組員2學號
12
13  .code
14  start@0 PROC
15      INVOKE Str_copyN,        ; copy string_1 to string_2
16      ADDR string_1,
17      ADDR string_2 + 9,
18      (SIZEOF string_2) - 1 ; save space for null byte
19  ; Display the destination string
20      mov     edx,OFFSET string_2
21      call    WriteString
22      call    Crlf
23      call    WaitMsg
24      exit
25  start@0 ENDP
26
27  Str_copyN PROC USES eax ecx esi edi,
28      source:PTR DWORD,    ; source string
29      target:PTR DWORD,    ; target string
30      maxChars:DWORD       ; max chars to copy
31  ; Copy a string from source to target, limiting the number
32  ; of characters to copy. Value in maxChars does not include
33  ; the null byte.
34
35      mov     ecx,maxChars    ; set counter for REP
36      mov     esi,source      ; assign source
37      mov     edi,target      ; assign target
38      cld                    ; direction = forward
39  ; (clear direction flag)
40      rep     movsb           ; copy the string
41      mov     al,0            ; set null byte
42      stosb                    ; append null byte
43      ret
44  Str_copyN ENDP
45  END start@0
```

Result :

A screenshot of a debugger window. The top line shows a memory address: 102403020102403016. Below it, the text 'Press any key to continue...' is displayed. The rest of the window is black. At the bottom left, there is a small text '微軟注音 半 :'. On the right side, there are vertical scroll bars.

Explanation :

把 string1 第一個值的位址，和 string2 最後一個值的位址傳入 Str_copyN PROC 中，後者即為前者要插入字串的位址。在 Str_copyN PROC 中，先設迴圈次數為字元數目，string1 即為 source 並設定給 esi，string2 + 9 即為 target 並設定給 edi，呼叫 cld 清除 direction flag。再呼叫 rep movsb，根據 ecx 的次數把 string1 每一個元素依序放到 string2 末端後。最後把 al 設為 0，設定 null byte，再呼叫 stosb，把 al 的值放到 target 的末端，就成功把 null byte 加入最終的 string 了。

Review :

這週教的指令 stosb 和 lodsb 能把 esi 或 edi 的 offset 與 eax 做交換，但是不知道這樣跟單純的 mov 有什麼差別？這次的 code 也有出現 cld 這個指令，看起來好像是 esi 和 edi 的升冪排列或降冪排列？想知道輸入這個指令的功能到底是如何呢？最後是在末端增加的 null byte，之前聽助教說這可以幫助電腦知道一個字串的結束，但這個 null byte 真的有必要嗎？