

In-Class Exercise # 12

"Find the Character"

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

Objective :

請利用之前所學內容完成程式。首先利用組員 1 學號最後一碼來找出在包含學號字串中第一次出現這個字元的位置，並且計算在這字元之前有幾個字元，最後存入 edi 且印出暫存器結果，接下來使用之前所學的 String remove 來把這個字元之前所有字元移除掉，並且顯示移除後的字串。

Code :

```
1  TITLE exercise13[exercise13.asm]
2  INCLUDE Irvine32.inc
3
4  Str_remove PROTO,
5      pStart:PTR BYTE,
6      nChars:DWORD
7
8  main EQU start@0
9
10 .data
11 target BYTE "1024030151024030162ABCDEF",0
12
13 .code
14 main PROC
15     mov edi, OFFSET target        ;edi = target起始位置
16     mov al, [edi+8]               ;al = 學號1最後一碼(字元)
17     mov ecx, LENGTHOF target     ;ecx = target 字串的長度
18     cld                          ;clear direction flag
19     repne scasb                  ;repeat searching while not equal ([edi] & al)
20     dec edi                      ;將找到的字元位置減一
21     mov eax, OFFSET target       ;eax = target起始位置
22     sub edi, OFFSET target       ;edi -= target起始位置 (要移除的字元數)
23     call DumpRegs               ;顯示目前暫存器狀況
24
25     mov edx,OFFSET target
26     call WriteString
27     call Crlf
28     INVOKE Str_remove, eax, edi   ;呼叫 Str_remove
29     mov edx,OFFSET target
30     call WriteString
31     call Crlf
32     call WaitMsg
33     exit
34 main ENDP
35
36 Str_remove PROC,
37     pStart:PTR BYTE,             ; 要移除的字串頭
38     nChars:DWORD                 ; 將移除的字元數
39
40     INVOKE Str_length, pStart
41     mov ecx, eax                 ;ecx = 字串長度
42
43     .IF nChars <= ecx            ; check range of nChars
44         sub ecx, nChars          ; set counter for REP prefix
45     .ENDIF
46
47     mov esi, pStart              ;esi = 字串起始位置
48     add esi, nChars              ;esi += 要移除的字元數
49     mov edi, pStart              ;edi = 字串起始位置
50     cld                          ;clear direction flag
51     rep movsb                    ;do the move
52     mov BYTE PTR [edi],0         ; insert new null byte
53
54 Exit_proc:
55     ret
56 Str_remove ENDP
57 END main
```

Result :

```
EAX=00405000  EBX=7FFDE000  ECX=00000011  EDX=0040100A
ESI=0040100A  EDI=00000008  EBP=0019FF94  ESP=0019FF84
EIP=0040103D  EFL=00000202  CF=0  SF=0  ZF=0  OF=0  AF=0  PF=0

1024030151024030162ABCDEF
51024030162ABCDEF
Press any key to continue...
```

Explanation :

在 main function 中，先把 target 的起始位置設給 edi，並把學號 1 的最後一碼設給 al，之後呼叫 scasb，掃描 target 的第一個元素到學號 1 的最後一字，若掃描到字元使兩者相等，就把 edi 減一(因為 index 是從 0 開始算，減一才能算出正確欲刪除的字元數)，把 target 初始位置設給 eax，再拿 edi 位置減去 target 初始位置，這樣即可得出欲刪除的字元數目，最後把 eax 和 edi 作為參數傳入 Str_remove Procedure 之中。

在 Str_remove Procedure 中，先將字串長度設給 ecx，作為迴圈的執行次數，接著執行 if，判斷 ecx 是否大於欲刪除的字元數 nChars，若是，則將 ecx 減去 nChars，可以減少執行的次數。最後把 esi 指向字串起始位置，並向前移動欲刪除的字元數，呼叫 movsb，將 esi 指向的值複製給 edi 相對應的位置，這即是複製貼上的概念，把欲刪除的子字串給蓋掉，最終就會達成我們要的結果了。

Review :

這次作業中，讓我們比較納悶的是在 Str_remove Procedure 中的 INVOKE Str_length, pStart 指令，INVOKE 不是在需要傳入參數的函式呼叫時使用嗎？但在程式碼中好像並沒有 Str_length Procedure，後來我們測試時也把這行註解掉，結果看起來也沒有差異，想知道這一行指令到底有什麼功能呢？

另外還有 if 區塊的 sub 函式，我們猜測這可以幫助程式減少「刪除字元」的執行次數，但觀察暫存器時似乎又跟註解掉這一區塊沒什麼差異，所以也想問問助教或老師，這裡的 if 區塊有什麼功用呢？