

HW#05 FIND DELIMITER

資管 3A 102403015 程祥恩


Code:

```

1  TITLE hw5[hw5.asm]
2  INCLUDE Irvine32.inc
3
4  Str_nextWord PROTO,
5      pString:PTR BYTE,    ;pointer to string
6      delimiter:BYTE       ;delimiter to find
7
8  .data
9      testStr BYTE "ABC\DE\FGHIJK\LM",0
10
11  .code
12  Str_nextWord PROC,
13      ptrString: PTR BYTE, delimiterChar: BYTE
14      mov al, delimiterChar
15      mov edi, ptrString
16  L2:
17      mov dl, [edi]
18      inc edi
19      cmp dl, 0          ;end of the string?
20      je L4
21      cmp dl, al         ;meet the delimiter?
22      jne L2
23  L3:
24      mov BYTE PTR [edi-1], 0
25      mov eax, [edi]
26      jmp Exit_Proc
27  L4:
28      OR al, 1          ;clear ZF
29  Exit_Proc:
30      ret
31  Str_nextWord ENDP
32
33  start@0 PROC
34      call Clrscr
35      mov edx, OFFSET testStr ;display starting string
36      call WriteString
37      call Crlf
38      ;Loop through the string, replace each delimiter, and
39      ;display the remaining string.
40      mov esi, OFFSET testStr
41  L1:
42      INVOKE Str_nextWord, esi, "\" ;look for delimiter
43      jnz Exit_prog              ;quit if not found
44      mov esi, edi               ;point to next substring
45      mov edx, esi              ;edx must be a pointer
46      call WriteString          ;display remainder of string
47      call Crlf
48      jmp L1
49  Exit_prog:
50      call WaitMsg
51      exit
52  start@0 ENDP
53  end start@0

```

Result:



```
ABC\DE\FGHIJK\LM
DE\FGHIJK\LM
FGHIJK\LM
LM
Press any key to continue...
```

Explanation:

在 Str_nextWord PROC 中，先把字串的 offset 設給 edi，同時也把分隔符號設給 al。第 17 行將 edi 指向的值，也就是 string 的第一個值設給 dl，方便之後比較用，inc edi 則是將字串指向下一個字元。第 19 行判斷是否到字串的末端了，若是，則執行 L4 將 zero flag 消去。第 21 行比較字元與分隔符號，如果該字元等於分隔符號，則執行 L3，把多加一次的 edi 檢回來後設給 eax 後跳出 PROC；若不等於，則執行 L2 迴圈直到字元等於分隔符號或是字串結束。Main 的部分則是把 edi 設給 edx，如此就能透過 call WriteString 顯示在 cmd 上了。

Review:

這次作業對我來說，實在是一個磨練耐心的好機會…首先是在第 42 行中，不知道為什麼在編譯期間一直跳出 invalid characters 的語法錯誤，明明我是複製 PPT 上的，也重打了很多次，甚至把變數、註解改得更單純，但是不給過就是不過，一段時間後刪除那一行再重打一次，不知道為什麼就通過編譯了…。在 Procedure 之中，一開始我不知道如何在跳出 Procedure 時控制 Zero Flag 的開關，花了不少時間想辦法，後來上網查後才發現可以透過 OR 指令的特性，搭配 al 暫存器強制修改 Zero Flag 的值(1 或 0)，後來我也想到，或許也可以透過 TEST 來強制修改 Zero Flag？