

# Report: BONUS

CSIE Grade.2B 108502571 Paul

- **Code Flows**

(1) 宣告一個字串叫 testStr

```
10 .data
11 testStr byte "ABC\DE\FGHIJK\LM", 0
```

(下一頁)

(2) main 函式主要是將每次處理前的 testStr 印出，然後決定如何結束函式

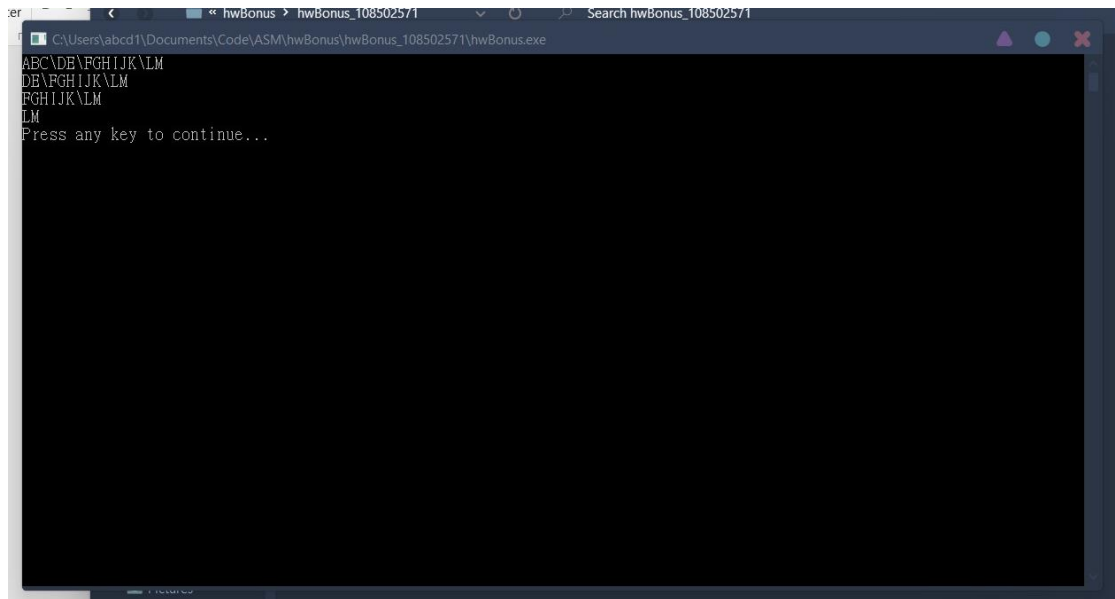
細節：18 行把畫面清除後，先把 testStr 印出一次，接下來呼叫 Str\_nextWord 函式，函式會將結果從 eax 傳回，並且決定 ZF，26 行判斷 ZF 如果 unset 表示迴圈結束，跳至 Exit\_prog，則繼續執行 27 行，將要印出的記憶體指標位置給 edx，28 行將值也交給 esi，作為下次執行迴圈的 string 位置

```
13  .code
14  main    EQU      start@0
15  main    PROC
16
17
18          call    Clrscr
19          mov     edx, OFFSET testStr
20          call    WriteString
21          call    Crlf
22
23          mov     esi, OFFSET testStr
24
25  L1:     INVOKE   Str_nextWord, esi, "\"
26          jnz     Exit_prog
27          mov     edx, eax
28          mov     esi, edx
29          call    WriteString
30          call    Crlf
31          jmp     L1
32
33  Exit_prog:
34          call    WaitMsg
35          exit
36
37  main    ENDP
```

(3) 這個函式主要做的有二：一是當執行到 pString 的尾時，就利用 52 行將 ZF unset，二是當讀取的字等於分界符時，回傳分界符下一個的 pString 位置，如果都沒有就換讀取下一個字

```
39 Str_nextWord PROC USES esi edi,
40     pString: ptr byte,
41     delimiter: byte
42
43
44     mov     edi, pString
45     mov     eax, 0      You, an hour
46
47 L1:     mov     al, [edi]
48         inc     edi
49
50         ; end loop in main by ZF=0
51         .IF     al == 0
52             or     al, 1
53             ret
54         .ENDIF
55         .IF     al == delimiter
56             mov     eax, edi
57             ret
58         .ENDIF
59         jmp     L1
60
61 Str_nextWord ENDP
62
63 END main
```

## ● **Finished**～～



## ● **Review**

這個 bonus 有點難，難在有 ZF 的處理和要回傳讀取 testStr 到的位置給 main 函式，一開始找了好久，找到了可以藉由改變 eax 改換 ZF 的方法，之後才發現好像 scasb 指令也行，不過已經寫好了就不想改了；記憶體位置的部分主要要調整 inc 的位置，如果像以前一樣放在 jmp (loop) 前一行回傳的位置就會在分界符的位置，就會陷入 main 裡的無窮回圈。

還蠻有意思的，這題目。只是因為是組合語言所以需要了解指令的每個功用，不過卻讓我有之前第一次剛碰程式語言那種踏實有趣的感覺，好開心。