組合語言與系統程式第10週上機實習

資工2B 第16組 102502559吳承霖 102502557林唐正

程式執行原理

附上右圖為程式碼的片段，這次的上機實作主要要我們練習在Procedure內使用參數和區域變數，*INVOKE*這種呼叫Procedure的方式，有別於*CALL*，可以在呼叫的時候帶入參數，而不用在call procedure前把要使用的參數push進stack中。

而在程式碼中.stack段中，我們可以發現關於procedure的protype宣告，宣告Prototype可以告訴Assembler這個Procedure會在stack中用到哪些參數。另外，值得注意的是帶入參數若要傳入的是某個變數的記憶體位址，則須加上*ADDR*這個指令(不能用*OFFSET*)。

```
.stack 4096
ExitProcess proto dwExitCode:dword
FindLargest proto aPtr:PTR SDWORD, arraySize:DWORD
.data
Ex1Array sdword 103522039 , 102502559 , 102502557
Ex2Array sdword -103522039 ,-102502559 , -102502557
.code
main proc
    INVOKE FindLargest, ADDR Ex1Array, LENGTHOF Ex1Array
    INVOKE FindLargest, ADDR Ex2Array, LENGTHOF Ex2Array
    call WaitMsg
    invoke ExitProcess,0
main endp

FindLargest proc aPtr:PTR SDWORD, arraySize:DWORD
    push esi
    push ecx
    mov  eax,80000000h
```

程式運行結果截圖

```
EAX=062B9EF7  EBX=7EFDE000  ECX=00000000  EDX=00401000
ESI=0040400C  EDI=00000000  EBP=0018FF7C  ESP=0018FF74
EIP=00401044  EFL=00000206  CF=0  SF=0  ZF=0  OF=0  AF=0  PF=1


EAX=F9E3EF63  EBX=7EFDE000  ECX=00000000  EDX=00401000
ESI=00404018  EDI=00000000  EBP=0018FF7C  ESP=0018FF74
EIP=00401044  EFL=00000206  CF=0  SF=0  ZF=0  OF=0  AF=0  PF=1

Press any key to continue...
```

從先前的程式碼片段可以看到，在main中我們呼叫了FindLargest這個Procedure兩次，每次Procedure在執行的最後會把每個register的Dump出來，所以console中我們可以看到有兩組暫存器狀態值。第一次呼叫是對於Ex1Array當作參數傳入Procedure，然後比較陣列中最大的數並存在EAX暫存器中，所以我們可以看到console中第一組暫存器的EAX值在十進位為103522039，第二組暫存器的EAX值在十進位表示則是-102502557，符合題目要求。

心得

這個禮拜上課學到好多種方式關於參數和區域變數的用法，大多都是利用stack的特性幫我們做到這件事情，不過我想寫程式的人一定也需要對stack的運作和結構有一些瞭解，傳地參數時才能正確地存取到我們所要的值，一路過來我們所學到的組合語言已經越來越接近高階語言的用法了，也對於組語和高階語言的map關係越來越熟悉。