Assembly programming

Dec 7th, 2018

Dept. of Computer Science and Information Engineering,
Fu Jen Catholic University,
Hsin Chuang, 24205
周賜福

Practice these programs to familiarize with the Invoke procedure (ADVANCED) method.

ODD program Question is as follows.

Given the following description, write the program in x86 Assembly Language.

Given the following **data section** and the **main procedure** for the x86 assembly language, complete the rest of the **two procedures** using **invoke** to accomplish your task. After getting signature from the TA or the instructor, **you can write the two procedures that you wrote to complete the program on the back of the answer sheet before you return.**

COMMENT !
Create a procedure named **FindThrees** that returns 1 if an array has three consecutive values of 3 somewhere in the array. Otherwise, return 0. The procedure's input parameter list contains a pointer to the array and the array's size. Use the PROC directive with a parameter list when declaring the procedure. Preserve all registers (except EAX) that are modified by the procedure. Write a test program that calls FindThrees several times with different arrays.
Another procedure **SuccessFailure** to display if the result is a failure or success
!

; Use the following Data section for your program.

```
include Irvine32.inc
; The following two lines are meant for declaring two array procedures for INVOKE
FindThrees proto aPtr:PTR SDWORD, arraySize:DWORD
SuccessFailure proto aPtr: PTR BYTE, aptr1: PTR BYTE

.data
Ex5Array1 sdword  4, 6, 3, 3, 2, 5
Ex5Array2 sdword 3,3,3,9,5
Ex5Array3 sdword 1,2,3,3,4,3,3,3,9
Ex5Array4 sdword 1,2,4,-4,-5,9
Ex5Array5 sdword  3,4,5,6,3,3,3
str1 BYTE "Success",0dh,0ah,0
str2 BYTE "Failure",0dh,0ah,0
```

Use the following main procedure for your program

```
.code
main proc
    invoke FindThrees, ADDR Ex5Array1, LENGTHOF Ex5Array1
    invoke SuccessFailure, ADDR str1, ADDR str2
    comment !
    invoke FindThrees, ADDR Ex5Array2, LENGTHOF Ex5Array2
    invoke SuccessFailure, ADDR str1, ADDR str2
    invoke FindThrees, ADDR Ex5Array3, LENGTHOF Ex5Array3
    invoke SuccessFailure, ADDR str1, ADDR str2
    invoke FindThrees, ADDR Ex5Array4, LENGTHOF Ex5Array4
    invoke SuccessFailure, ADDR str1, ADDR str2
 !
    exit
 main endp
                                ; Write your procedure here to complete.
FindThrees proc,
                                                    ; complete the code for the procedure here.
FindThrees endp
SuccessFailure PROC,
                                                  ; complete the code for the procedure here.
Successfailure endp
end main
```

You are not given any run of the program!

Even program given in the following page

EVEN program Question is as follows.

Given the following description, write the program in x86 Assembly Language.

Given the following **data section** and the **main procedure** for the x86 assembly language, complete the rest of the two procedures using invoke to accomplish your task. After getting signature from the TA or the instructor, **you can write the two procedures that you wrote to complete the program on the back of the answer sheet before you return.**

; Quiz 2, EVEN Question: Counting Matching Elements

COMMENT !

Write a procedure named **CountMatches** that receives points to two arrays of signed doublewords, and a third parameter that indicates the length of the two arrays. For each element x_i in the first array, if the corresponding y_i in the second array is equal, increment a counter. At the end, return a count of the number of matching array elements in EAX. Write a test program that calls CountMatches and passes pointers to two different pairs of arrays. Use **the INVOKE** statement to call CountMatches and pass stack parameters and create a PROTO statement for CountMatches. Save and restore any registers (other than EAX) changed by your procedure.

Another procedure **Printthematches** to print out the number of matches found in the array.

!

; Use the following Data section for your program.

```
include Irvine32.inc     ; the following two lines are meant to declare the procedure
CountMatches proto, ptr1:PTR SDWORD,ptr2:PTR SDWORD, arraySize:DWORD
Printthematches proto, aptr:PTR BYTE, aptr1:PTR BYTE

.data
array1 sdword 10,5,4,-6,2
array2 sdword 10,5,3,-6,2      ; 4 matches
array3 sdword 4,1,2,8,9        ; 0 matches
array4 sdword 10,4,4,6,3       ; 2 matches
count = LENGTHOF array1
str1 BYTE "Number of matches are:  ",0
str2 BYTE "No matches found at all: ",0
last BYTE " Good Bye. I have identified correctly",0dh,0ah,0
```

Use the following Main procedure and complete the three procedures needed.

```
.code
main proc
        invoke CountMatches,ADDR array1, ADDR array2, count
        invoke Printthematches, ADDR str1, ADDR str2
Comment !
        invoke CountMatches, ADDR array1, ADDR array3, count
        invoke Printthematches, ADDR str1, ADDR str2
        invoke CountMatches, ADDR array1, ADDR array4, count
        invoke Printthematches, ADDR str1, ADDR str2
!
        mov edx, OFFSET last
        call writestring
        exit
main endp
CountMatches proc,
                                              ; Write the CountMatches procedure here.
CountMatches endp
Printthematches PROC,
                                       ; Write the procedure Printthematches here.
Printthematches endp
end main
```

You are not given any run of the program!

Good Luck to you to complete the Program!