5.5.6

3. RET 명령어를 프로시저에서 빠뜨린다면 무슨 일이 일어날까?

- 프로시저의 끝을 지나 계속 실행된다.

15. ArraySum 프로시저에서 16비트 워드 배열에 대해서 동작하려면 어느 문장이 수정될 필요가 있는가?

- add eax, [esi] -> add ax, [si]

add esi, 4 -> add si, 2

5.6.3

5. 16비트 워드 배열을 다룰 수 있게 하려면 합 계산 프로그램의 PromptForIntegers 프로시저에서 어떤 줄을 수정해야 하는가?

- Receives: ESI points to an array of doubleword intergers, ECX = arraysize

-> Receives: AX points to an array of doubleword intergers, BX = arraysize

5.8

7. 무직위 화면 위치

100msec의 시간 지연을 사용하여 100개의 임의의 화면 위치에 문자를 표시하는 프로그램을 작성하라. 힌트: 콘솔 윈도우의 현재 크기를 결정하기 위해서 GetMaxXY 프로시저를 사용한다.

-

include irvine32.inc

COUNT           equ     100     ; 문자 표시 횟수

DISPCH          equ     '1'     ; 출력 문자

DELAYTIME       equ     301     ; 무작위 지연 시간

.data

x       byte    ?

y       byte    ?       ; x, y 좌표

.code

main    proc

        call    Randomize

        call    Clrscr

        mov     ecx, COUNT

lp:

        mov     eax, 80         ; X 좌표 랜덤 위치 구함.

        call    RandomRange

        mov     x, al

        mov     eax, 25         ; Y 좌표 랜덤 위치 구함.

        call    RandomRange

        mov     y, al

        mov     dh, y

        mov     dl, x

        call    Gotoxy

        mov     eax, DELAYTIME

        call    RandomRange

        add     eax, 10

        call    Delay           ; 시간 지연

        mov     al, DISPCH

        call    WriteChar

        loop    lp

        exit

main    endp

end     main

9. 합 계산 프로그램

5.6.1절의 합 계산 프로그램을 다음과 같이 수정하라. 상수를 사용하여 배열 크기를 선택한다.

ARRAY\_SIZE = 20

Array DWORD ARRAY\_SIZE DUP (?)

사용자에게 처리할 정수 개수 입력 안내문을 출력하는 새 프로시저를 작성하라. 배열 크기를 PromptForIntergers 프로시저에게 전달한다. 예를 들면 다음과 같다.

How many integers will be added? 5

-

TITLE 5\_8\_9

INCLUDE Irvine32.inc  
ARRAY\_SIZE = 20  
.data  
str1 BYTE "Enter a signed integer: ",0  
str2 BYTE "The sum of the integers is: ",0  
str3 BYTE "How many integers will be added?: ",0  
array DWORD ARRAY\_SIZE DUP(?)

.code  
main PROC  
 call Clrscr  
 mov esi,OFFSET array  
 mov ecx,ARRAY\_SIZE  
 call ArraySize  
 call PromptForIntegers  
 call ArraySum  
 call DisplaySum  
 exit  
main ENDP  
ArraySize PROC   
  
    mov edx,OFFSET str3  
    call WriteString  
    call ReadInt  
    call Crlf  
    mov ecx, eax  
    mov [esi], ecx  
    ret  
ArraySize ENDP  
    mov edx,OFFSET str1  
L1: call WriteString  
    call ReadInt  
    call Crlf  
    mov [esi],eax  
    add esi, TYPE DWORD  
    loop L1  
    ret  
PromptForIntegers ENDP  
 mov eax,0  
L1:  add eax,[esi]  
 add esi,TYPE DWORD  
 loop L1  
 ret  
ArraySum ENDP  
 mov edx,OFFSET str2  
 call WriteString  
 call WriteInt  
 call Crlf  
 ret  
DisplaySum ENDP  
END main