< 과제 2 >

2017253041_홍성우

1.

(1)

```
C:₩asm>notepad arith.asm

C:₩asm>asm arith

C:₩asm>REM asm.bat - batch file for assemble & link  assembly programs
Microsoft (R) Macro Assembler Version 14.27.29112.0
Copyright (C) Microsoft Corporation.  All rights reserved.

 Assembling: arith.asm
Microsoft (R) Incremental Linker Version 14.27.29112.0
Copyright (C) Microsoft Corporation.  All rights reserved.

C:₩asm>arith
FFFFFFDC
```

```
C:₩asm>notepad flag.asm

C:₩asm>asm flag

C:₩asm>REM asm.bat - batch file for assemble & link  assembly programs
Microsoft (R) Macro Assembler Version 14.27.29112.0
Copyright (C) Microsoft Corporation.  All rights reserved.

 Assembling: flag.asm
Microsoft (R) Incremental Linker Version 14.27.29112.0
Copyright (C) Microsoft Corporation.  All rights reserved.

C:₩asm>flag

  EAX=0097FB00  EBX=00781000  ECX=005C1005  EDX=005C1005
  ESI=005C1005  EDI=005C1005  EBP=0097FB80  ESP=0097FB74
  EIP=005C1069  EFL=00000257  CF=1  SF=0  ZF=1  OF=0  AF=1  PF=1


  EAX=0097FBFF  EBX=00781000  ECX=005C1005  EDX=005C1005
  ESI=005C1005  EDI=005C1005  EBP=0097FB80  ESP=0097FB74
  EIP=005C1072  EFL=00000297  CF=1  SF=1  ZF=0  OF=0  AF=1  PF=1


  EAX=0097FB80  EBX=00781000  ECX=005C1005  EDX=005C1005
  ESI=005C1005  EDI=005C1005  EBP=0097FB80  ESP=0097FB74
  EIP=005C107B  EFL=00000A92  CF=0  SF=1  ZF=0  OF=1  AF=1  PF=0


  EAX=00000000  EBX=00781000  ECX=005C1005  EDX=005C1005
  ESI=005C1005  EDI=005C1005  EBP=0097FB80  ESP=0097FB74
  EIP=005C1082  EFL=00000246  CF=0  SF=0  ZF=1  OF=0  AF=0  PF=1


  EAX=FFFFFFFF  EBX=00781000  ECX=005C1005  EDX=005C1005
  ESI=005C1005  EDI=005C1005  EBP=0097FB80  ESP=0097FB74
  EIP=005C1088  EFL=00000296  CF=0  SF=1  ZF=0  OF=0  AF=1  PF=1
```

```
C:₩asm>notepad operator.asm

C:₩asm>asm operator

C:₩asm>REM asm.bat - batch file for assemble & link  assembly programs
Microsoft (R) Macro Assembler Version 14.27.29112.0
Copyright (C) Microsoft Corporation.  All rights reserved.

 Assembling: operator.asm
Microsoft (R) Incremental Linker Version 14.27.29112.0
Copyright (C) Microsoft Corporation.  All rights reserved.

C:₩asm>operator

  EAX=012FFD56  EBX=010FF078  ECX=00F21056  EDX=00F21005
  ESI=00F24000  EDI=00F24004  EBP=012FFD38  ESP=012FFD2C
  EIP=00F21080  EFL=00000246  CF=0  SF=0  ZF=1  OF=0  AF=0  PF=1


  EAX=00000002  EBX=00000008  ECX=00000004  EDX=00F21005
  ESI=00F24000  EDI=00F24004  EBP=012FFD38  ESP=012FFD2C
  EIP=00F21094  EFL=00000246  CF=0  SF=0  ZF=1  OF=0  AF=0  PF=1
```

```
C:₩asm>notepad indirect.asm

C:₩asm>asm indirect

C:₩asm>REM asm.bat - batch file for assemble & link  assembly programs
Microsoft (R) Macro Assembler Version 14.27.29112.0
Copyright (C) Microsoft Corporation.  All rights reserved.

 Assembling: indirect.asm
Microsoft (R) Incremental Linker Version 14.27.29112.0
Copyright (C) Microsoft Corporation.  All rights reserved.

C:₩asm>indirect

  EAX=00000010  EBX=00000020  ECX=00000030  EDX=00BD1005
  ESI=00BD4008  EDI=00BD1005  EBP=010FFDE4  ESP=010FFDD8
  EIP=00BD1076  EFL=00000202  CF=0  SF=0  ZF=0  OF=0  AF=0  PF=0


  EAX=00000030  EBX=00000030  ECX=00000030  EDX=00BD1005
  ESI=00000008  EDI=00000002  EBP=010FFDE4  ESP=010FFDD8
  EIP=00BD1092  EFL=00000202  CF=0  SF=0  ZF=0  OF=0  AF=0  PF=0


  EAX=00000010  EBX=00BD4000  ECX=00000030  EDX=00BD1005
  ESI=00000008  EDI=00000002  EBP=010FFDE4  ESP=010FFDD8
  EIP=00BD109F  EFL=00000202  CF=0  SF=0  ZF=0  OF=0  AF=0  PF=0
```

neg 명령어는 2의 보수를 취하는 것이고 add 와 sub 명령어는 각각 더하고 빼는 기능을 한다.

cf는 msb에서 carry가 일어났을 때 set되고 of는 (msb의 올림수) XOR (msb 하나 아래 비트의 올림수)가 일어났을 때 set된다.

al에서 byte ptr은 dword를 byte로 크기를 바꾸어 리틀엔디언에 맞게 34를 가져온것이고 bl과 cl은 각각 78, 56을 가져온 것이다. type는 크기를 말하고 sizeof는 type*lengthof를 뜻하고 lengthof는 개수를 말한다.

처음엔 barr의 주소를 esi에 넣고 esi의 값을 dword 단위인 4씩 늘려가며 나타낸 것이고, 다음엔 esi에 처음부터 8이란 값을 넣어서 barr의 세번째 원소를 나타낸 것이다. 마지막으로 ptrD에 barr의 주소를 넣어놨기 때문에 ebx에 ptrD의 주소가 아닌 barr의 주소가 들어가있으며 eax에는 barr의 첫번째 원소인 10이 나타난다.

(2)

```
include print.inc

.data
intarray WORD 100h,200h,300h,400h

.code
main proc
        mov edi, OFFSET intarray      ; address of intarray
        mov ecx, LENGTHOF intarray; loop counter(4)
        mov ax, 0                     ; zero the accumulator
L1:
        add ax, [edi]                           ; add an integer
        add edi,2                     ; point    to next    integer(+2)
        loop L1
        movzx eax,ax
        call DumpRegs

        mov eax, 0                              ; exit 0
        call ExitProcess
main endp
end main
```

```
C:₩asm>notepad loop1.asm

C:₩asm>asm loop1

C:₩asm>REM asm.bat - batch file for assemble & link  assembly programs
Microsoft (R) Macro Assembler Version 14.27.29112.0
Copyright (C) Microsoft Corporation.  All rights reserved.

 Assembling: loop1.asm
Microsoft (R) Incremental Linker Version 14.27.29112.0
Copyright (C) Microsoft Corporation.  All rights reserved.

C:₩asm>loop1

  EAX=00780A00  EBX=00483000  ECX=00000000  EDX=001B1005
  ESI=001B1005  EDI=001B4008  EBP=0078FC64  ESP=0078FC58
  EIP=001B107B  EFL=00000202  CF=0  SF=0  ZF=0  OF=0  AF=0  PF=0


C:₩asm>notepad loop1.asm

C:₩asm>asm loop1

C:₩asm>REM asm.bat - batch file for assemble & link  assembly programs
Microsoft (R) Macro Assembler Version 14.27.29112.0
Copyright (C) Microsoft Corporation.  All rights reserved.

 Assembling: loop1.asm
Microsoft (R) Incremental Linker Version 14.27.29112.0
Copyright (C) Microsoft Corporation.  All rights reserved.

C:₩asm>loop1

  EAX=00000A00  EBX=011DD000  ECX=00000000  EDX=00E51005
  ESI=00E51005  EDI=00E54008  EBP=012FFD98  ESP=012FFD8C
  EIP=00E5107E  EFL=00000202  CF=0  SF=0  ZF=0  OF=0  AF=0  PF=0
```

intarray의 원소들의 합을 구하기 위해 loop문을 사용했고 edi를 크기 2씩 늘려주며 더한 값을 ax에 저장하고 그것을 zero extension을 사용해 eax에 출력했다.

```
C:₩asm>notepad loop2.asm

C:₩asm>asm loop2

C:₩asm>REM asm.bat - batch file for assemble & link  assembly programs
Microsoft (R) Macro Assembler Version 14.27.29112.0
Copyright (C) Microsoft Corporation.  All rights reserved.

 Assembling: loop2.asm
Microsoft (R) Incremental Linker Version 14.27.29112.0
Copyright (C) Microsoft Corporation.  All rights reserved.

C:₩asm>loop2
This is the source string
```

source의 한 글자씩 esi값을 1씩 늘려가며 loop문을 사용했고 그것을 target으로 보내 출력했다.

2.

(1)

```
C:₩asm>notepad loop1.asm

C:₩asm>asm loop1

C:₩asm>REM asm.bat - batch file for assemble & link  assembly programs
Microsoft (R) Macro Assembler Version 14.27.29112.0
Copyright (C) Microsoft Corporation.  All rights reserved.

 Assembling: loop1.asm
Microsoft (R) Incremental Linker Version 14.27.29112.0
Copyright (C) Microsoft Corporation.  All rights reserved.

C:₩asm>loop1

  EAX=00197000  EBX=0029F000  ECX=00000000  EDX=001C1005
  ESI=001C1005  EDI=001C400C  EBP=0019F7E0  ESP=0019F7D4
  EIP=001C107B  EFL=00000206  CF=0  SF=0  ZF=0  OF=0  AF=0  PF=1
```

초기값을 바꿔 설정한 결과 ax값이 저렇게 나왔는데 이것은 계산이 틀렸다. 왜냐하면 초기값을 5000까지 더하면 F000h가 맞는데 8000까지 더하게 되면 carry가 일어나 17000h가 나와야하는데 이것은 ax의 범위로는 표현할 수 없기 때문이다.

(2)

-밑에 이어짐

```
include print.inc

.data
intarray DWORD 1000h,2000h,3000h,4000h,5000h,8000h

.code
main proc
        mov edi, OFFSET intarray      ; address of intarray
        mov ecx, LENGTHOF intarray ; loop counter(4)
        mov eax, 0                              ; zero the accumulator
L1:
        add eax, [edi]                          ; add an integer
        add edi,4                       ; point     to next    integer(+2)
        loop L1
        call DumpRegs

        mov eax, 0                              ; exit 0
        call ExitProcess
main endp
end main
```

```
C:₩asm>notepad loop1.asm

C:₩asm>asm loop1

C:₩asm>REM asm.bat - batch file for assemble & link  assembly programs
Microsoft (R) Macro Assembler Version 14.27.29112.0
Copyright (C) Microsoft Corporation.  All rights reserved.

 Assembling: loop1.asm
Microsoft (R) Incremental Linker Version 14.27.29112.0
Copyright (C) Microsoft Corporation.  All rights reserved.

C:₩asm>loop1

  EAX=00017000  EBX=008E5000  ECX=00000000  EDX=003B1005
  ESI=003B1005  EDI=003B4018  EBP=00AFF820  ESP=00AFF814
  EIP=003B107B  EFL=00000206  CF=0  SF=0  ZF=0  OF=0  AF=0  PF=1
```

32비트 덧셈을 위해 word를 dword로 변경하였고 ax를 eax로 변경하였고 edi 값에 더하는 값을 2가 아닌 4로 증가시켰다. 결과는 00017000으로 맞게 출력됐다.