



어셈블리 프로그래밍 (Assembly Programming)

16bit Assembly Programming

CSE3030 Dept. of CS&E
Sogang University

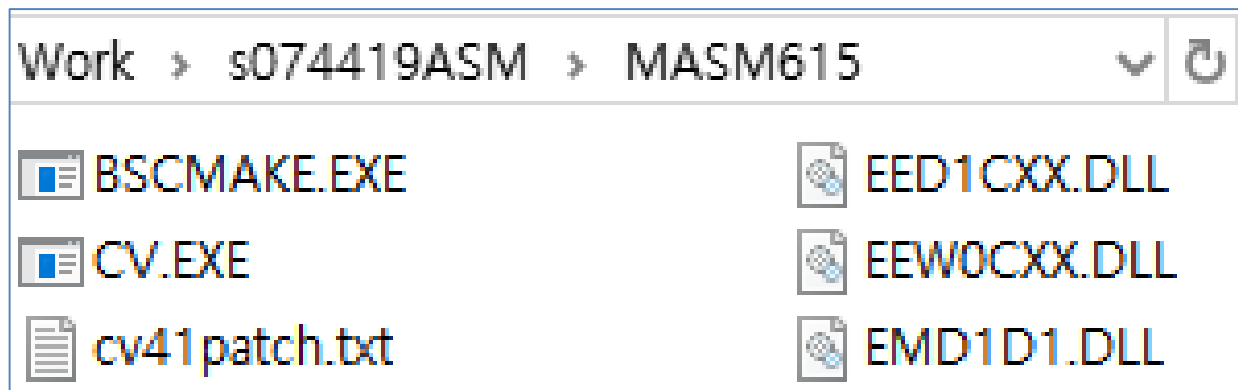
Making 16 Bit Applications Using VC2017

◆ 준비물

- ◆ VC2017, Irvine Library : 이 둘의 설치에 대해서는 이미 설명하였다.
- ◆ Codeview(DOS용 디버거), DOSBox (DOS emulator) : 이들은 새로 설치하여야 한다.

◆ Codeview Install

- ◆ cv41patch.exe.zip를 풀어 생성된 파일을 더블 클릭한 후 과정을 따른다 (클릭 unzip).
- ◆ C:\MASM615라는 폴더가 생길 것이다. 이 폴더를 이전에 작성한 \Work\s074419ASM 폴더로 이동한다.
- ◆ MASM615 폴더 안의 내용 일부





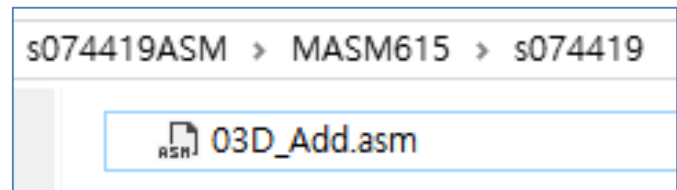
◆ 16 bit Assembly Program 편집

- ◆ MASM615 폴더에 다시 snnnnnnn 폴더를 만든다⁽¹⁾.
- ◆ 적절한 편집기로 프로그램을 작성한다(파일명: 03D_Add.asm).
- ◆ 작성한 파일을 폴더 s074419ASM\MASM615\s074419에 저장한다.
- ◆ 프로그램 내용 폴더 내용

```
TITLE Add and Subtract (16-bit.asm)
INCLUDE Irvine16.inc

.data
    var1      WORD 1000h
    sum       WORD 0

.code
main PROC
    mov  ax,@data
    mov  ds,ax
    mov  ax,var1      ; AX = 1000h
    add  ax,4000h      ; AX = 5000h
    sub  ax,2000h      ; AX = 3000h
    mov  sum, ax
    call DumpRegs
    exit
main ENDP
END main
```



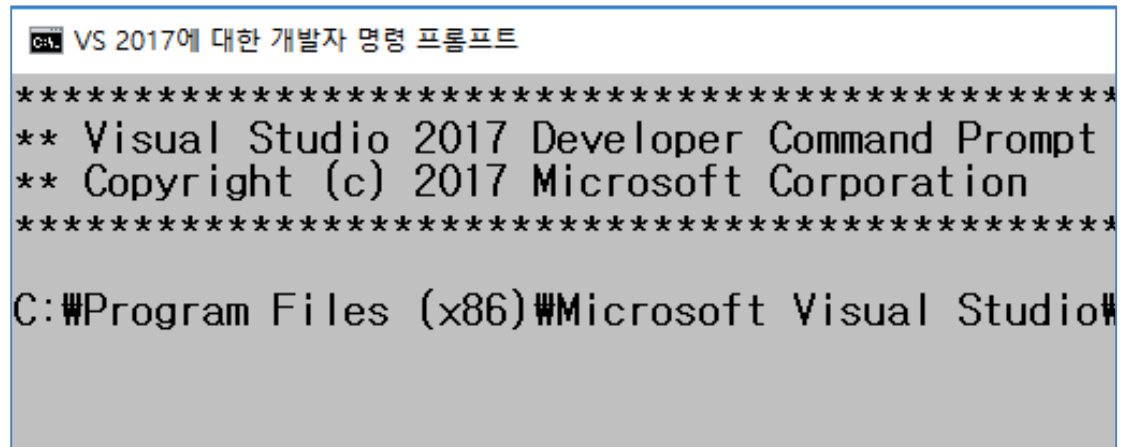
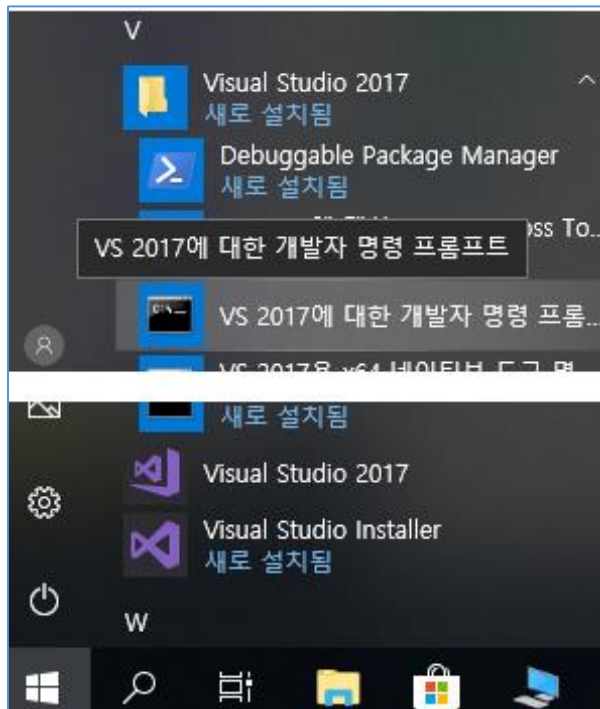
(1) nnnnnnn은 자신의 학번 뒤 6자리.

◆ 16 bit 어셈블리 프로그램 빌드

- ◆ 작성한 프로그램을 assemble하려면 **ML.EXE**라는 파일이 필요한데, 이는 다음에 보이는 폴더에 있다:

`\\Microsoft Visual Studio\\2017\\Community\\VC\\Tools\\MSVC\\14.16.27023\\bin\\Hostx86\\x86`

- ◆ **ML.EXE** 파일은 일반 명령 프롬프트에서는 수행이 되지 않는다.
- ◆ 시작 메뉴 → **Visual Studio 2017** → **Visual Studio Tools** → 클릭 **VS2017에 대한 개발자 명령 프롬프트**를 연다.



- ◆ D:\Work\snnnnnnnASM\Masm615\snnnnnnn으로 이동⁽¹⁾

```
C:\Program Files (x86)\Microsoft Visual Studio\2017\Community>D:
D:\>cd D:\Work\s074419ASM\MASM615\s074419
D:\Work\s074419ASM\MASM615\s074419>
```

- ◆ 다음 두 줄의 명령을 입력한다 :

```
SET INCLUDE=C:\Irvine
SET LIB=C:\Irvine
```

Include 파일 위치 지정
Library 위치 지정

- ◆ 다음 명령으로 03D_Add.asm 파일을 어셈블 → 03D_Add.obj를 만든다.

```
ML /nologo -c -omf -Fl -Zi 03D_Add.asm
```

- ◆ 다음 명령으로 03D_Add.exe 실행파일을 만든다 :

```
C:\Irvine\LINK16 03D_Add, , NUL, Irvine16 /CODEVIEW;
```

- ◆ Dir ↵을 입력하여 생성된 파일을 확인하자.

```
D:\Work\s074419ASM\MASM615\s074419>dir
D 드라이브의 볼륨: MYBACKUP
볼륨 일련 번호: B085-A8E3

D:\Work\s074419ASM\MASM615\s074419 디렉터리

2019-01-08 오후 03:44 <DIR> .
2019-01-08 오후 03:44 <DIR> ..
2010-03-03 오후 01:34 320 03D_Add.asm
2019-01-08 오후 04:06 2,710 03D_Add.obj
2019-01-08 오후 04:06 10,218 03D_Add.lst
2019-01-08 오후 04:08 6,912 03D_Add.exe
                4개 파일              20,160 바이트
                2개 디렉터리  44,673,204,224 바이트 남음
```

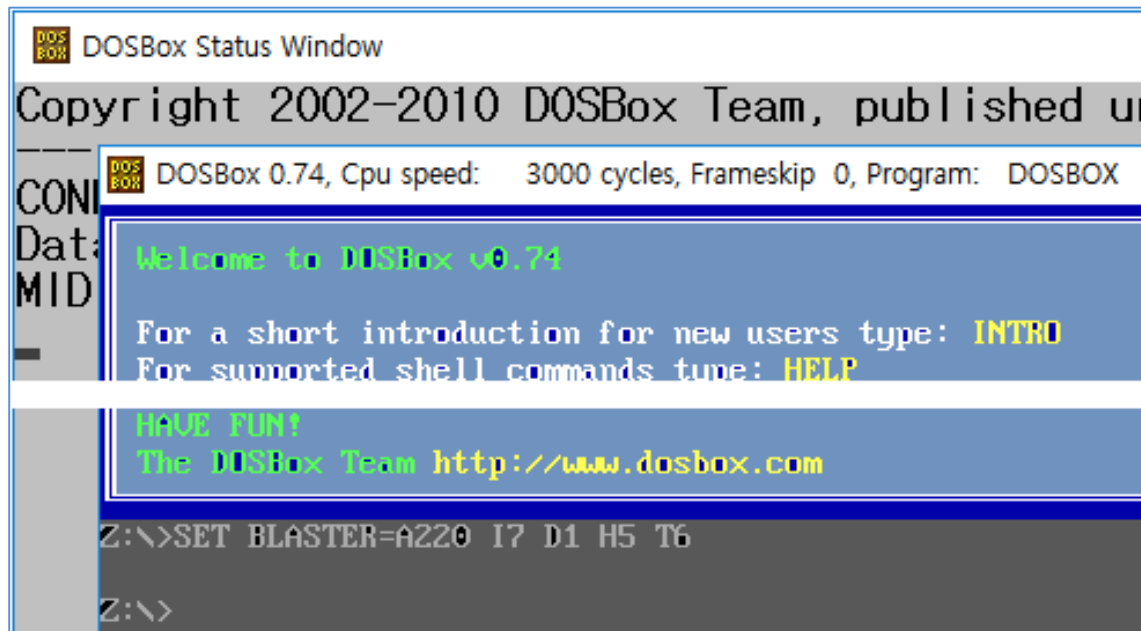
(1) nnnnnnn은 자신의 학번 뒤 6자리.



DOSBox를 통한 프로그램 실행

◆ DOSBox 설치 및 설정

- ◆ DOSBox0.74-win32-installer.exe.zip를 풀어 이를 설치한다(더블 클릭).
- ◆ 바탕화면에 단축 아이콘이 생길 것이다(A, 1).
- ◆ DOSBox를 더블 클릭하면 다음과 같은 화면이 보일 것이다.
- ◆ 참고(중요!) 만일 mouse를 DOSBox 밖으로 이동할 수 없을 경우 Ctrl+F10을 누른다.



(A) 단축
아이콘

(1) 만일 단축 아이콘이 안보이면 시작 메뉴에서 찾는다.



- ◆ 대상 폴더를 DOSBox 폴더로 다음과 같은 방법으로 마운트 하여야 한다:

```
mount D D:\Work\s074419ASM\MASM615 ↵  
D: ↵
```

```
Z:\>mount D D:\Work\s074419ASM\MASM615  
Drive D is mounted as local directory D:\Work\s074419ASM\MASM615\  
Z:\>D:  
D:\>
```

- ◆ 프로그램 실행은 실행 파일이 있는 폴더로 가서 실행한다.

```
cd s074419 ↵  
03D_Add ↵
```

```
D:\>cd s074419  
D:\S074419>03D_Add  
EAX=00000000 EBX=00000000 ECX=000000FF EDI=00000192  
ESI=00000000 EDI=00000400 EBP=0000091C ESP=00000400  
EIP=0000001A EFL=00007206 CF=0 SF=0 ZF=0 OF=0  
D:\S074419>
```

CodeView를 통한 Debugging

◆ CodeView

- ◆ CodeView는 Visual Studio의 디버깅 모드의 DOS 버전이다.
- ◆ 이를 사용하여 보면, 컴퓨터 기술이 얼마나 발전되었는지 알 수 있다.
- ◆ 다음과 같이 입력하여 CodeView를 실행한다 :

cd .. ↵

DOSBox 홈으로 돌아가기

CV s074419\03D_Add ↵

CodeView 실행

- ◆ 아래와 같은 창이 보인다.

```
DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: CV
File Edit Search Run Data Options Calls Windows Help
[11] locals
[31] source1 CS:IP 03D_Add.asm
5:      sum  WORD 0
6:      .code
7:      main PROC
8:      mov     ax,@data
9:      mov     ds,ax
10:
11:      mov     ax,var1      ; AX = 1000h
12:      add     ax,4000h      ; AX = 5000h
13:      sub     ax,2000h      ; AX = 3000h
14:      mov     sum, ax
15:      call    DumpRegs
[9] command
CV1053 Warning:  TOOLS.INI not found
CV1057 Warning:  CURRENT.STS not found; creating
>
<Trace> <Step> <Go> <After Return> <F3=S1 Fmt> INS DEC
```


- ◆ 상부 Windows항을 클릭하여 아래 그림과 같이 register와 watch 창을 이쁘게 배열해 보자(원시적이기는 하지만 VS2017와 거의 동일하다).

DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: CV

File Edit Search Run Data Options Calls Windows Help

[1] locals

[2] watch

var1 = 4096
sum = 0

[3] source1 CS:IP 03D_Add.asm

```
5:      sum  WORD 0
6:      .code
7:      main PROC
8:      mov  ax,@data
9:      mov  ds,ax
10:
11:      mov  ax,var1      ; AX = 1000h
12:      add  ax,4000h      ; AX = 5000h
13:      sub  ax,2000h      ; AX = 3000h
14:      mov  sum, ax
15:      call DumpRegs
```

[7] reg

AX = 0000
BX = 0000
CX = 0000
DX = 0000
SP = 0400
BP = 0000
SI = 0000
DI = 0000
DS = 0507
ES = 0507
SS = 067C
CS = 05D7
IP = 0000
FL = 0200
NU UP EI PL
NZ NA PO NC

[9] command

CV1053 Warning: TOOLS.INI not found
>

<Trace> <Step> <Go> <After Return> <F3=S1 Fmt> INS DEC

◆ CodeView 실행(1)

- ◆ **F5** : go, **F10** : step, **F7** : go to cursor, **F8** : trace, **F9** : break 등의 단축키를 사용할 수 있다.
- ◆ F10을 사용하여 한 스텝씩 진행하며, 레지스터 값과 Watch 창 의 변수 값을 관찰하자.
- ◆ 아래는 sum 값이 저장된 후 캡처한 화면을 보인 것이다.

DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: CV

File Edit Search Run Data Options Calls Windows Help

[1] locals [2] watch [7] reg

var1 = 4096
sum = 12288

[3] source1 CS:IP 03D_Add.asm

```
12: add ax,4000h ; AX = 5000h
13: sub ax,2000h ; AX = 3000h
14: mov sum, ax
15: call DumpRegs
16:
17: exit
18: main ENDP
19: END main
20:
```

[7] reg

AX = 3000
BX = 0000
CX = 0000
DX = 0000
SP = 0400
BP = 0000
SI = 0000
DI = 0000
DS = 0659
ES = 0507
SS = 0670
CS = 05D7
IP = 3011
FL = 0206
NU UP EI PL
NZ NA PE NC

[9] command

CV1053 Warning: TOOLS.INI not found
>

<Trace> <Step> <Go> <After Return> <F3-S1 Fmt> INS DEC

CodeView 종료는
quit ↵
DOSBox 종료는
exit ↵

(1) (참고) Mouse를 DOSBox 밖으로 옮길 수 없을 때는 Ctrl-F10을 누른다.