

리버스 엔지니어링(1)

발표자: 양유진

링크: https://youtu.be/jOm_IZCAQE

리버스 엔지니어링 (역공학, Reverse Engineering) 이란?

“장치, 시스템의 기술적인 원리를 구조분석을 통해 발견하는 과정”

OllyDbg(올리 디버거)란?



“바이너리 코드 분석을 위한 x86 디버거”

- 소스코드가 없을 때 사용됨.
- 동적분석 진행할 수 있게 도움.

실습 환경 조성 (1) 다운로드 및 설치

1. OllyDbg 다운로드

<http://www.ollydbg.de/> >> Download >> (final version)다운로드

2. Virtualbox 다운로드 및 설치

<https://www.virtualbox.org/wiki/Downloads>

3. Windows7 Virtual Machine 다운로드 >>IE8 on Win7(x86) >>VirtualBox

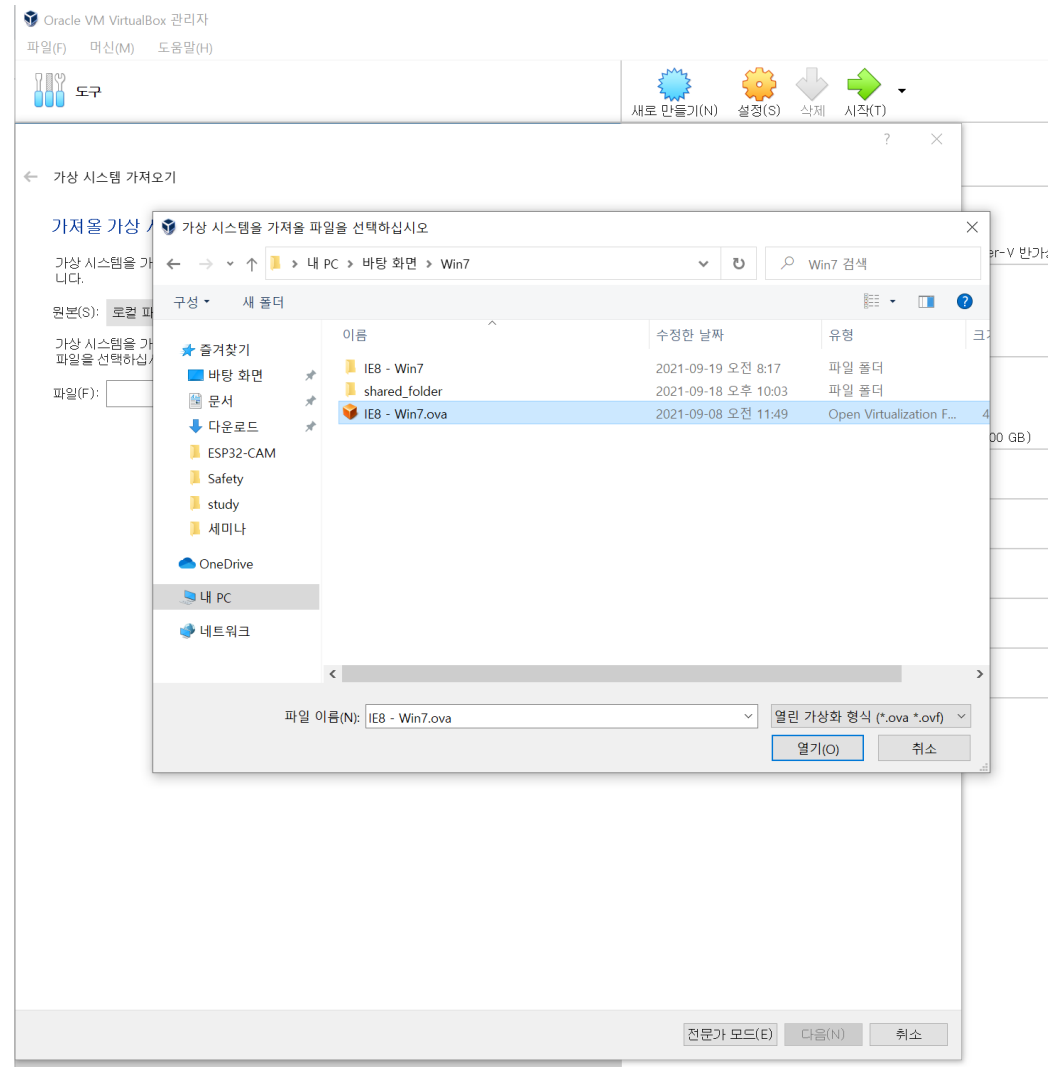
<https://developer.microsoft.com/en-us/microsoft-edge/tools/vms/>

4. Dev C++ 다운로드

<https://sourceforge.net/projects/orwelldevcpp/>

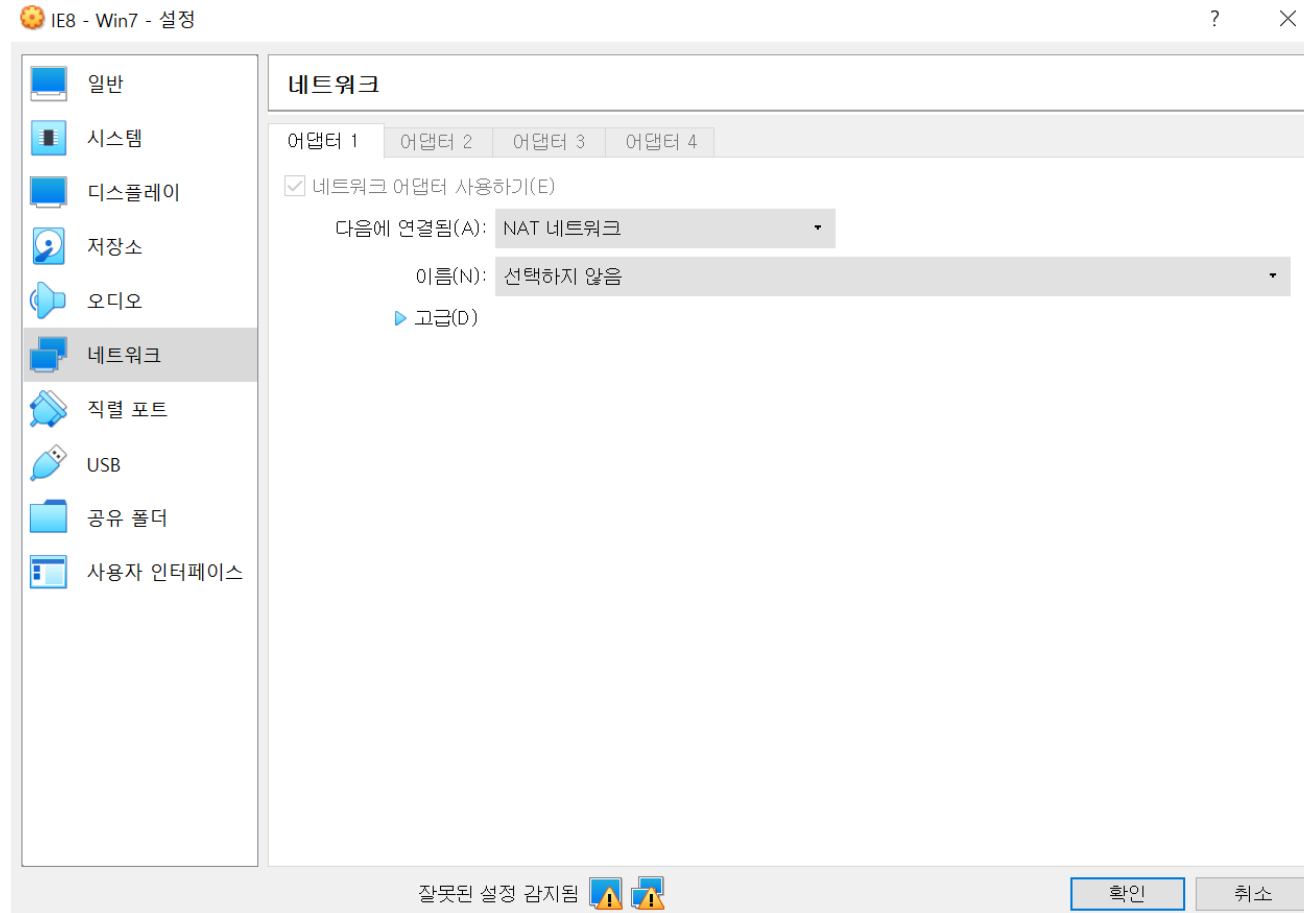
실습 환경 조성 (2) VM 설정 - VM 갖고오기

파일 >> 가상 시스템 가져오기 >> VM 선택 >> 다음 >> 가져오기



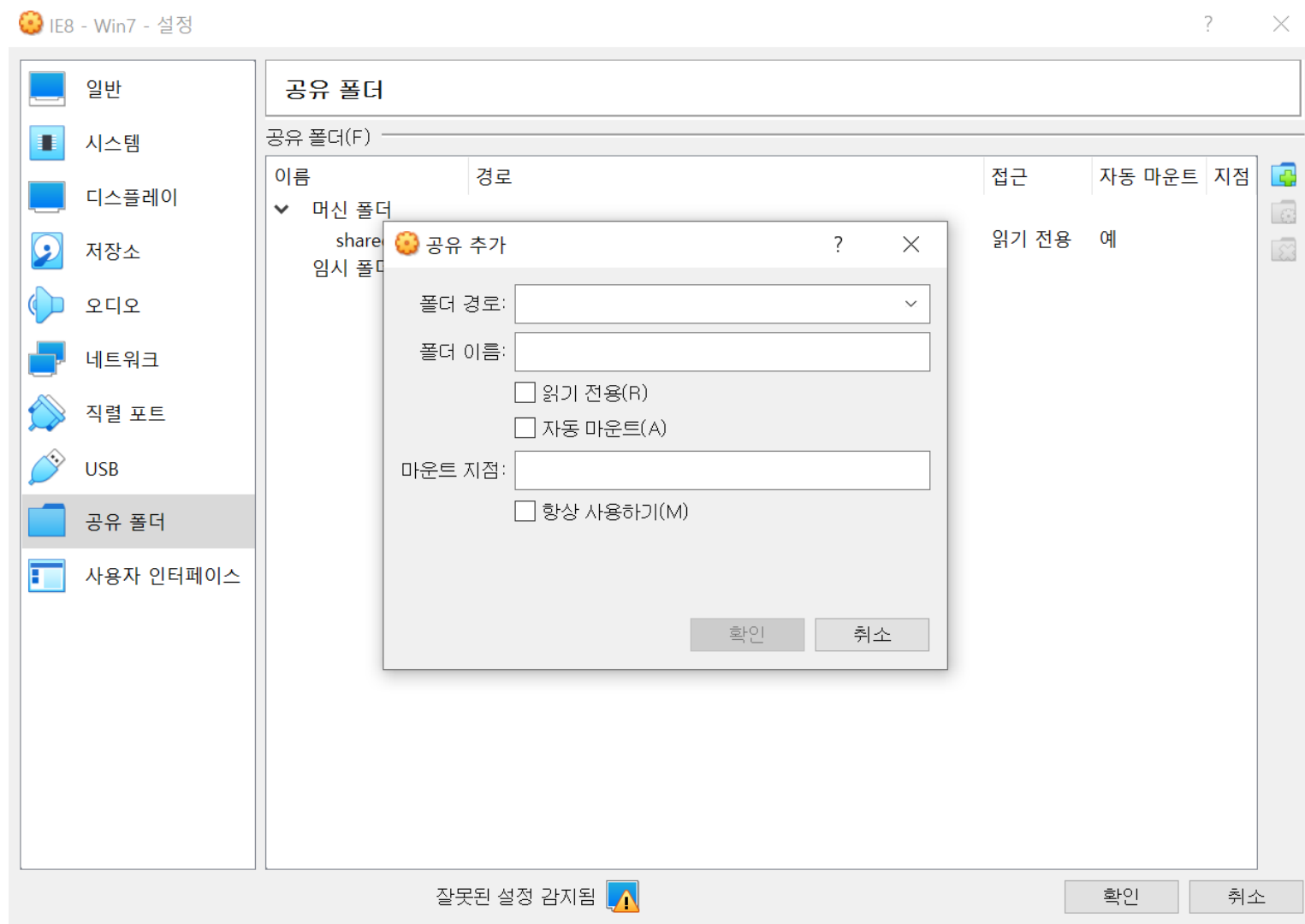
실습 환경 조성 (2) VM 설정 - 네트워크 설정

설정 >> 네트워크 >> NAT 네트워크 >> 확인

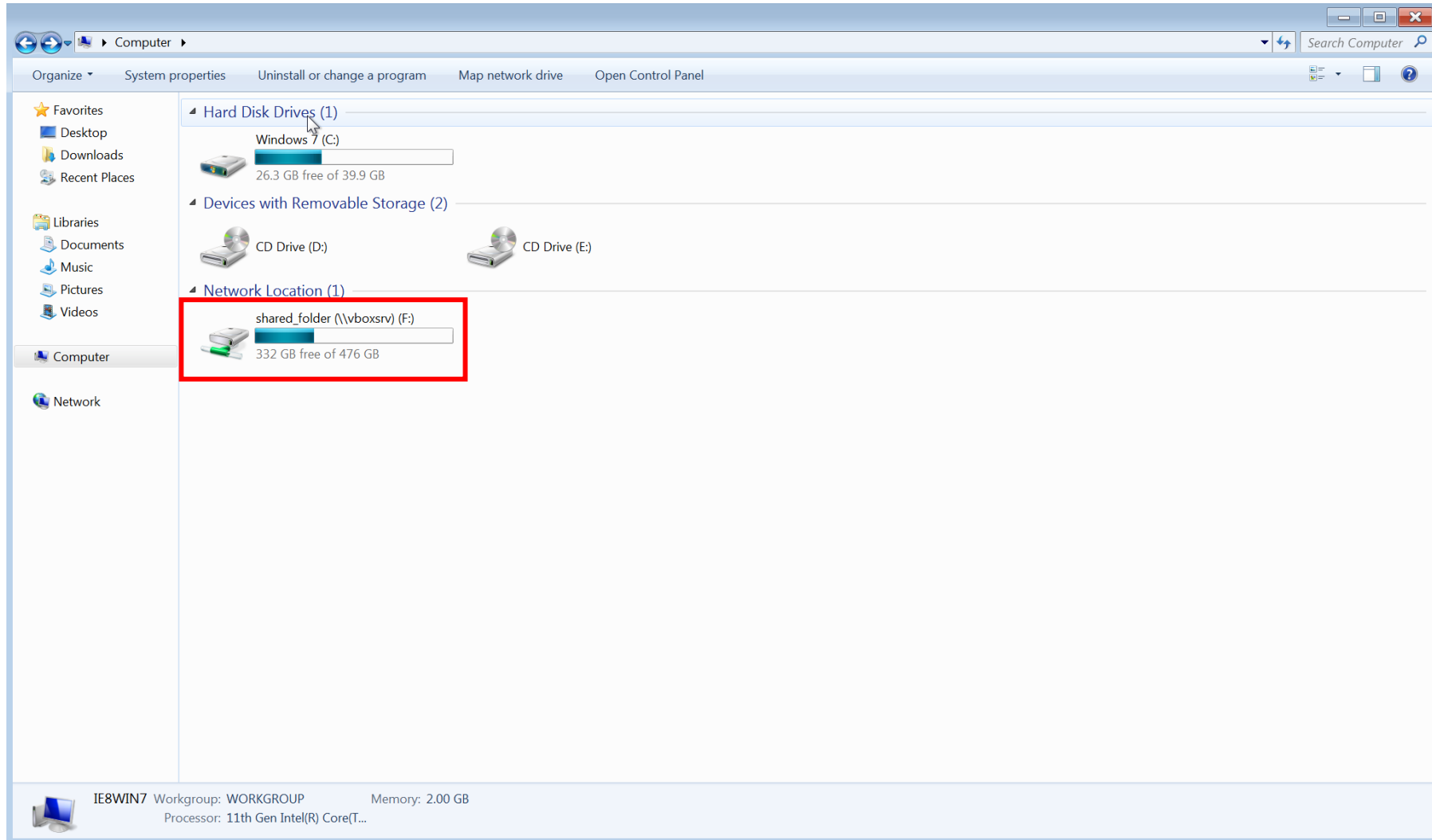


실습 환경 조성 (2) VM 설정 - 공유폴더 설정

설정 >> 공유폴더 >> (경로, 이름, 읽기전용, 자동마운트) 지정 >> 확인 >> 다운로드 받은 파일 해당 폴더에 붙여넣기

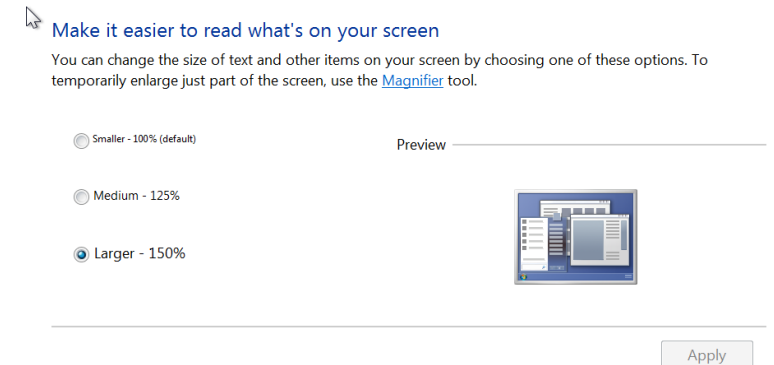
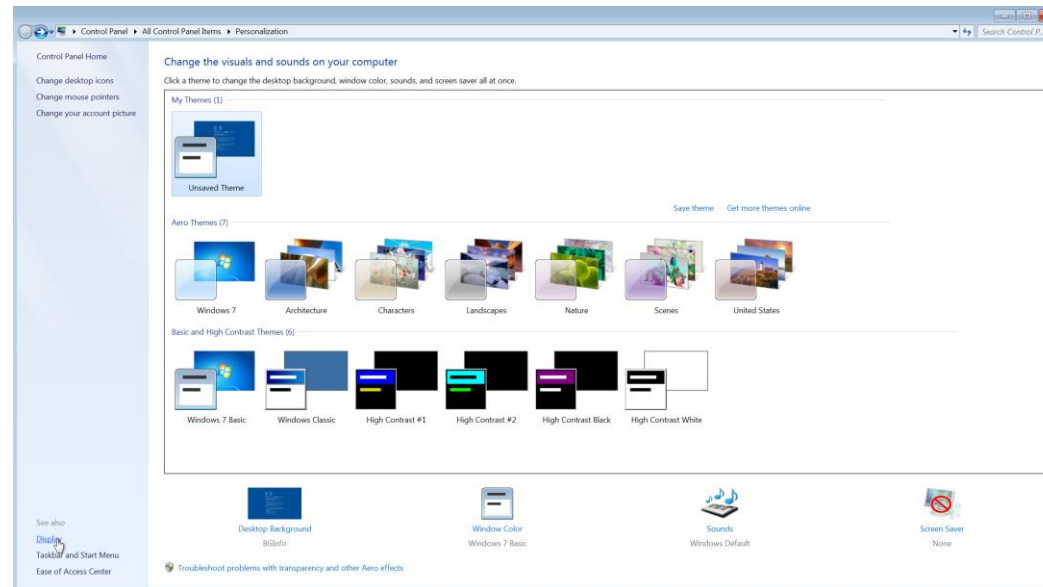
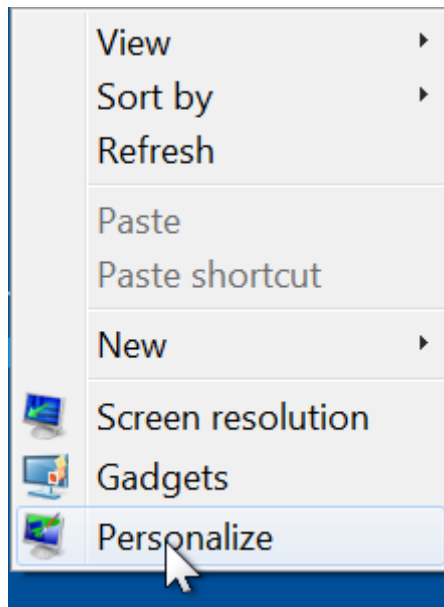


실습 환경 조성 (2) VM 설정 - 공유폴더 설정



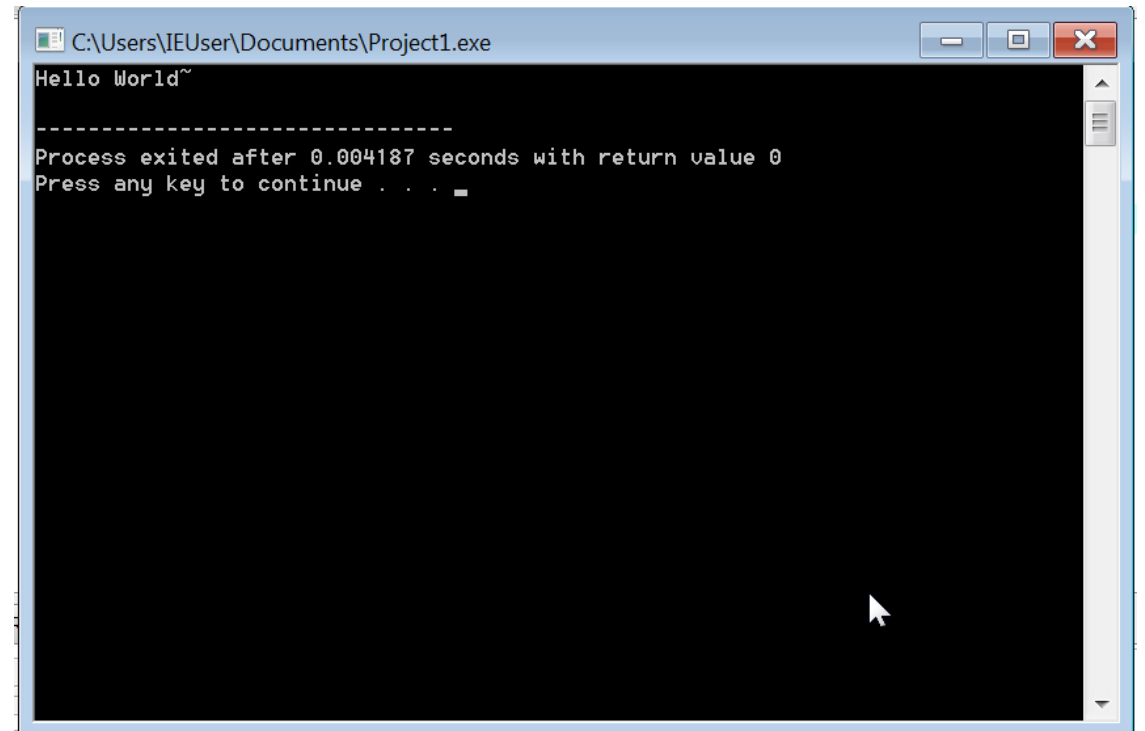
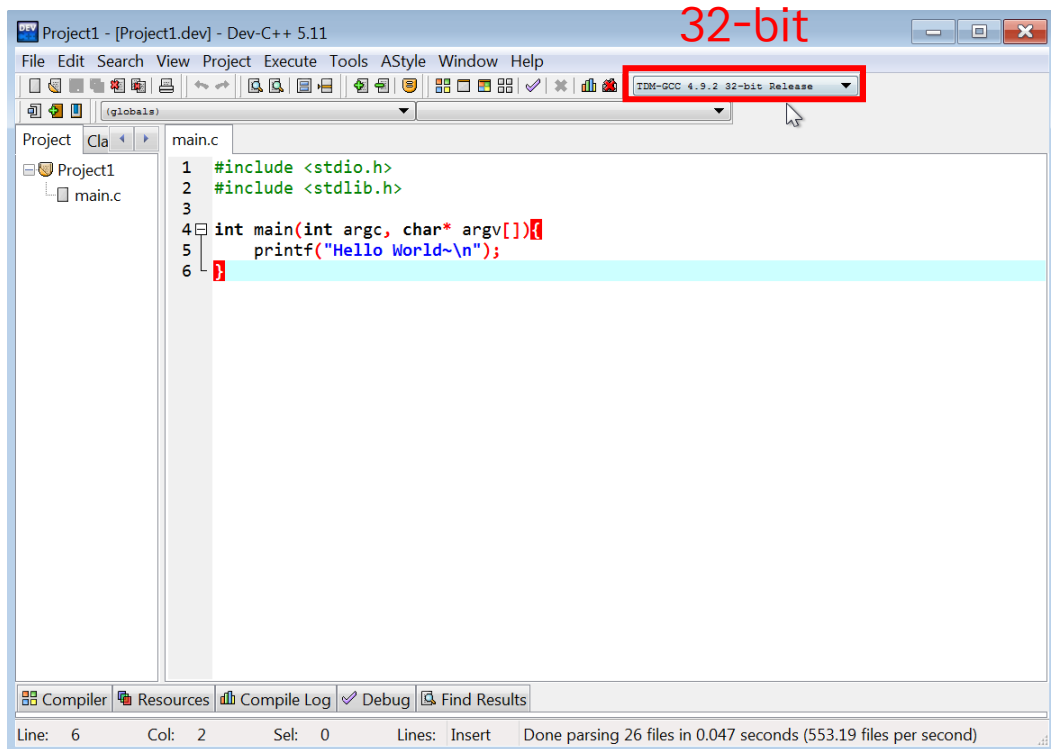
실습 환경 조성 (2) VM 설정 - 글자 크기 조정 (옵션)

Personalize >> Display >> (원하는 비율 고르기) >> Apply



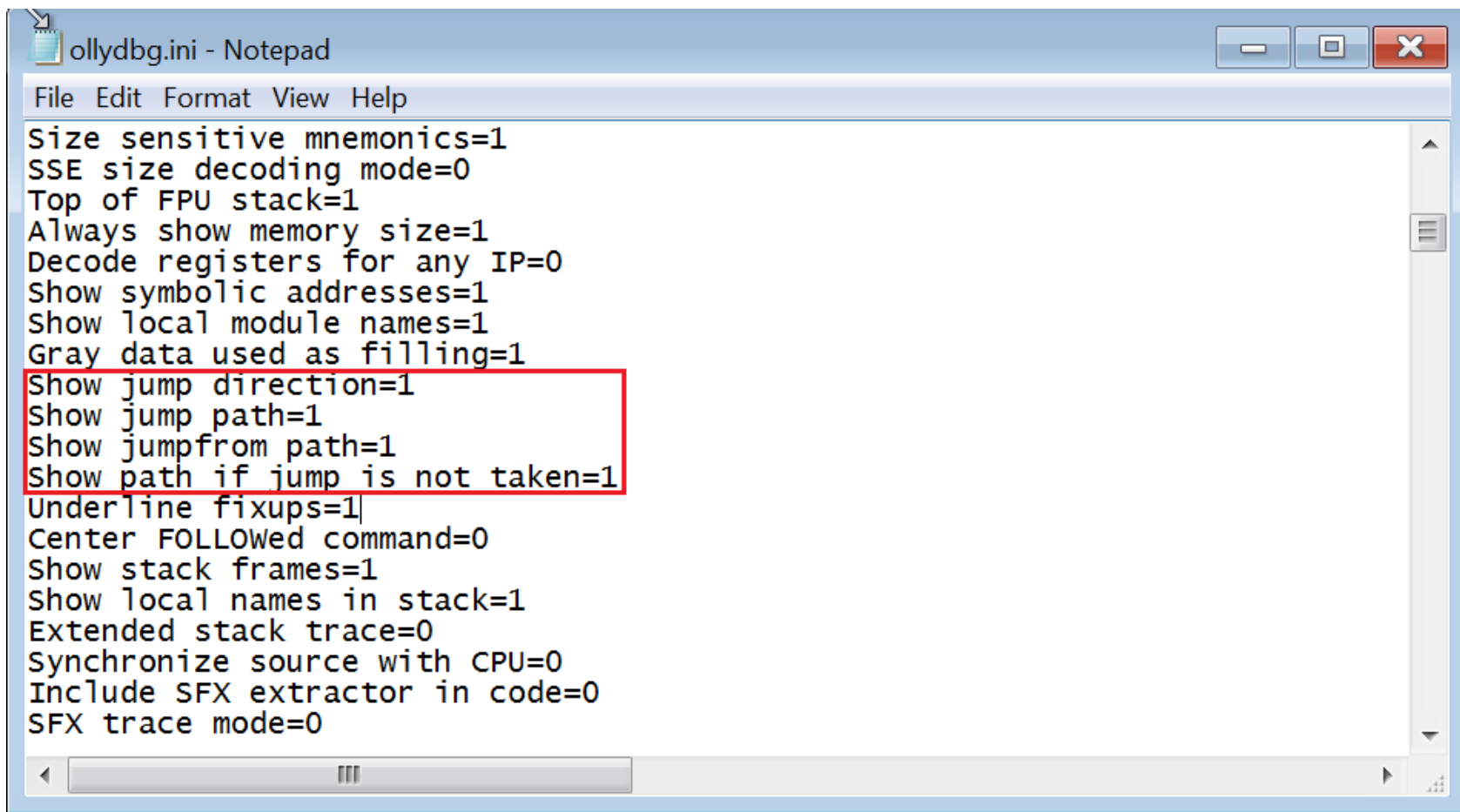
(Username) IEUser
(Password) Passw0rd!

c언어 작성 후, Compile&Run



OllyDbg 설정

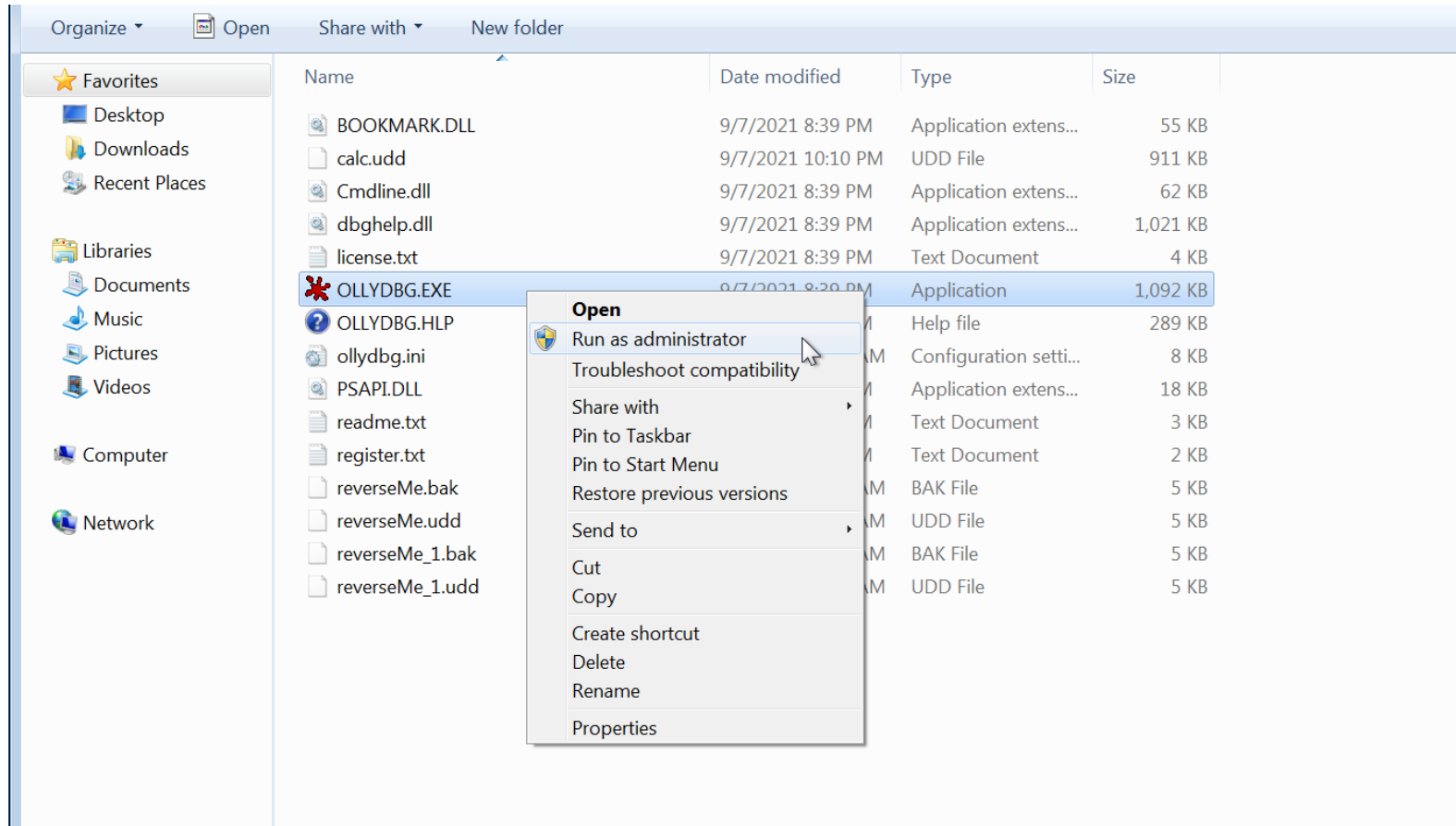
ollydbg.ini 폴더에서 jump라는 keyword가 들어간 부분을 0에서 1로 바꿉니다.



```
ollydbg.ini - Notepad
File Edit Format View Help
Size sensitive mnemonics=1
SSE size decoding mode=0
Top of FPU stack=1
Always show memory size=1
Decode registers for any IP=0
Show symbolic addresses=1
Show local module names=1
Gray data used as filling=1
Show jump direction=1
Show jump path=1
Show jumpfrom path=1
Show path if jump is not taken=1
Underline fixups=1
Center FOLLOWed command=0
Show stack frames=1
Show local names in stack=1
Extended stack trace=0
Synchronize source with CPU=0
Include SFX extractor in code=0
SFX trace mode=0
```

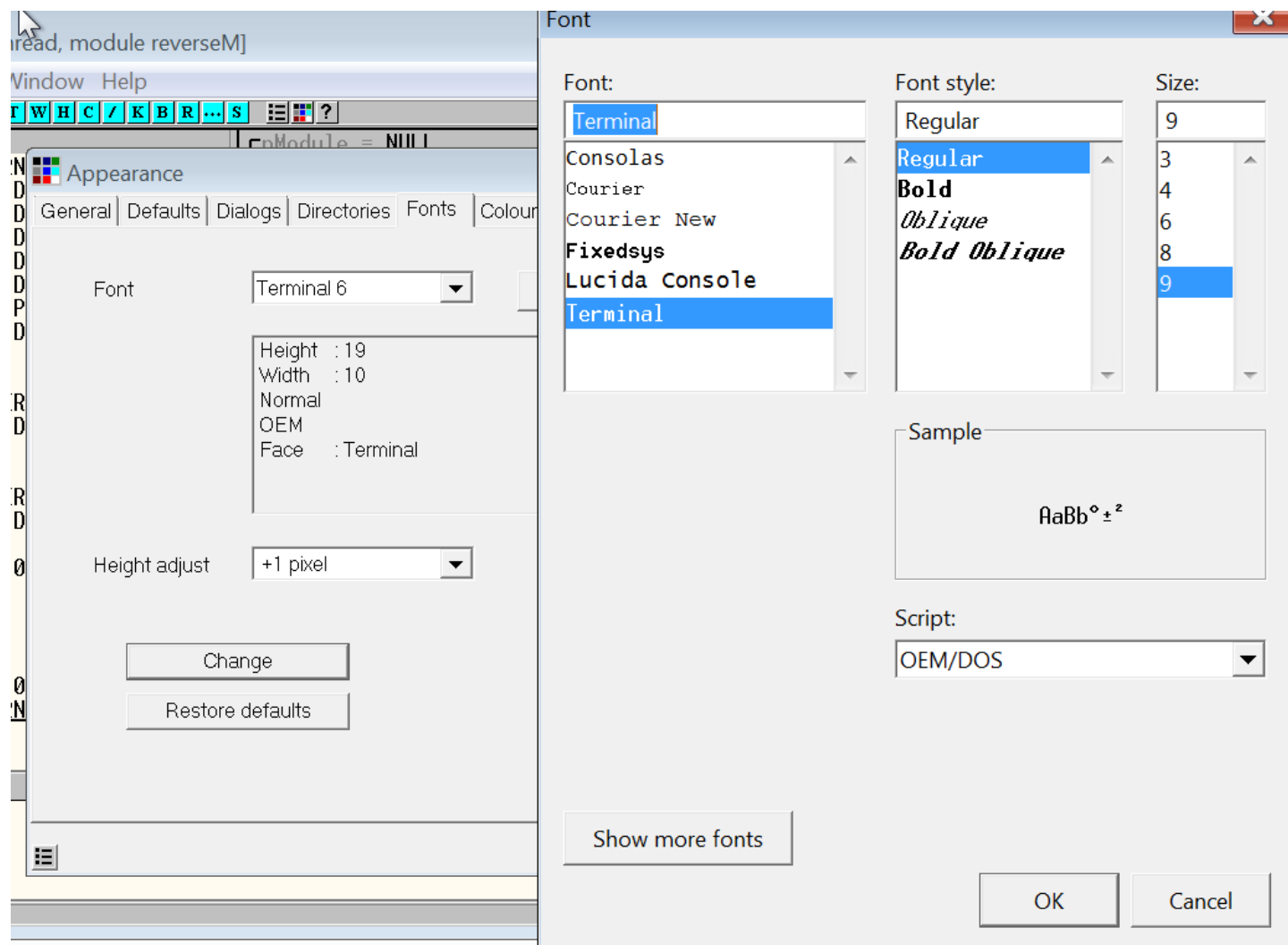
OllyDbg 실행

반드시 관리자 모드로 실행해주세요



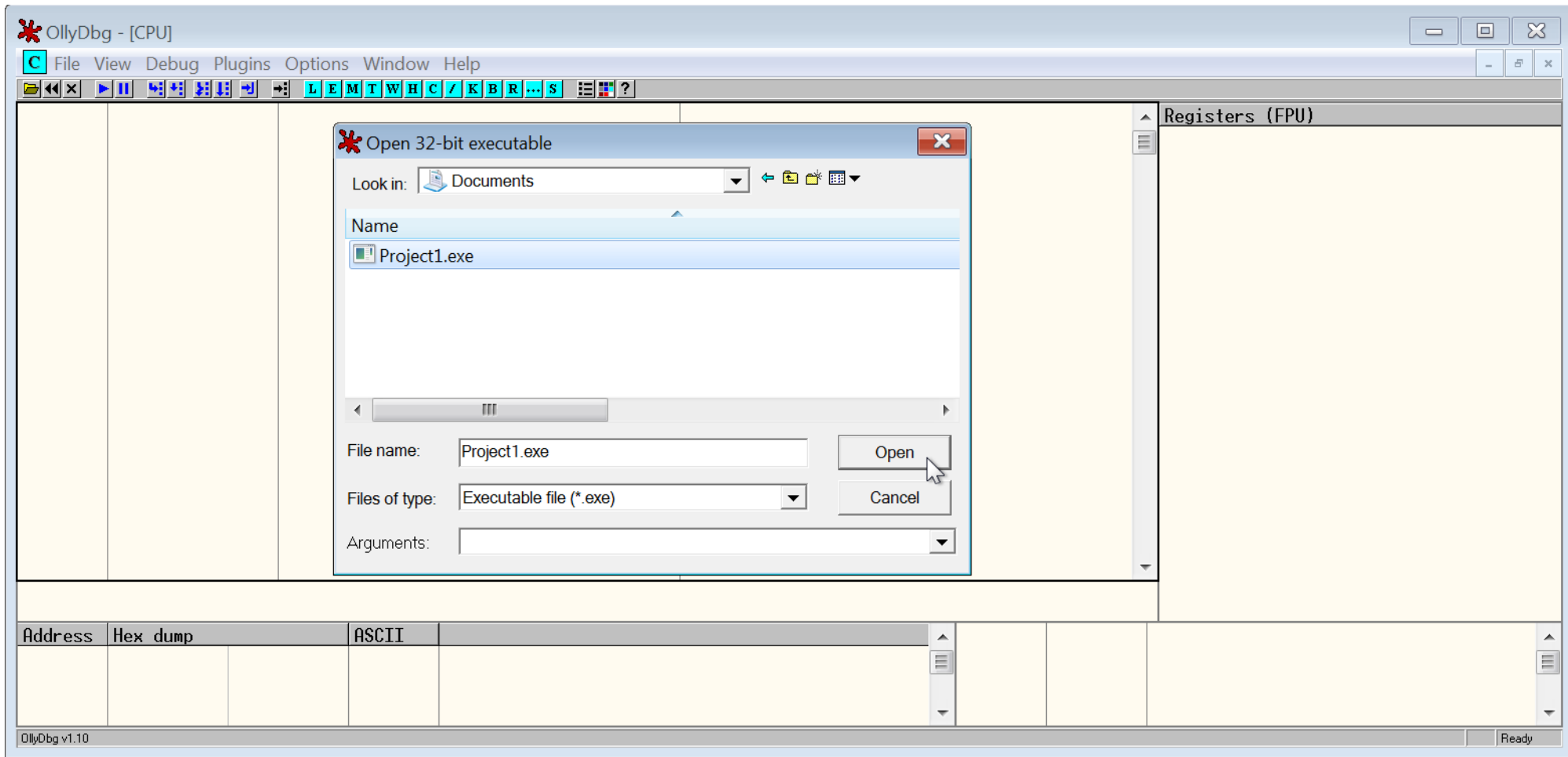
OllyDbg 글씨 설정 (옵션)

Options >> Appearance >> Fonts >> Change



실행파일 열기

File >> Open >> 프로젝트의 실행파일 선택



OllyDbg 기본 화면 설명

The screenshot displays the OllyDbg interface with four windows highlighted by red boxes and numbered 1 through 4.

1) Code Window: Shows the assembly code for the main thread of Project1.exe. The code starts at address 00401000 and includes instructions like RETN, LEA, SUB, XOR, CMP, MOV, JE, CALL, and JMP. The code is disassembled into a list of instructions with their corresponding addresses and operands.

2) Register Window: Displays the current state of the CPU registers. The registers are listed on the left, and their values are shown on the right. The registers include EAX, ECX, EDX, EBX, ESP, EBP, ESI, EDI, EIP, C, P, A, Z, S, T, D, O, EFL, ST0, ST1, ST2, ST3, ST4, ST5, ST6, ST7, FST, and FCW. The values are shown in hexadecimal and decimal formats.

3) Dump Window: Shows the memory dump of the program. The dump is organized into columns for Address, Hex dump, and ASCII. The address range is from 00403000 to 004030A0. The hex dump shows the raw memory data, and the ASCII column shows the corresponding text data.

4) Stack Window: Displays the current stack frame. The stack is shown as a list of memory addresses and their corresponding values. The stack grows downwards, and the values are shown in hexadecimal. The stack window also shows the return address and the return value of the current function.

OllyDbg 기본 화면 설명

1) Code Window

- Disassembly code를 표시 → comment, label 보여줌
- 코드 분석 → 정보(loop, jump 위치 등) 표시
- 기계어 → 어셈블리어

3) Dump Window

- 프로세스에서 원하는 memory 주소 위치를 Hex(16진법)와 ASCII/unicode 값으로 보여줌.
- 수정, 저장도 가능함.

2) Register Window

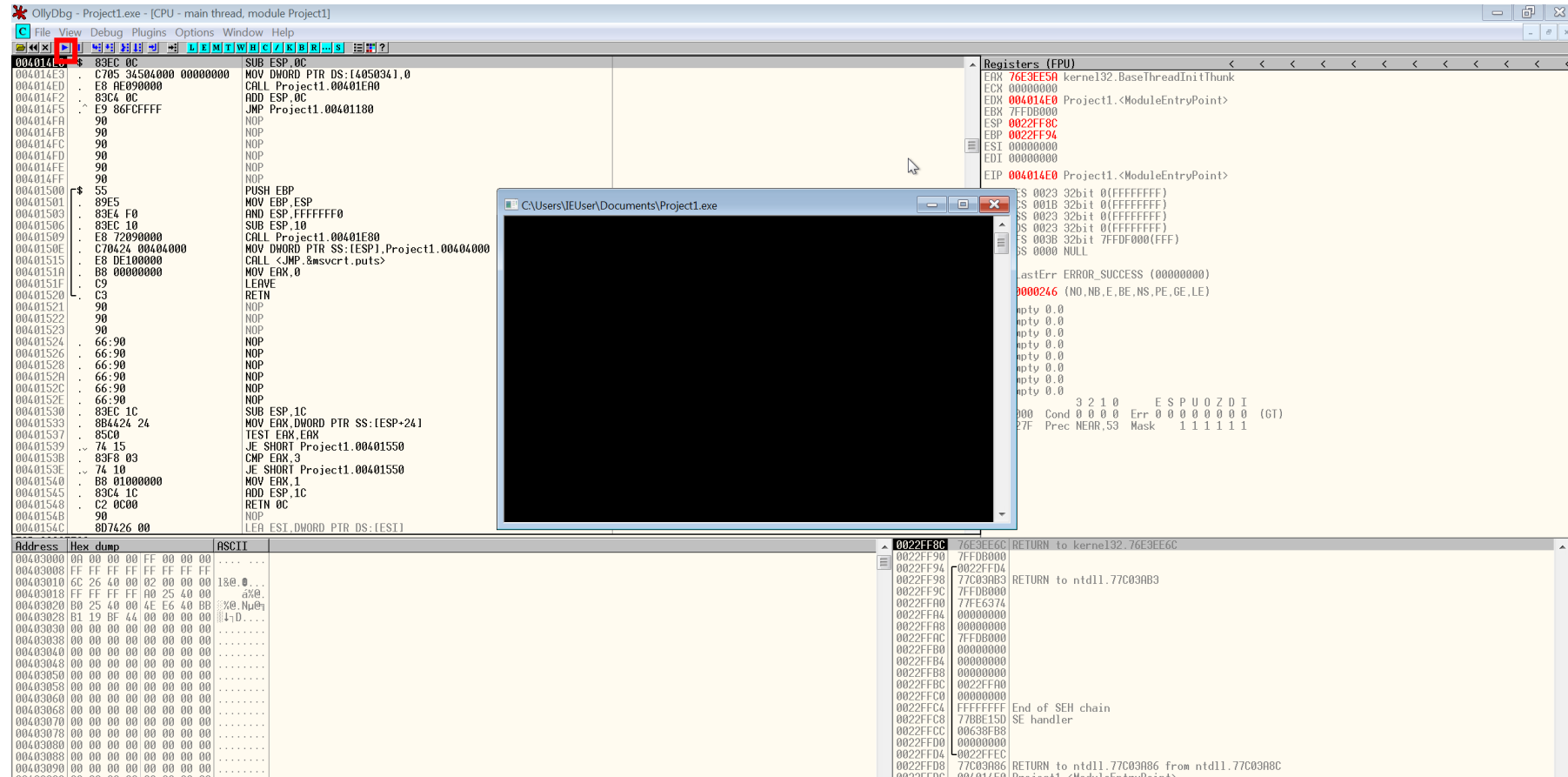
- CPU Register 값 실시간으로 표시
- 특정 Register 값 수정도 가능함

4) Stack Window

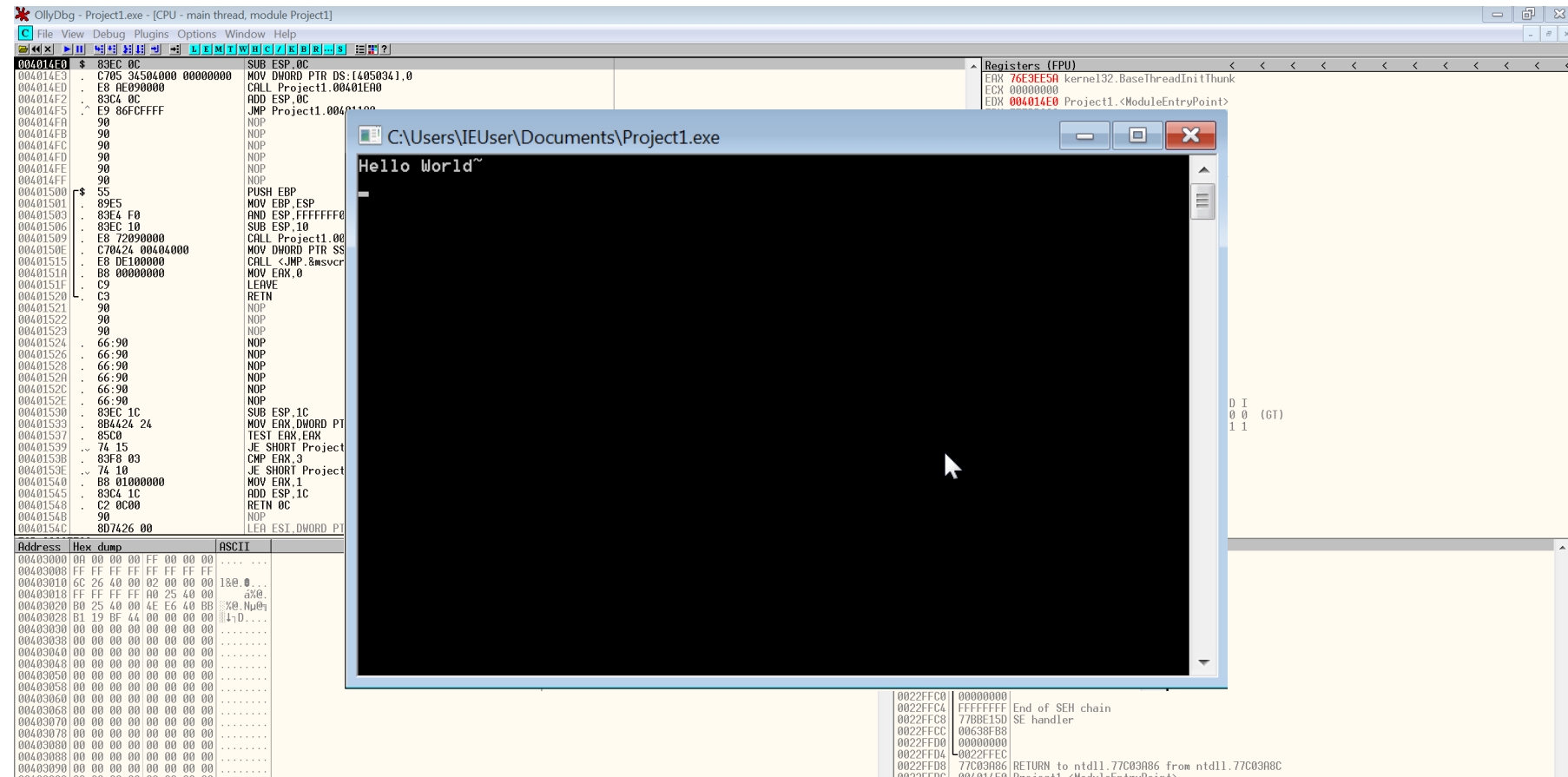
- ESP Register가 가리키는 프로세스
- stack memory를 실시간으로 표시하고 수정도 가능함.

실행파일 열기

실행버튼 / F9



실행파일 열기



OllyDbg 자주 쓰이는 기본 명령어&단축기

Ctrl + F2 : Restart (처음부터 디버깅 재시작)

F7 : Step into (하나의 OP code 실행 - CALL명령 만나면 함수 코드 내부로 감)

F8 : Step over (하나의 OP code 실행 - CALL명령 만나면 함수 실행)

Ctrl + G : Go to (원하는 주소 입력하면 찾아갈 수 있음 - 실행X)

F9 : Run (실행)

F2 : Break Point/SET (BP 설정/해제)

Assembly 기초 명령어

CALL [주소]: [주소]에 위치한 함수 호출

JMP [주소]: [주소]로 점프(이동)

PUSH [주소]: 스택에 [주소] 저장

RETN: 스택에 저장된 복귀 주소로 점프

MOV [DEST], [SRC]: SRC에 있는 값을 DEST로 복사함.

- MOV EBP, ESP → ESP에 있는 값을 EBP에 복사.

INC [A]: A에 +1. (레지스터, 메모리에만 사용 가능)

DEC [A]: A에 -1. (레지스터, 메모리에만 사용 가능)

ADD [A], [B]: $A += B$

SUB [A], [B]: $A -= B$

Project1.exe

Stub Code

: 컴파일러가 프로그램 만들 때 집어넣는 코드

- 프로그램 실행에 필요한 정보를 얻어오는 코드로 구성되어 있음.

OllyDbg - Project1.exe - [CPU - main thread, module Project1]

File View Debug Plugins Options Window Help

004014E0 83EC 0C SUB ESP,0C

004014E3 C705 34504000 MOV DWORD PTR DS:[405034],0

004014E6 E8 9E090000 CALL Project1.00401E90

004014F2 83C4 0C ADD ESP,0C

004014F5 E9 86FCFFFF JMP Project1.00401180

004014FA 90 NOP

004014FB 90 NOP

004014FC 90 NOP

004014FD 90 NOP

004014FE 90 NOP

004014FF 90 NOP

00401500 55 PUSH EBP

00401501 89E5 MOV EBP,ESP

00401503 83E4 F0 AND ESP,FFFFFFF0

00401506 83EC 10 SUB ESP,10

00401509 E8 62090000 CALL Project1.00401E70

0040150E C70424 00404000 MOV DWORD PTR SS:[ESP],Project1.00404000 ASCII "Hello World"

00401515 E8 CE100000 CALL <JMP.&msvcrt.puts> puts

0040151A C9 LEAVE

0040151B C3 RETN

0040151C 66:90 NOP

0040151E 66:90 NOP

00401520 83EC 1C SUB ESP,1C

00401523 8B4424 24 MOV EAX,DWORD PTR SS:[ESP+24]

00401527 85C0 TEST EAX,EAX

00401529 74 15 JE SHORT Project1.00401540

0040152B 83F8 03 CMP EAX,3

0040152E 74 10 JE SHORT Project1.00401540

00401530 B8 01000000 MOV EAX,1

00401535 83C4 1C ADD ESP,1C

00401538 C2 0C00 RETN 0C

0040153B 90 NOP

0040153C 8D7426 00 LEA ESI,DWORD PTR DS:[ESI]

00401540 8B5424 28 MOV EDI,DWORD PTR SS:[ESP+28]

00401544 894424 04 MOV DWORD PTR SS:[ESP+4],EAX

00401548 8B4424 20 MOV EAX,DWORD PTR SS:[ESP+20]

0040154C 895424 08 MOV DWORD PTR SS:[ESP+8],EDI

00401550 890424 MOV DWORD PTR SS:[ESP],EAX

00401553 E8 080C0000 CALL Project1.00402160

00401558 B8 01000000 MOV EAX,1

0040155D 83C4 1C ADD ESP,1C

00401560 C2 0C00 RETN 0C

ESP=0022FF8C

Registers (FPU)

EAX 758EEE6C kernel32.BaseThreadInitThunk

ECX 00000000

EDX 004014E0 Project1.<ModuleEntryPoint>

EBX 7FFDB000

ESP 0022FF8C

EBP 0022FF94

ESI 00000000

EDI 00000000

EIP 004014E0 Project1.<ModuleEntryPoint>

C 0 ES 0023 32bit 0(FFFFFFFF)

P 1 CS 001B 32bit 0(FFFFFFFF)

A 0 SS 0023 32bit 0(FFFFFFFF)

Z 1 DS 0023 32bit 0(FFFFFFFF)

S 0 FS 003B 32bit 7FFDF000(FFF)

T 0 GS 0000 NULL

D 0

O 0 LastErr ERROR_SUCCESS (00000000)

EFL 00000246 (NO,NB,E,BE,NS,PE,GE,LE)

ST0 empty 0.0

ST1 empty 0.0

ST2 empty 0.0

ST3 empty 0.0

ST4 empty 0.0

ST5 empty 0.0

ST6 empty 0.0

ST7 empty 0.0

3 2 1 0 E S P U O Z D I

FST 0000 Cond 0 0 0 0 Err 0 0 0 0 0 0 0 (GT)

FCW 027F Prec NEAR,53 Mask 1 1 1 1 1

Address Hex dump ASCII

00403000 0A 00 00 00 FF 00 00 00

00403008 FF FF FF FF FF FF FF FF

00403010 5C 26 40 00 02 00 00 00 \&@.0...

00403018 FF FF FF FF 90 25 40 00 E%0.

00403020 A0 25 40 00 4E E6 40 BB a%0.Nu0n

00403028 B1 19 BF 44 00 00 00 00 |||D....

00403030 00 00 00 00 00 00 00 00

00403038 00 00 00 00 00 00 00 00

0022FF8C 758EEE6C RETURN to kernel32.758EEE6C

0022FF90 7FFDB000

0022FF94 0022FFD4

0022FF98 76EF3AB3 RETURN to ntdll.76EF3AB3

0022FF9C 7FFDB000

0022FFA0 76D724B9

0022FFA4 00000000

0022FFA8 00000000

0022FFAC 7FFDB000

실습1_출력 문자 바꾸기 (1)

OllyDbg - Project1.exe - [CPU - main thread, module Project1]

File View Debug Plugins Options Window Help

004014E0 \$ 83EC 0C SUB ESP,0C

004014E3 . C705 34504000 MOV DWORD PTR DS:[405034],0

004014ED . E8 9E090000 CALL Project1.00401E90

004014F2 . 83C4 0C ADD ESP,0C

004014F5 . E9 86FCFFFF JMP Project1.00401180

004014FA 90 NOP

004014FB 90 NOP

004014FC 90 NOP

004014FD 90 NOP

004014FE 90 NOP

004014FF 90 NOP

00401500 \$ 55 PUSH EBP

00401501 . 89E5 MOV EBP,ESP

00401503 . 83E4 F0 AND ESP,FFFFFF0

00401506 . 83EC 10 SUB ESP,10

00401509 . E8 62090000 CALL Project1.00401E70

0040150E . C70424 00404000 MOV DWORD PTR SS:[ESP],Project1.00404000

00401515 . E8 CE100000 CALL <JMP.&msvcrt.puts>

0040151A . C9 LEAVE

0040151B . C3 RETN

0040151C . 66:90 NOP

0040151E . 66:90 NOP

00401520 . 83EC 1C SUB ESP,1C

00401523 . 8B4424 24 MOV EAX,DWORD PTR DS:[404024]

00401527 . 85C0 TEST EAX,EAX

00401529 . 74 15 JE SHORT Project1.0040153B

0040152B . 83F8 03 CMP EAX,3

0040152E . 74 10 JE SHORT Project1.0040153B

00401530 . B8 01000000 MOV EAX,1

00401535 . 83C4 1C ADD ESP,1C

00401538 . C2 0C00 RETN 0C

0040153B 90 NOP

0040153C . 8D7426 00 LEA ESI,DWORD PTR DS:[ESI]

00401540 . 8B5424 28 MOV EDX,DWORD PTR DS:[ESP+28]

00401544 . 894424 04 MOV DWORD PTR SS:[ESP+4],EAX

00401548 . 8B4424 20 MOV EAX,DWORD PTR SS:[ESP+20]

0040154C . 895424 08 MOV DWORD PTR SS:[ESP+8],EDX

00401550 . 890424 MOV DWORD PTR SS:[ESP],EAX

00401553 . E8 080C0000 CALL Project1.00402160

00401558 . B8 01000000 MOV EAX,1

0040155D . 83C4 1C ADD ESP,1C

Registers (FPU)

EAX 758EEE5A kernel32.BaseThreadInitThunk

ECX 00000000

EDX 004014E0 Project1.<ModuleEntryPoint>

EBX 7FFD8000

ESP 0022FF8C

EBP 0022FF94

ESI 00000000

EDI 00000000

EIP 004014E0 Project1.<ModuleEntryPoint>

C 0 ES 0023 32bit 0(FFFFFFFF)

P 1 CS 001B 32bit 0(FFFFFFFF)

A 0 SS 0023 32bit 0(FFFFFFFF)

Z 1 DS 0023 32bit 0(FFFFFFFF)

S 0 FS 003B 32bit 7FFDF000(FFF)

T 0 GS 0000 NULL

D 0

ESP=0022FF8C

DWORD PTR SS: SS(stack segment)의 크기를 DWORD(4Byte)로 재설정함
→ stack segment register에 해당 영역(ss)의 시작주소를 저장함

MOV [ESP], Project1.00404000: ESP주소가 가리키는 값에 Project1.00404000
를 넣음. stack의 제일 아래부분(낮은 주소지),
현재 진행되는 stack 지점

실습1_출력 문자 바꾸기 (1)

Address	Hex dump	ASCII
00404000	48 65 6C 6C 6F 20 57 6F	Hello Wo
00404008	72 6C 64 7E 00 00 00 00	rld~....
00404010	70 15 40 00 55 6E 6B 6E	pS@.Unkn
00404018	6F 77 6E 20 65 72 72 6F	own erro
00404020	72 00 00 00 5F 6D 61 74	r..._mat
00404028	68 65 72 72 28 29 3A 20	herr():
00404030	25 73 20 69 6E 20 25 73	%s in %s
00404038	28 25 67 2C 20 25 67 29	(%g, %g)
00404040	20 20 28 72 65 74 76 61	(retva
00404048	6C 3D 25 67 29 0A 00 00	l=%g)...
00404050	41 72 67 75 6D 65 6E 74	Argument
00404058	20 64 6F 6D 61 69 6E 20	domain
00404060	65 72 72 6F 72 20 28 44	error (D
00404068	4F 4D 41 49 4E 29 00 41	OMAIN).A
00404070	72 67 75 6D 65 6E 74 20	rgument
00404078	73 69 6E 67 75 6C 61 72	singular
00404080	69 74 79 20 28 53 49 47	ity (SIG
00404088	4E 29 00 00 4F 76 65 72	N)..Over
00404090	66 6C 6F 77 20 72 61 6E	flow ran
00404098	67 65 20 65 72 72 6F 72	ge error
004040A0	20 28 4F 56 45 52 46 4C	(OVERFL

space bar 누르고 변경

Edit data at 00404000

ASCII

H

UNICODE

?

HEX +00

48

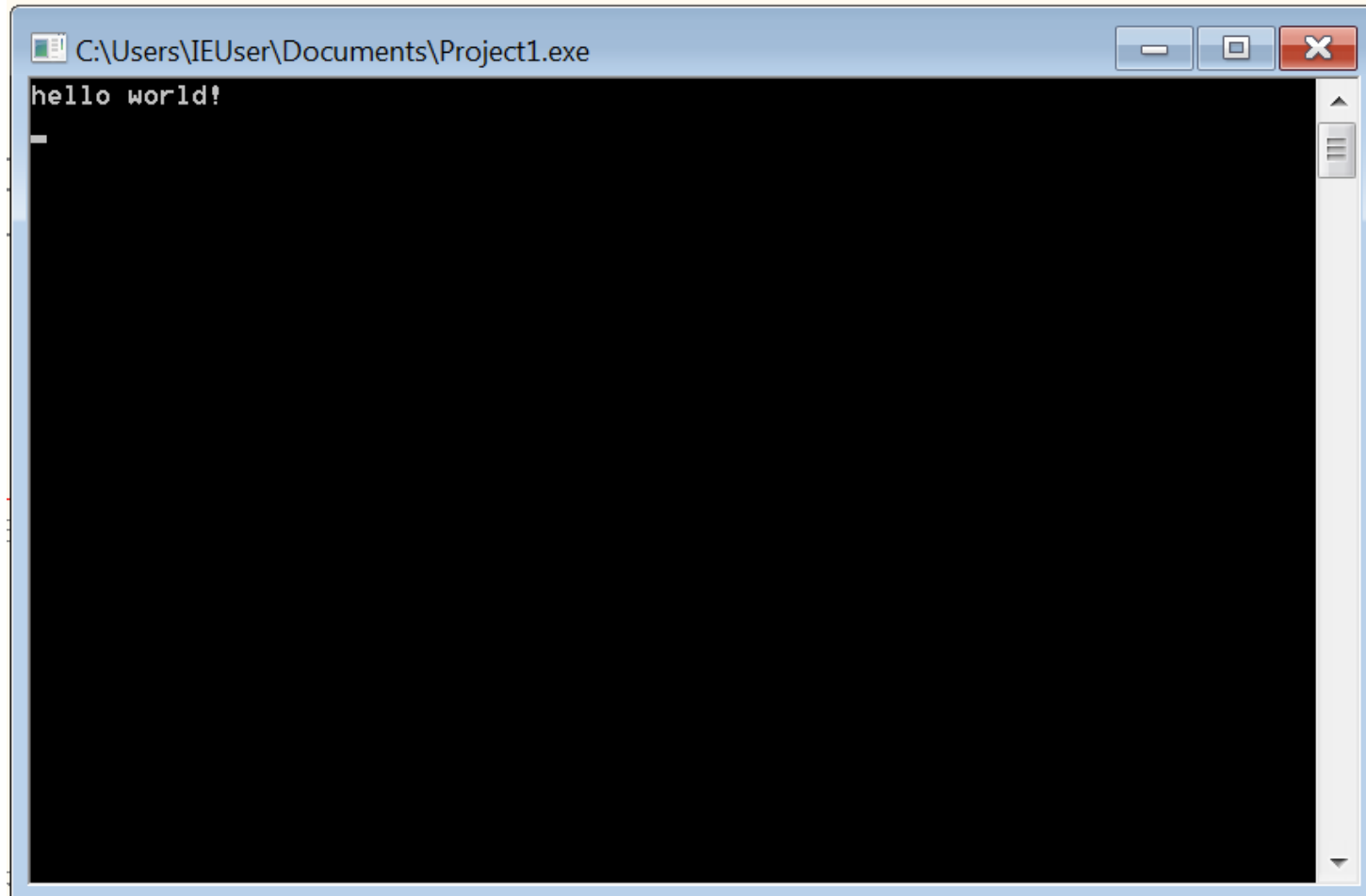
☒ Keep size

OK

Cancel

Address	Hex dump	ASCII
00404000	68 65 6C 6C 6F 20 77 6F	hello wo
00404008	72 6C 64 21 00 00 00 00	rld!....
00404010	70 15 40 00 55 6E 6B 6E	pS@.Unkn
00404018	6F 77 6E 20 65 72 72 6F	own erro
00404020	72 00 00 00 5F 6D 61 74	r..._mat
00404028	68 65 72 72 28 29 3A 20	herr():
00404030	25 73 20 69 6E 20 25 73	%s in %s
00404038	28 25 67 2C 20 25 67 29	(%g, %g)
00404040	20 20 28 72 65 74 76 61	(retva
00404048	6C 3D 25 67 29 0A 00 00	l=%g)...
00404050	41 72 67 75 6D 65 6E 74	Argument
00404058	20 64 6F 6D 61 69 6E 20	domain
00404060	65 72 72 6F 72 20 28 44	error (D
00404068	4F 4D 41 49 4E 29 00 41	OMAIN).A
00404070	72 67 75 6D 65 6E 74 20	rgument
00404078	73 69 6E 67 75 6C 61 72	singular
00404080	69 74 79 20 28 53 49 47	ity (SIG
00404088	4E 29 00 00 4F 76 65 72	N)..Over
00404090	66 6C 6F 77 20 72 61 6E	flow ran
00404098	67 65 20 65 72 72 6F 72	ge error
004040A0	20 28 4F 56 45 52 46 4C	(OVERFL

실습1_출력 문자 바꾸기 (1)

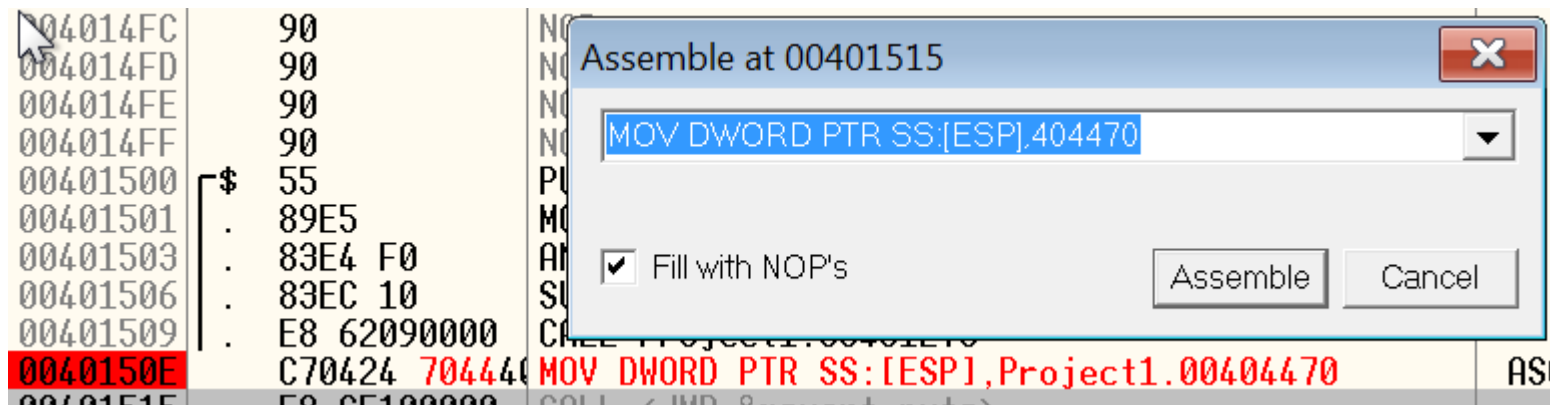
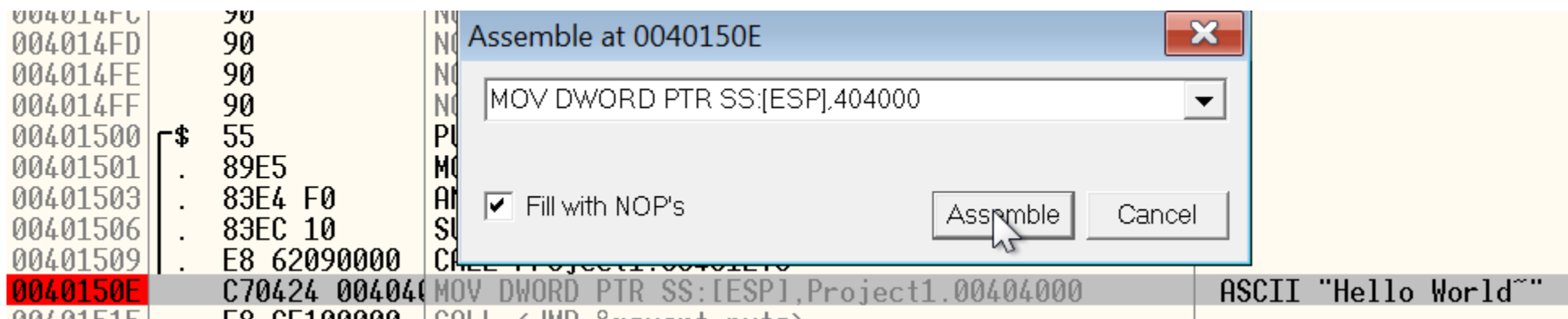


실습2_출력 문자 바꾸기 (2)

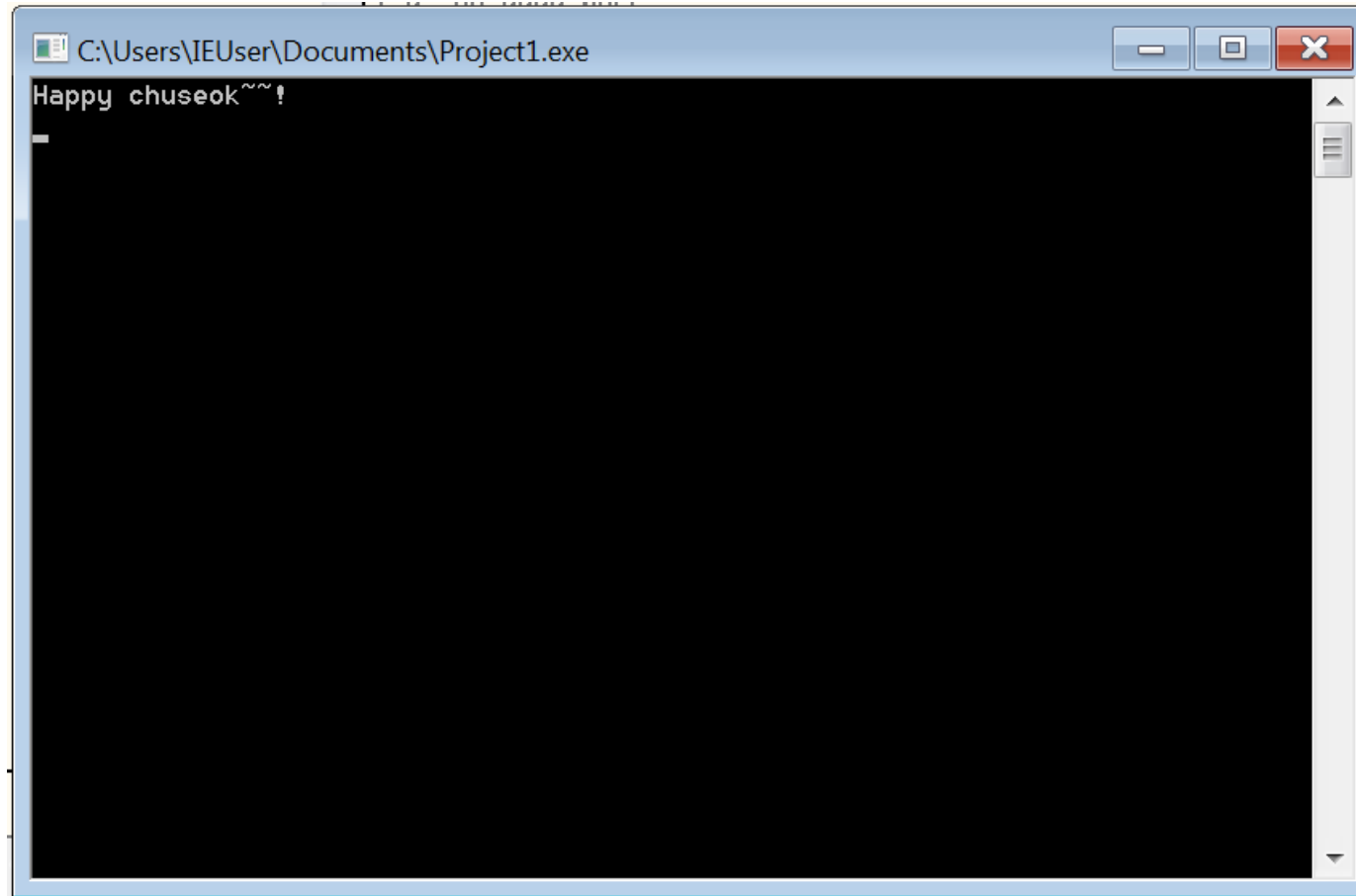
Address	Hex dump	ASCII
00404470	00 00 00 00 00 00 00 00
00404478	00 00 00 00 00 00 00 00
00404480	00 00 00 00 00 00 00 00
00404488	00 00 00 00 00 00 00 00
00404490	00 00 00 00 00 00 00 00
00404498	00 00 00 00 00 00 00 00
004044A0	00 00 00 00 00 00 00 00
004044A8	00 00 00 00 00 00 00 00
004044B0	00 00 00 00 00 00 00 00
004044B8	00 00 00 00 00 00 00 00
004044C0	00 00 00 00 00 00 00 00
004044C8	00 00 00 00 00 00 00 00
004044D0	00 00 00 00 00 00 00 00
004044D8	00 00 00 00 00 00 00 00
004044E0	00 00 00 00 00 00 00 00
004044E8	00 00 00 00 00 00 00 00
004044F0	00 00 00 00 00 00 00 00
004044F8	00 00 00 00 00 00 00 00
00404500	00 00 00 00 00 00 00 00
00404508	00 00 00 00 00 00 00 00
00404510	00 00 00 00 00 00 00 00

Address	Hex dump	ASCII
00404470	48 61 70 70 79 20 63 68	Happy ch
00404478	75 73 65 6F 6B 7E 7E 21	useok~~!
00404480	00 00 00 00 00 00 00 00
00404488	00 00 00 00 00 00 00 00
00404490	00 00 00 00 00 00 00 00
00404498	00 00 00 00 00 00 00 00
004044A0	00 00 00 00 00 00 00 00
004044A8	00 00 00 00 00 00 00 00
004044B0	00 00 00 00 00 00 00 00
004044B8	00 00 00 00 00 00 00 00
004044C0	00 00 00 00 00 00 00 00
004044C8	00 00 00 00 00 00 00 00
004044D0	00 00 00 00 00 00 00 00
004044D8	00 00 00 00 00 00 00 00
004044E0	00 00 00 00 00 00 00 00
004044E8	00 00 00 00 00 00 00 00
004044F0	00 00 00 00 00 00 00 00
004044F8	00 00 00 00 00 00 00 00
00404500	00 00 00 00 00 00 00 00
00404508	00 00 00 00 00 00 00 00
00404510	00 00 00 00 00 00 00 00

실습2_출력 문자 바꾸기 (2)



실습2_출력 문자 바꾸기 (2)



감사합니다