

따라 하면서 배우는 IT

기본적으로 알아야 할
컴퓨터 기초

목차

INDEX

HW
구성요소

OS
운영 체제

파일의 특징

프로그램의 특징

CPU
메모리
하드디스크
I/O 장치

OS란?
OS의 구성요소

데이터 표현
파일의 이름
파일의 경로

프로그램이란?
프로그램을 만들려면?

따라 하면서 배우는 IT

HW 구성요소

HW 구성요소

CPU

//

중앙 처리 장치
CPU

//



HW 구성요소

주 기억 장치

//

주 기억 장치
메모리
RAM

//



HW 구성요소

보조 기억 장치

//

보조 기억 장치
하드 디스크
SSD

//



HW 구성요소

입출력 장치

//

입출력 장치
I/O 장치

//



따라 하면서 배우는 IT

OS 운영 체제

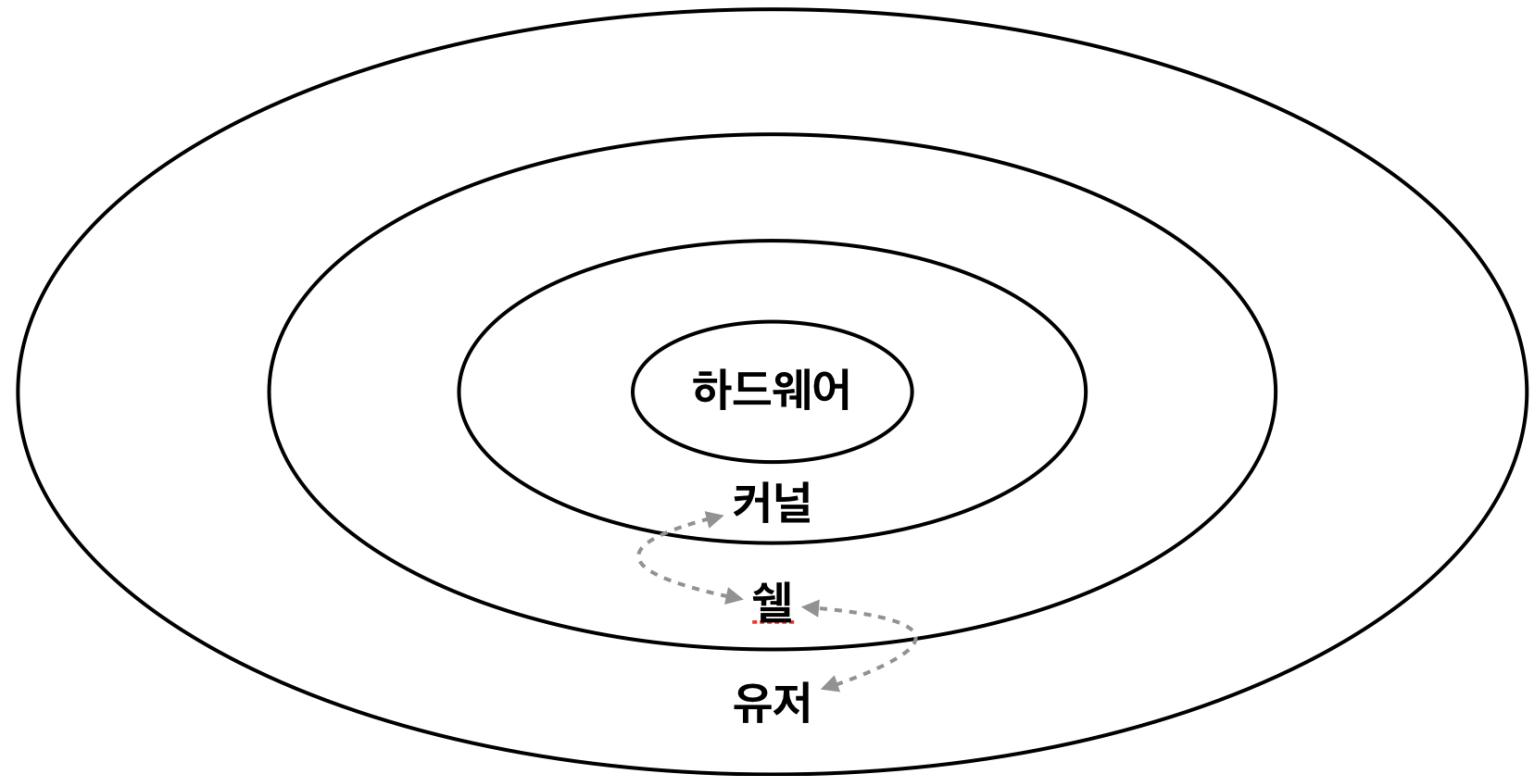
OS 운영 체제

OS란?

//

하드웨어를 관리해주는
프로그램

//



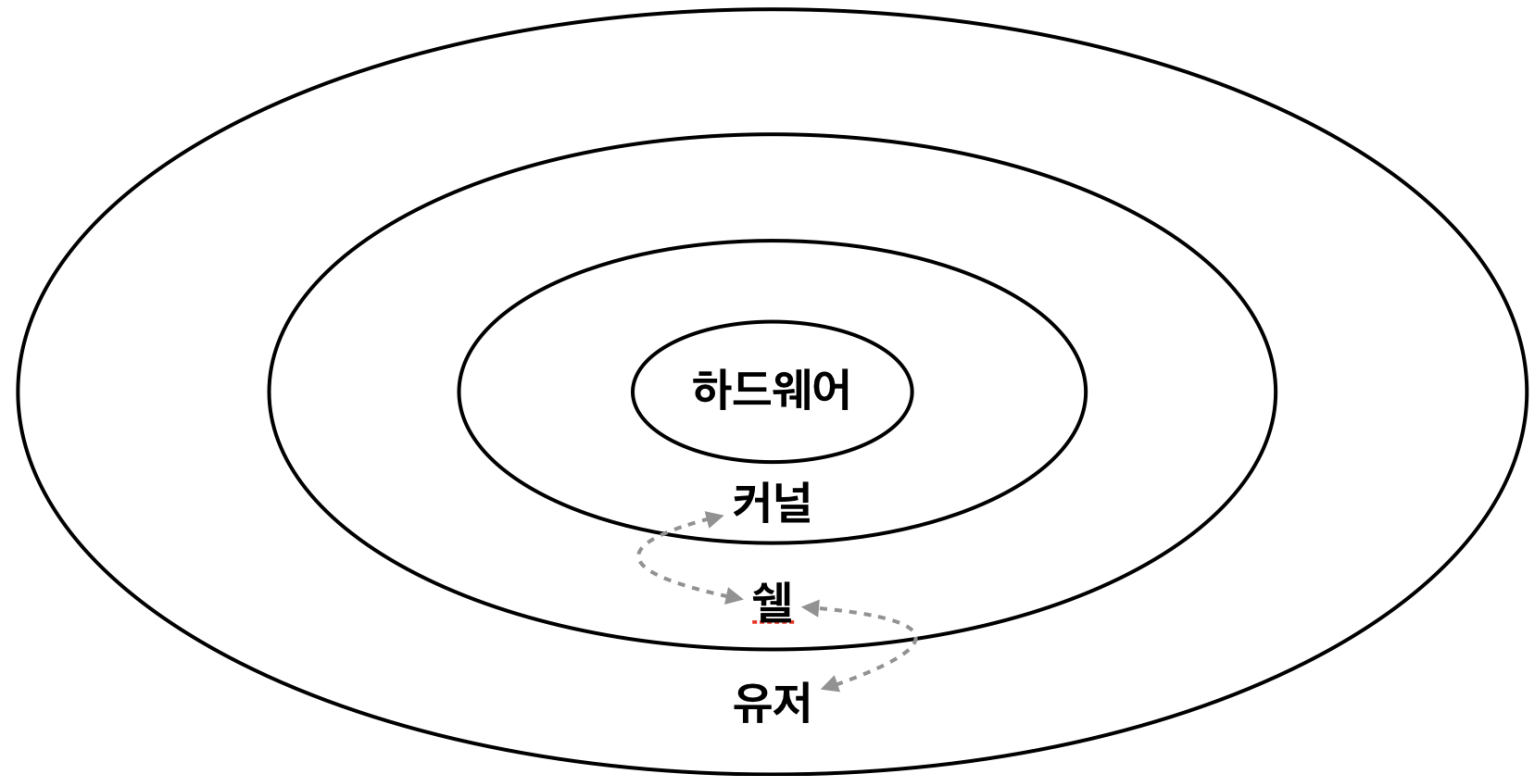
OS 운영 체제

OS의 구성요소

//

커널, 셸, 응용 프로그램

//



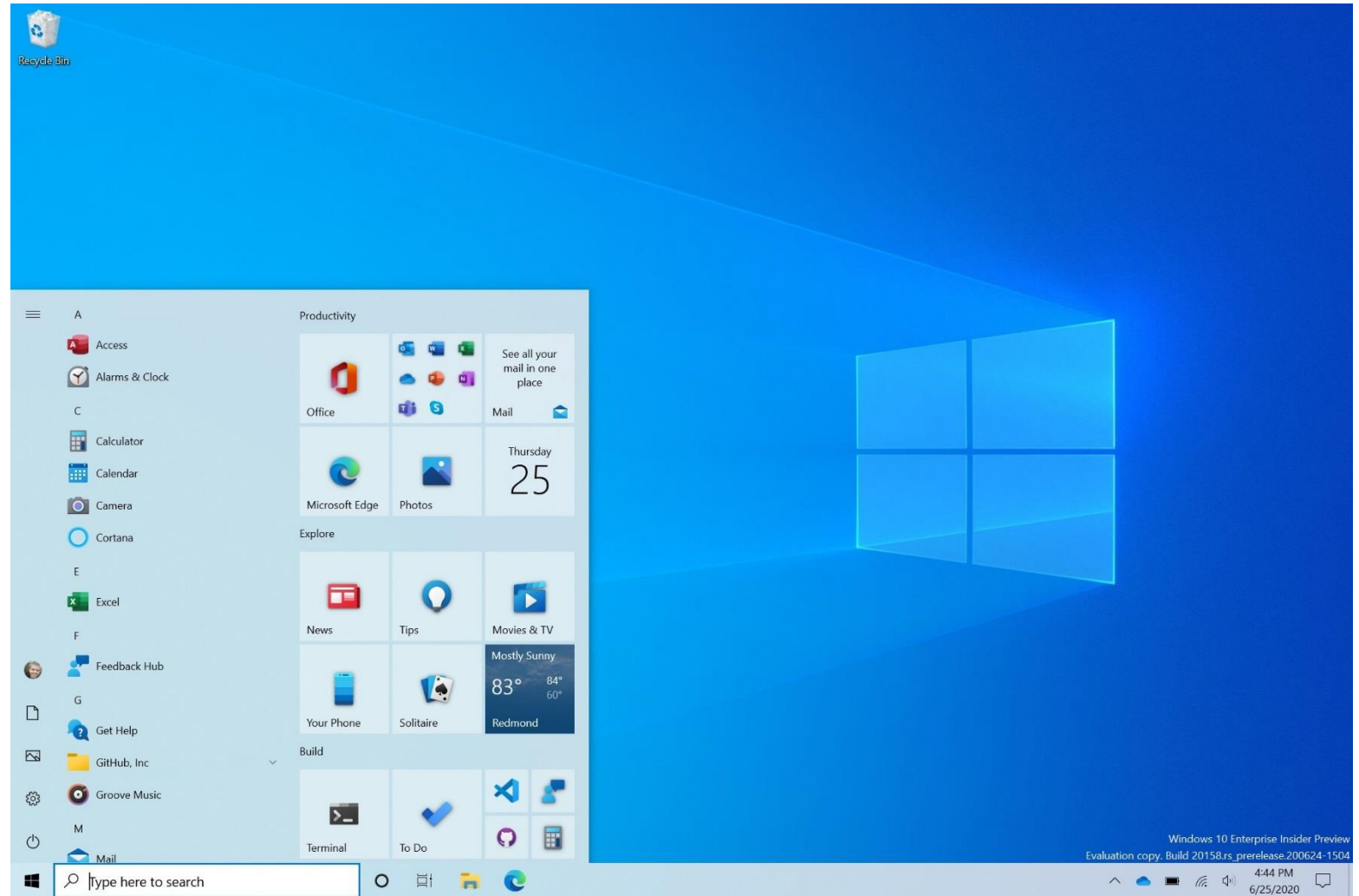
OS 운영 체제

OS의 구성요소

//

윈도우

//



OS 운영 체제

OS의 구성요소

//

리눅스

//

```
tecmin@ubuntu: ~  
tecmin@ubuntu:~$ sudo apt install cassandra  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following additional packages will be installed:  
  libpython2-stdlib libpython2.7-minimal libpython2.7-stdlib python-is-python2  
  python2 python2-minimal python2.7 python2.7-minimal  
Suggested packages:  
  cassandra-tools python2-doc python-tk python2.7-doc binfmt-support  
The following NEW packages will be installed:  
  cassandra libpython2-stdlib libpython2.7-minimal libpython2.7-stdlib  
  python-is-python2 python2 python2-minimal python2.7 python2.7-minimal  
0 upgraded, 9 newly installed, 0 to remove and 253 not upgraded.  
Need to get 34.4 MB of archives.  
After this operation, 57.0 MB of additional disk space will be used.  
Do you want to continue? [Y/n] Y
```

따라 하면서 배우는 IT

파일의 특징

파일의 특징

데이터 표현

//

컴퓨터가 사용하는 2진법
다양한 진법
사람이 사용하는 10진법

//

- 진법 변환
 - 10진 변환

$$972 = 9 \times 10^2 + 7 \times 10^1 + 2 \times 10^0$$

- 2진수 \rightarrow 10진수 변환

$$\begin{aligned} 11101_{(2)} &= 1 \times 2^4 + 1 \times 2^3 \\ &\quad + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\ &= 16 + 8 + 4 + 0 + 1 \\ &= 29 \end{aligned}$$

데이터 표현

문자 표현

//

컴퓨터가 사용하는 2진법

다양한 진법

사람이 사용하는 10진법

//

10진 수	16진 수	8진 수	2진 수	ASCII
64	0x40	100	1000000	@
65	0x41	101	1000001	A
66	0x42	102	1000010	B
67	0x43	103	1000011	C
68	0x44	104	1000100	D
69	0x45	105	1000101	E
70	0x46	106	1000110	F
71	0x47	107	1000111	G
72	0x48	110	1001000	H
73	0x49	111	1001001	I
74	0x4A	112	1001010	J
75	0x4B	113	1001011	K
76	0x4C	114	1001100	L
77	0x4D	115	1001101	M
78	0x4E	116	1001110	N
79	0x4F	117	1001111	O
80	0x50	120	1010000	P

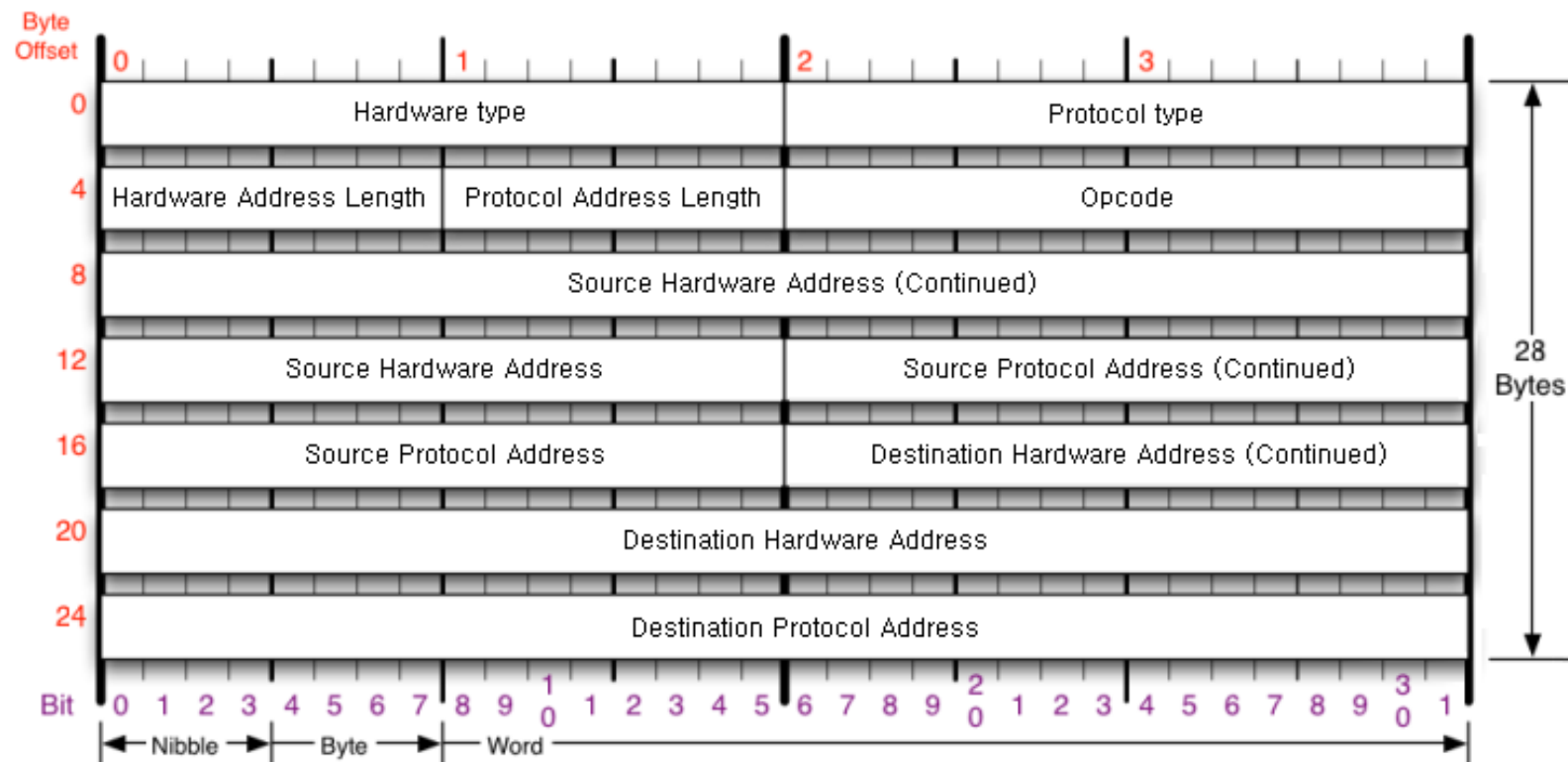
데이터 표현

네트워크 데이터

//

컴퓨터가 사용하는 2진법
다양한 진법
사람이 사용하는 10진법

//



데이터 표현

네트워크 데이터

//

컴퓨터가 사용하는 2진법
다양한 진법
사람이 사용하는 10진법

//

```
▼ Destination: Broadcast (ff:ff:ff:ff:ff:ff)
  Address: Broadcast (ff:ff:ff:ff:ff:ff)
    .... ..1. .... .. = LG bit: Locally administered address (this is NOT the factory default)
    .... ..1 .... .. = IG bit: Group address (multicast/broadcast)
▼ Source: SamsungE_85:19:fe (98:83:89:85:19:fe)
  Address: SamsungE_85:19:fe (98:83:89:85:19:fe)
    .... ..0. .... .. = LG bit: Globally unique address (factory default)
    .... ..0 .... .. = IG bit: Individual address (unicast)
  Type: ARP (0x0806)
  Padding: 00000000000000000000000000000000
▼ Address Resolution Protocol (request)
  Hardware type: Ethernet (1)
  Protocol type: IPv4 (0x0800)
  Hardware size: 6
  Protocol size: 4
  Opcode: request (1)
  Sender MAC address: SamsungE_85:19:fe (98:83:89:85:19:fe)
  Sender IP address: 10.0.0.254
  Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
  Target IP address: 10.0.0.34
```

0000	ff ff ff	ff ff ff 98 83	89 85 19 fe 08 06 00 01	...
0010	08 00 06 04 00 01 98 83	89 85 19 fe 0a 00 00 fe	
0020	00 00 00 00 00 00 0a 00	00 22 00 00 00 00 00 00"
0030	00 00 00 00 00 00 00 00	00 00 00 00	

파일의 특징

파일의 이름

//

다양한 형식의
파일의 이름

//



파일의 특징

파일의 이름

//

다양한 형식의
파일의 이름

//

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
00000000	4D	5A	90	00	03	00	00	00	04	00	00	00	FF	FF	00	00	MZ yy
00000010	B8	IMAGE_DOS_HEADER							00	40	00	00	00	00	00	00	@
00000020	00								00	00	00	00	00	00	00	00	
00000030	00	00	00	00	00	00	00	00	00	00	00	00	E8	00	00	00	e
00000040	0E	1F	BA	0E	00	B4	09	CD	21	B8	01	4C	CD	e_ifanew->OFFSET NT_HEADERS			!Th
00000050	69	73	20	70	72	6F	67	72	61	6D	20	63	61	6E	6E	6F	is program canno
00000060	74	20	62	65	20	72	75	6E	20	69	6E	20	44	4F	53	20	t be run in DOS
00000070	6D	6F	64	65	2E	0D	0D	0A	24	00	00	00	00	00	00	00	mode. \$
00000080	29	DD	48	3E	6D	BC	26	6D	6D	BC	26	6D	6D	BC	26	6D)YH>m%sm%sm%sm
00000090	02	CA	BA	6D	6F	BC	26	6D	02	CA	8C	6D	7F	BC	26	6D	Êmo%sm Ê!m %sm
000000A0	64	도스용 코드				BC	26	6D	6D	BC	27	6D	0A	BC	26	6D	dÄjmd%sm%sm%sm
000000B0	02	CA	8D	6D	63	BC	26	6D	02	CA	BC	6D	6C	BC	26	6D	Ê mc%sm Ê%ml%sm
000000C0	6D	BC	B1	6D	6C	BC	26	6D	02	CA	BB	6D	6C	BC	26	6D	m%+ml%sm Ê>ml%sm
000000D0	52	69	63	68	6D	BC	26	6D	00	00	00	00	00	00	00	00	Richm%sm
000000E0	00	00	00	00	00	00	00	00	50	45	00	00	4C	01	07	00	PE L

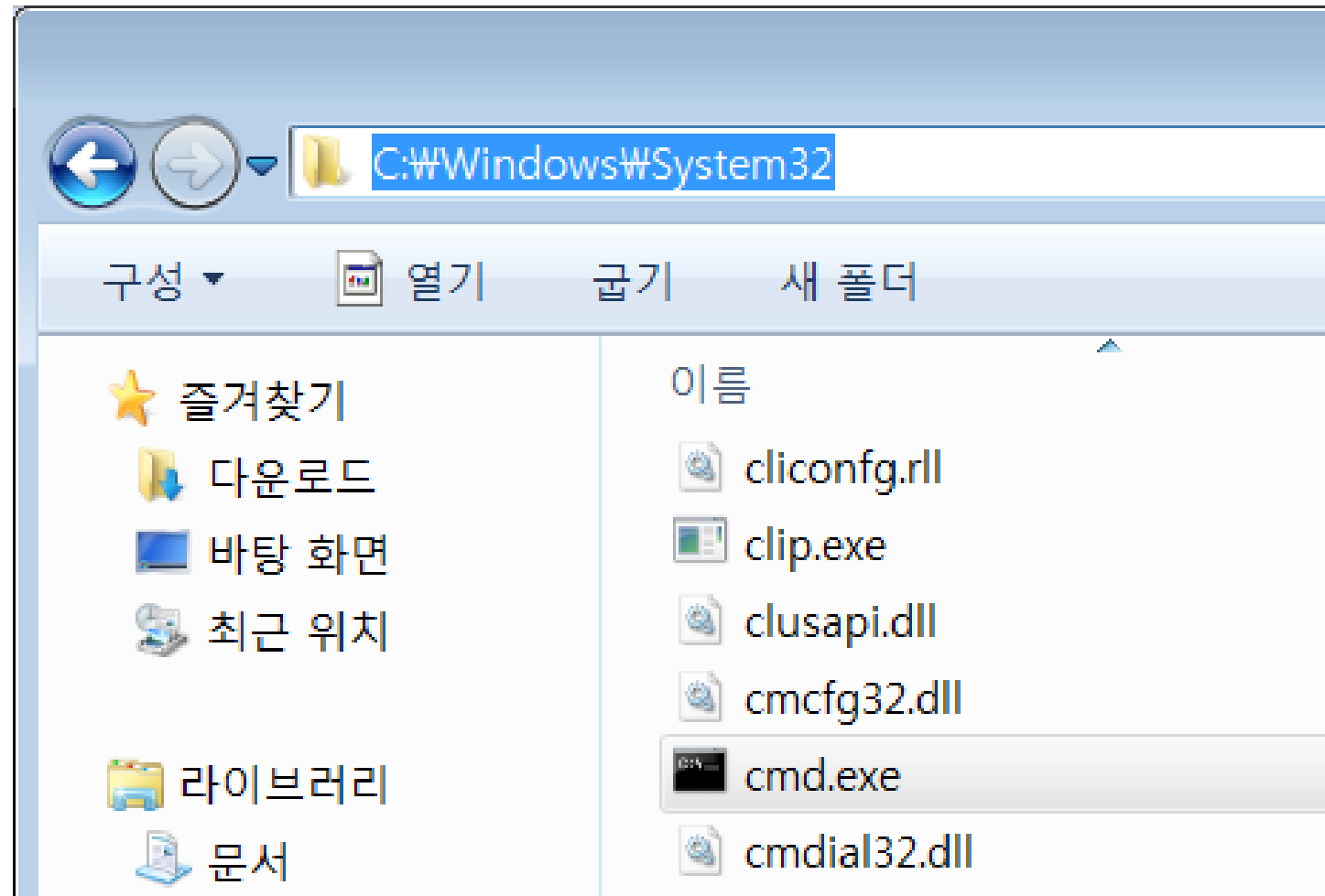
파일의 특징

파일의 경로

//

다양하게 지정할 수 있는
파일의 경로

//



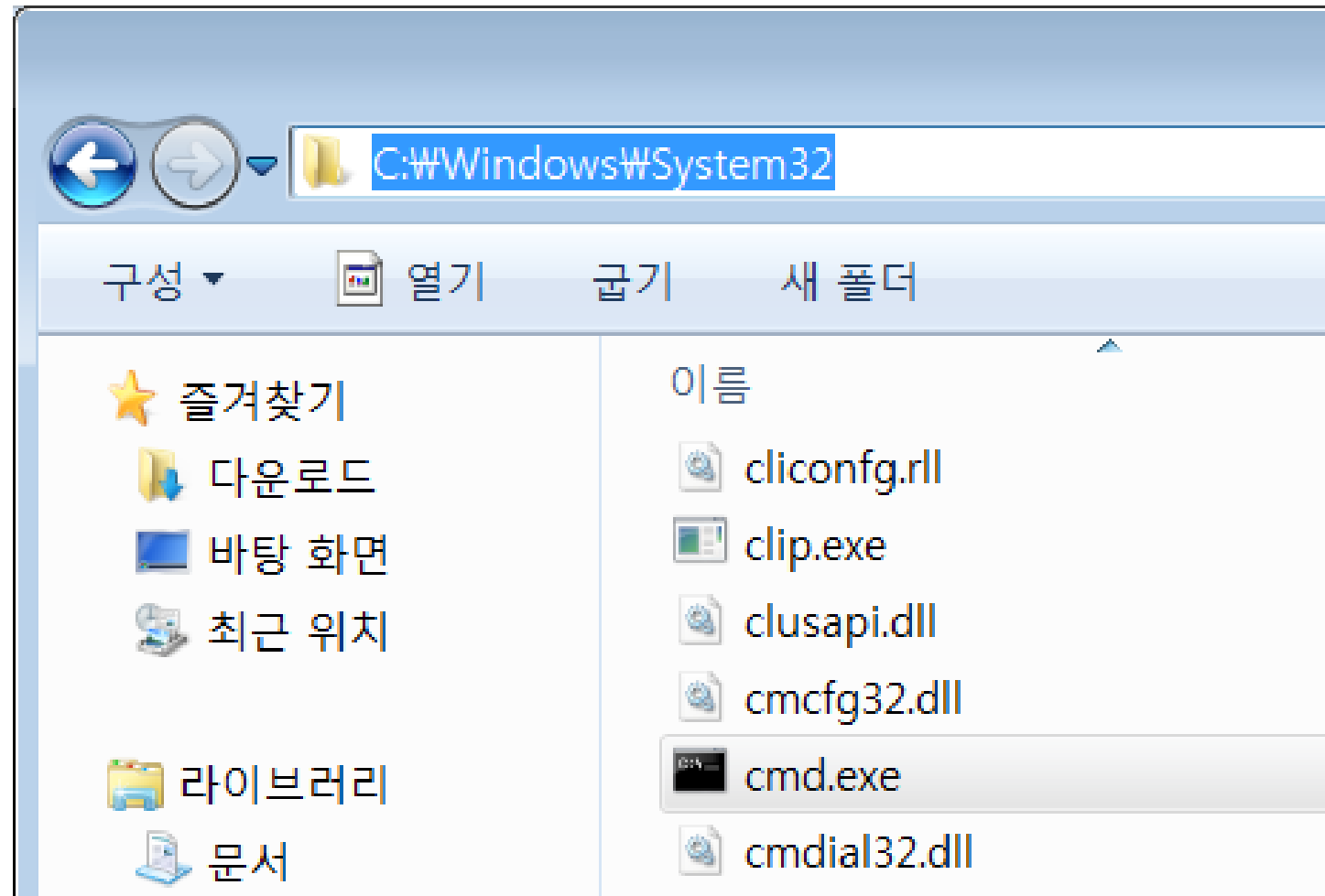
파일의 특징

파일의 경로

//

경로를 모두 입력해야하는
파일의 실행

//



따라 하면서 배우는 IT

프로그램의 특징

프로그램의 특징

프로그램이란?

//

다양한 작업을 수행하는
명령어의 집합

//

Immunity Debugger - abex1.exe - [CPU - main thread, module abex1]

File View Debug Plugins ImmLib Options Window Help Jobs

White Phosphorus

00401000 6A 00 PUSH 0
00401002 68 00204000 PUSH abex1.00402000
00401007 68 12204000 PUSH abex1.00402012
0040100C 6A 00 PUSH 0
0040100E E8 4E000000 CALL <JMP.&USER32.MessageBoxA>
00401013 68 24204000 PUSH abex1.00402094
00401018 E8 38000000 CALL <JMP.&KERNEL32.GetDriveTypeA>
0040101D 46 INC ESI
0040101E 48 DEC EAX
0040101F EB 00 JMP SHORT abex1.00401021
00401021 46 INC ESI
00401022 46 INC ESI
00401023 48 DEC EAX
00401024 3BC6 CMP EAX,ESI
00401026 74 15 JE SHORT abex1.0040103D
00401028 6A 00 PUSH 0
0040102A 68 35204000 PUSH abex1.00402035
0040102F 68 3B204000 PUSH abex1.0040203B
00401034 6A 00 PUSH 0
00401036 E8 26000000 CALL <JMP.&USER32.MessageBoxA>
0040103B EB 13 JMP SHORT abex1.00401050
0040103D 6A 00 PUSH 0
0040103F 68 5E204000 PUSH abex1.0040205E
00401044 68 64204000 PUSH abex1.00402064
00401049 6A 00 PUSH 0
0040104B E8 11000000 CALL <JMP.&USER32.MessageBoxA>
00401050 E8 06000000 CALL <JMP.&KERNEL32.ExitProcess>
00401055 FF25 50304000 JMP DWORD PTR DS:[&KERNEL32.GetDriveTypeA
0040105B FF25 54304000 JMP DWORD PTR DS:[&KERNEL32.ExitProcess
00401061 FF25 5C304000 JMP DWORD PTR DS:[&USER32.MessageBoxA
00401067 00 DB 00
00401068 00 DB 00
00401069 00 DB 00
0040106A 00 DB 00
0040106B 00 DB 00
0040106C 00 DB 00
0040106D 00 DB 00
0040106E 00 DB 00
0040106F 00 DB 00
00401070 00 DB 00
00401071 00 DB 00
00401072 00 DB 00

Registers (FPU)
EAX 00000000
ECX 0012FFB0
EDI 7C98E514 ntdll.KiFastS
EBX 7FFD4000
ESP 0012FFC4
EBP 0012FFF0
ESI 00720065
EDI 00670067
EIP 00401000 abex1.<Module
C 0 ES 0023 32bit 0(FFFFF
P 1 CS 001B 32bit 0(FFFFF
A 0 SS 0023 32bit 0(FFFFF
Z 1 DS 0023 32bit 0(FFFFF
S 0 FS 003B 32bit 7FFDF00
T 0 GS 0000 NULL
0 0 LastErr ERROR_MOD_NOT
EFL 00000246 (NO,NB,E,BE,N
ST0 empty 0.00000000000000
ST1 empty 0.00000000000000
ST2 empty 0.00000000000000
ST3 empty 0.00000000000000
ST4 empty 0.00000000000000
ST5 empty 0.00000000000000
ST6 empty 0.00000000000000
ST7 empty 1.25197751666951
FST 0000 Cond 0 0 0 0 Ex
FCW 027F Prec NEAR,53 Ma

Address Hex dump ASCII
00402000 61 62 65 78 27 20 31 73 abex' 1s
00402008 74 20 63 72 61 63 6B 6D t crackm

Show Breakpoints window (Alt+B) Paused

프로그램의 특징

프로그램을 만들려면?

//

특정 프로그래밍 언어를 활용
명령어의 집합
을 만든다

//

```
def add5(x):  
    return x+5  
  
def dotwrite(ast):  
    nodename = getNodeName()  
    label=symbol.sym_name.get(int(ast[0]),ast[0])  
    print '      %s [label="%s' % (nodename, label),  
    if isinstance(ast[1], str):  
        if ast[1].strip():  
            print '= %s";' % ast[1]  
        else:  
            print '"]'  
    else:  
        print '"]';'  
        children = []  
        for n, child in enumerate(ast[1:]):  
            children.append(dotwrite(child))  
    print ',      %s -> {' % nodename  
    for n, child in enumerate(children):  
        print '%s' % child,
```