Tracer

- 동적 분석 툴

이름 : 황 상 두
전화번호 : 010-4082-8382
이메일 : ghkdtkden93@gmail.com

목차

DBI란?

➡ *D*ynamic *B*inary *I*nstrumentation Tool

(Run Time)중 (기계어)를 (구성)하겠다
(실행) (삽입)

Made in intel Pin

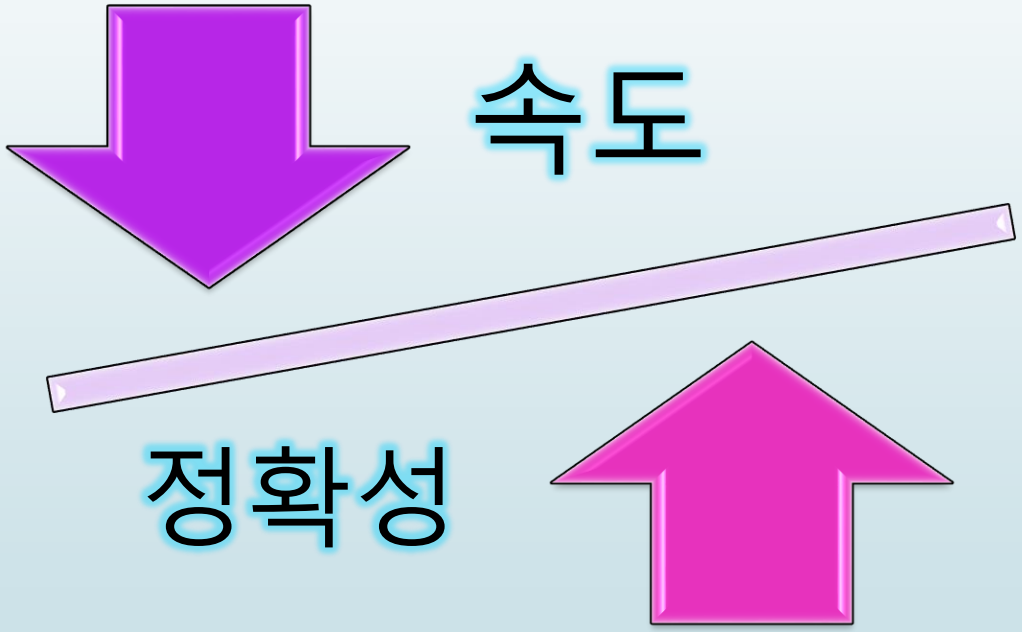
DownLoad ▼▼

http://software.intel.com/en-us/articles/pintool-downloads

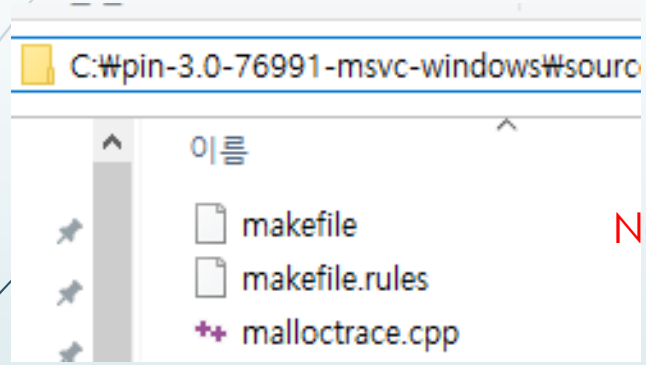
DBI 장단점

실행 중 code 삽입

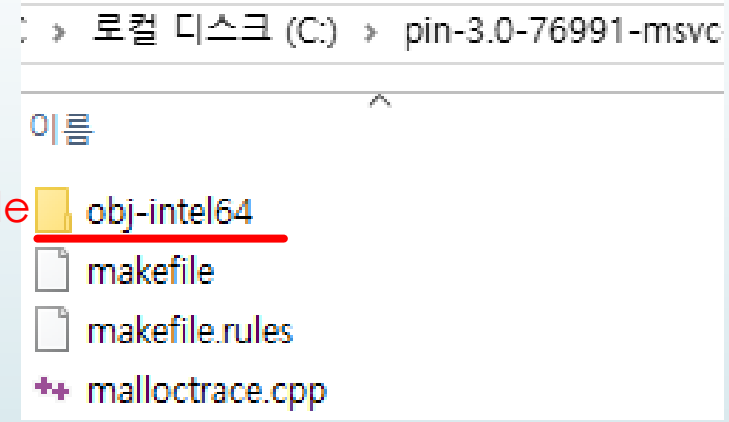
Recompile , Relink (X)

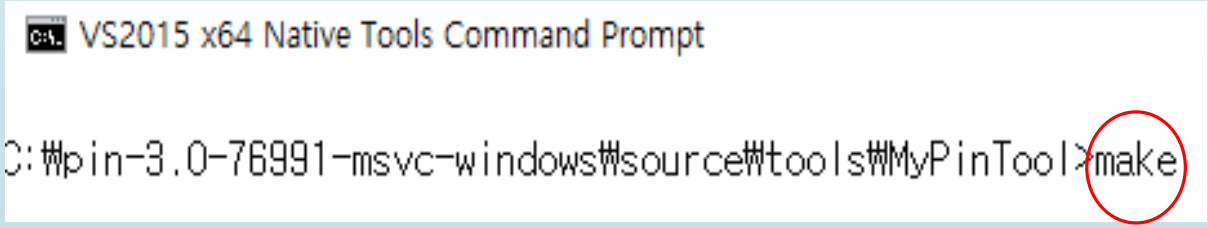


컴파일


C:₩pin-3.0-76991-msvc-windows₩sourc
이름
makefile
makefile.rules
malloctrace.cpp

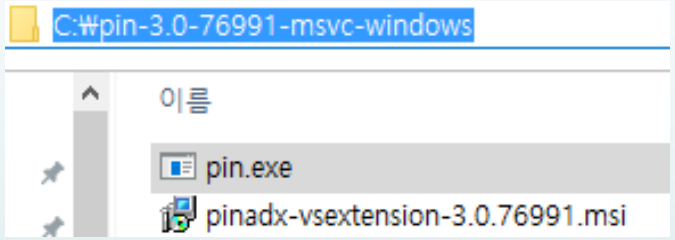
New File


로컬 디스크 (C:) › pin-3.0-76991-msvc
이름
obj-intel64
makefile
makefile.rules
malloctrace.cpp


VS2015 x64 Native Tools Command Prompt
C:₩pin-3.0-76991-msvc-windows₩source₩tools₩MyPinTool>make

사용법

1. Pin.exe 환경변수 지정



2. Pin -t [.dll] -- [.exe]

```
Tools Command Prompt

msvc-windows₩source₩tools₩MyPinTool>pin -t obj-intel64₩MyPinTool.dll -- notepad.exe
```

명령어 count

```
>cd C:₩Users₩haha₩Desktop₩공부자료₩pin-3.0-76991-msvc-windows₩source₩tools₩ManualExamples
```

« haha › Desktop › 공부자료 › pin-3.0-76991-msvc-windows › source › tools › ManualExamples ›

이름	수정한 날짜	유형	크기
obj-intel64	2016-07-29 오후...	파일 폴더	
			293,670KB
			10KB
			9KB
		C++ Source	23KB
		C++ Source	4KB
		C Source	5KB
		C Source	3KB
divide_by_zero_win.obj	2016-07-04 오전...	Object File	4KB
emudiv.cpp	2016-06-30 오후...	C++ Source	7KB
fibonacci.cpp	2016-06-30 오후...	C++ Source	2KB

```
>pin -t obj-intel64₩inscount0.dll -- notepad.exe
```

Pin -t [.dll] -- [.exe]

실행결과

« haha › Desktop › 공부자료 › pin-3.0-76991-msvc-windows › source › tools › ManualExamples

이름	수정한 날짜	유형	크기
follow_child_app2.cpp	2016-06-30 오후...	C++ Source	2KB
follow_child_tool.cpp	2016-06-30 오후...	C++ Source	3KB
fork_app.cpp	2016-06-30 오후...	C++ Source	2KB
fork_jit_tool.cpp	2016-07-29 오후...	C++ Source	4KB
imageload.cpp	2016-06-30 오후...	C++ Source	4KB
inscount.out	2016-08-08 오후...	OUT 파일	1KB
inscount_tls.cpp	2016-06-30 오후...	C++ Source	6KB
inscount_tls.out	2016-07-29 오후...	OUT 파일	1KB
inscount0.cpp	2016-06-30 오후...	C++ Source	4KB
inscount1.cpp	2016-06-30 오후...	C++ Source	4KB
inscount2.cpp	2016-06-30 오후...	C++ Source	5KB

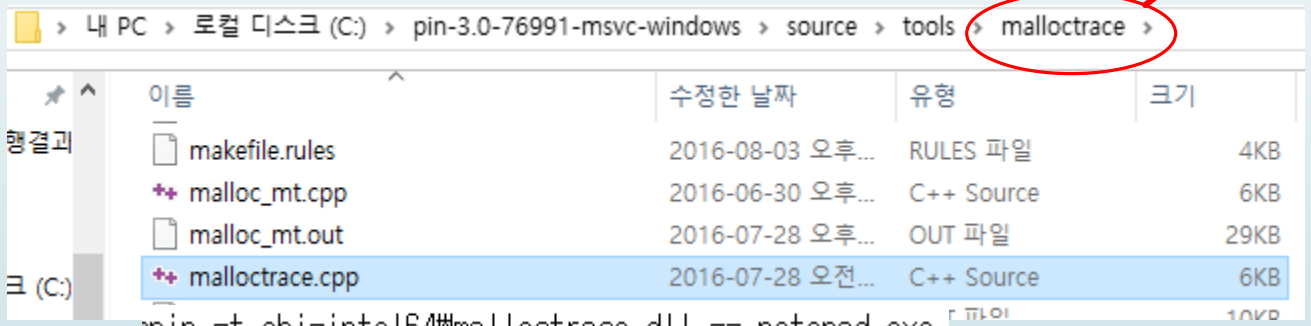
.out 파일 생성

명령어 개수

```
Count 379951017
```

malloctrace

```
>cd C:₩Users₩haha₩Desktop₩공부자료₩pin-3.0-76991-msvc-windows₩source₩tools₩malloctrace
```




```
pin -t obj-intel64₩malloctrace.dll -- notepad.exe
```

실행결과

makefile.rules	2016-08-03 오후...	RULES 파일	4KB
malloc_mt.cpp	2016-06-30 오후...	C++ Source	6KB
malloc_mt.out	2016-07-28 오후...	OUT 파일	29KB
malloctrace.cpp	2016-07-28 오전...	C++ Source	6KB
malloctrace.out	2016-07-28 오후...	OUT 파일	10KB
pin.log	2016-08-16 오전...	텍스트 문서	1KB

```
malloc(0x137c)
  returns 0x12911e61220
malloc(0x220)
  returns 0x12911e64a70
free(0x12911e61220)
malloc(0xa0)
  returns 0x12911e62230
malloc(0x18)
  returns 0x12911e622e0
malloc(0x28)
  returns 0x12911e62300
malloc(0x100)
  returns 0x12911e62330
malloc(0x100)
  returns 0x12911e62330
malloc(0x100)
  returns 0x12911e62440
malloc(0x100)
```

매개변수

Return 값

CreateFileTrace

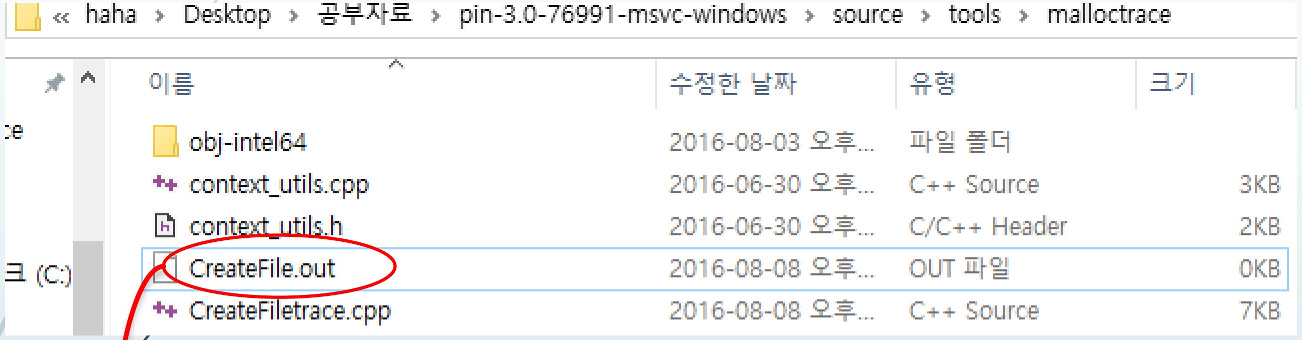
```
>cd C:₩Users₩haha₩Desktop₩공부자료₩pin-3.0-76991-msvc-windows₩source₩tools₩malloctrace
```

« haha › Desktop › 공부자료 › pin-3.0-76991-msvc-windows › source › tools › malloctrace

이름	수정한 날짜	유형	크기
obj-intel64	2016-08-03 오후...	파일 폴더	
context_utils.cpp	2016-06-30 오후...	C++ Source	3KB
context_utils.h	2016-06-30 오후...	C/C++ Header	2KB
CreateFiletrace.cpp	2016-08-03 오후...	C++ Source	7KB
			1KB
		파일	4KB
malloc_mt.cpp	2016-06-30 오후...	C++ Source	6KB
malloc_mt.out	2016-07-28 오후...	OUT 파일	29KB

```
>pin -t obj-intel64₩CreateFiletrace.dll -- notepad.exe
```

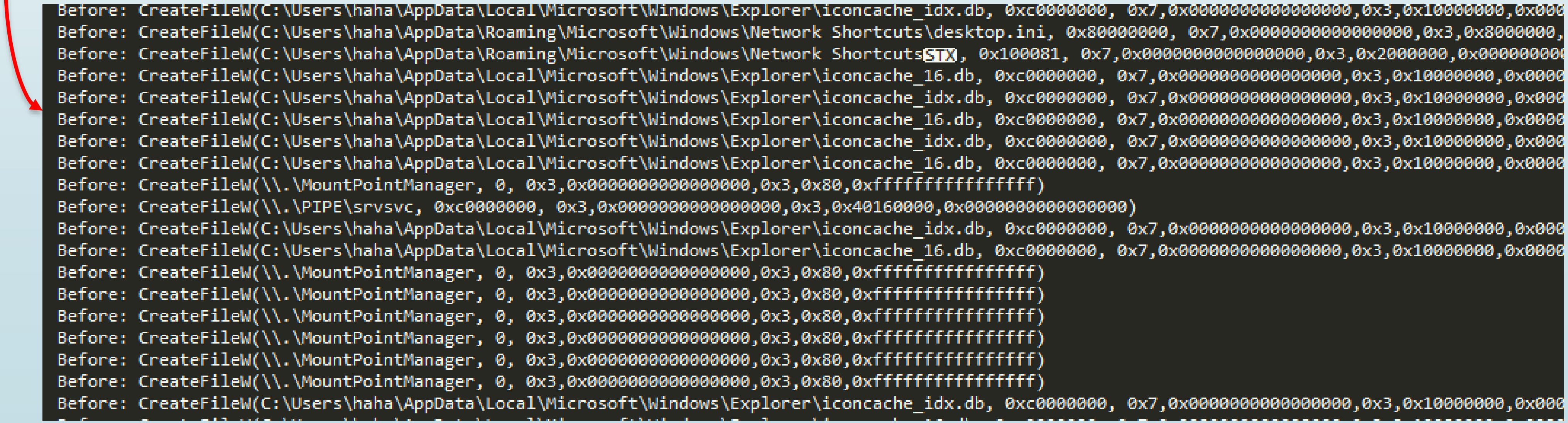
실행결과



« haha › Desktop › 공부자료 › pin-3.0-76991-msvc-windows › source › tools › malloctrace

이름	수정한 날짜	유형	크기
obj-intel64	2016-08-03 오후...	파일 폴더	
context_utils.cpp	2016-06-30 오후...	C++ Source	3KB
context_utils.h	2016-06-30 오후...	C/C++ Header	2KB
CreateFile.out	2016-08-08 오후...	OUT 파일	0KB
CreateFiletrace.cpp	2016-08-08 오후...	C++ Source	7KB

특정 함수 실시간 관찰 가능

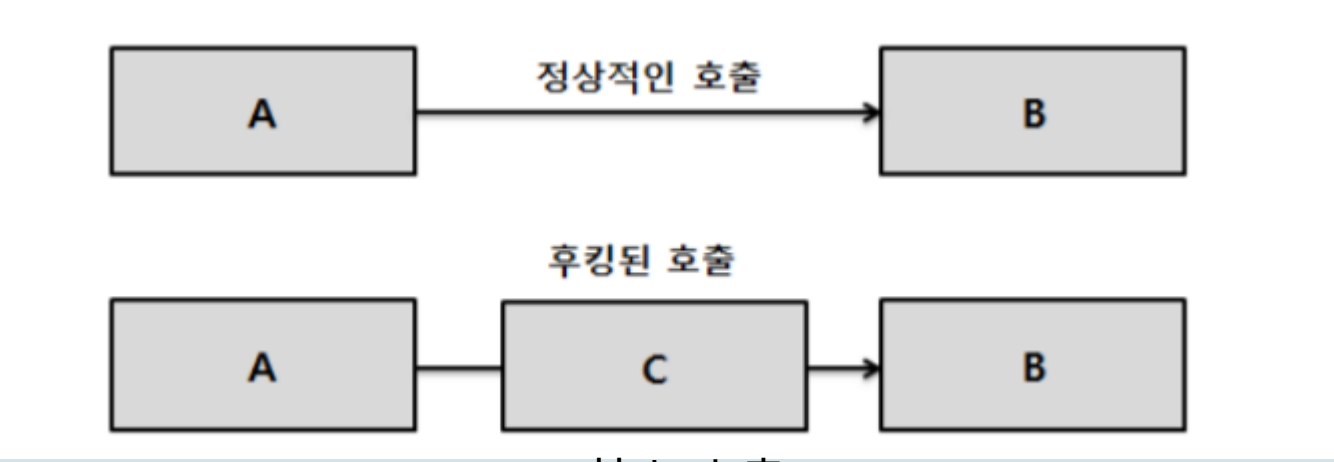


```
Before: CreateFileW(C:\Users\haha\AppData\Local\Microsoft\Windows\Explorer\iconcache_idx.db, 0xc0000000, 0x7,0x0000000000000000,0x3,0x10000000,0x000
Before: CreateFileW(C:\Users\haha\AppData\Roaming\Microsoft\Windows\Network Shortcuts\desktop.ini, 0x80000000, 0x7,0x0000000000000000,0x3,0x8000000,
Before: CreateFileW(C:\Users\haha\AppData\Roaming\Microsoft\Windows\Network ShortcutsSTX, 0x100081, 0x7,0x0000000000000000,0x3,0x2000000,0x00000000
Before: CreateFileW(C:\Users\haha\AppData\Local\Microsoft\Windows\Explorer\iconcache_16.db, 0xc0000000, 0x7,0x0000000000000000,0x3,0x10000000,0x0000
Before: CreateFileW(C:\Users\haha\AppData\Local\Microsoft\Windows\Explorer\iconcache_idx.db, 0xc0000000, 0x7,0x0000000000000000,0x3,0x10000000,0x000
Before: CreateFileW(C:\Users\haha\AppData\Local\Microsoft\Windows\Explorer\iconcache_16.db, 0xc0000000, 0x7,0x0000000000000000,0x3,0x10000000,0x0000
Before: CreateFileW(C:\Users\haha\AppData\Local\Microsoft\Windows\Explorer\iconcache_idx.db, 0xc0000000, 0x7,0x0000000000000000,0x3,0x10000000,0x000
Before: CreateFileW(C:\Users\haha\AppData\Local\Microsoft\Windows\Explorer\iconcache_16.db, 0xc0000000, 0x7,0x0000000000000000,0x3,0x10000000,0x0000
Before: CreateFileW(\\.\MountPointManager, 0, 0x3,0x0000000000000000,0x3,0x80,0xffffffffffffffff)
Before: CreateFileW(\\.\PIPE\srvsvc, 0xc0000000, 0x3,0x0000000000000000,0x3,0x40160000,0x0000000000000000)
Before: CreateFileW(C:\Users\haha\AppData\Local\Microsoft\Windows\Explorer\iconcache_idx.db, 0xc0000000, 0x7,0x0000000000000000,0x3,0x10000000,0x000
Before: CreateFileW(C:\Users\haha\AppData\Local\Microsoft\Windows\Explorer\iconcache_16.db, 0xc0000000, 0x7,0x0000000000000000,0x3,0x10000000,0x0000
Before: CreateFileW(\\.\MountPointManager, 0, 0x3,0x0000000000000000,0x3,0x80,0xffffffffffffffff)
Before: CreateFileW(\\.\MountPointManager, 0, 0x3,0x0000000000000000,0x3,0x80,0xffffffffffffffff)
Before: CreateFileW(\\.\MountPointManager, 0, 0x3,0x0000000000000000,0x3,0x80,0xffffffffffffffff)
Before: CreateFileW(\\.\MountPointManager, 0, 0x3,0x0000000000000000,0x3,0x80,0xffffffffffffffff)
Before: CreateFileW(\\.\MountPointManager, 0, 0x3,0x0000000000000000,0x3,0x80,0xffffffffffffffff)
Before: CreateFileW(\\.\MountPointManager, 0, 0x3,0x0000000000000000,0x3,0x80,0xffffffffffffffff)
Before: CreateFileW(C:\Users\haha\AppData\Local\Microsoft\Windows\Explorer\iconcache_idx.db, 0xc0000000, 0x7,0x0000000000000000,0x3,0x10000000,0x000
```

함수 가로채기 예제

정상 :

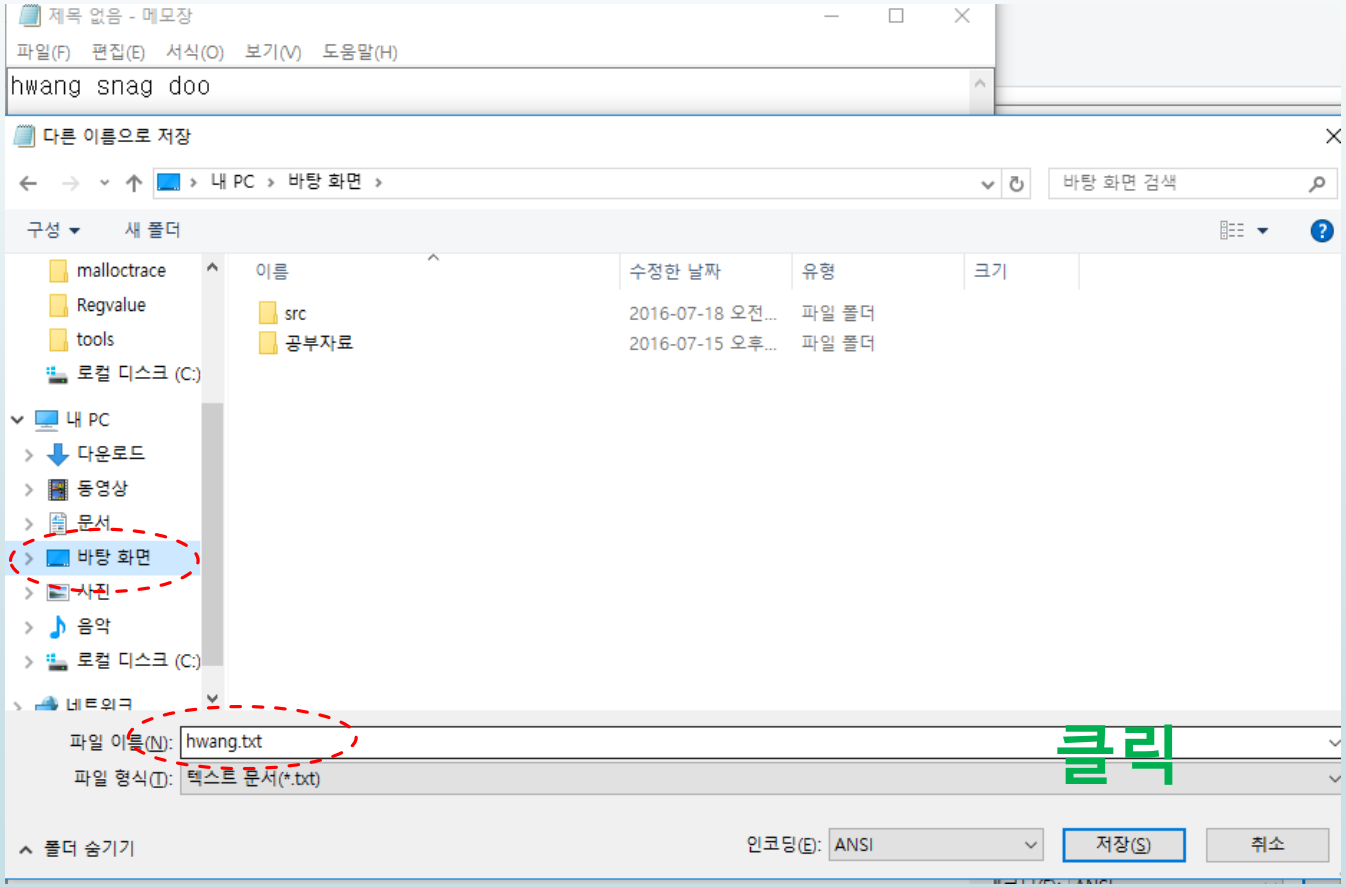
후킹 :

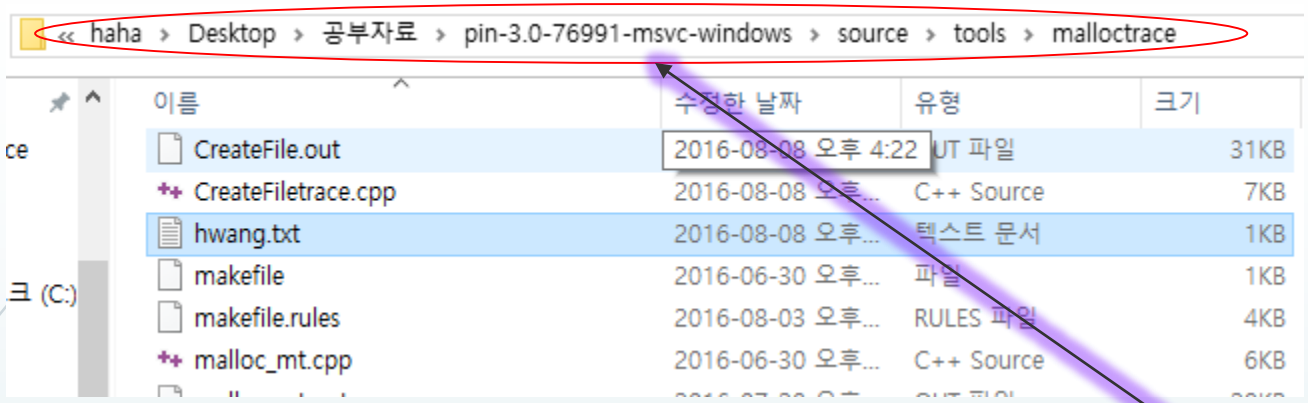


C 함수 호출

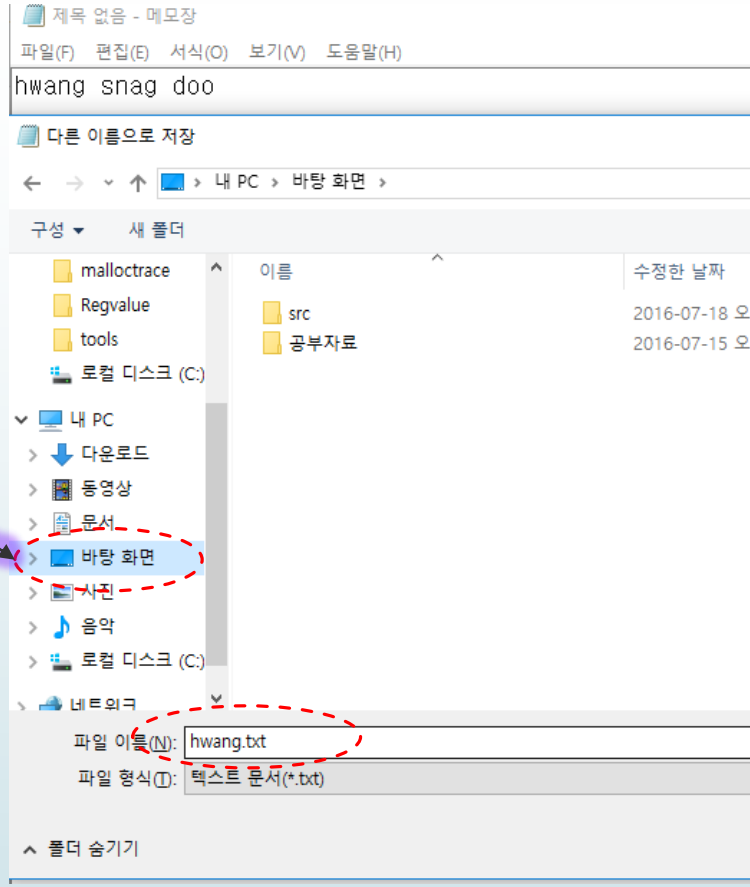
단, C함수는 사용자가 정의

```
>pin -t obj-intel64₩replacesigprobedc.dll -- notepad.exe
```





바탕화면이 아닌 위치 저장



Probe모드 → 함수 삽입 가능

Main 함수

```
int main( INT32 argc, CHAR *argv[] )
{
    // Initialize symbol processing
    //
    PIN_InitSymbols();

    // Initialize pin
    //
    if (PIN_Init(argc, argv)) return Usage();
```

초기화

```
    // Register ImageLoad to be called when an image is loaded
    //
    IMG_AddInstrumentFunction( ImageLoad, 0 );
```

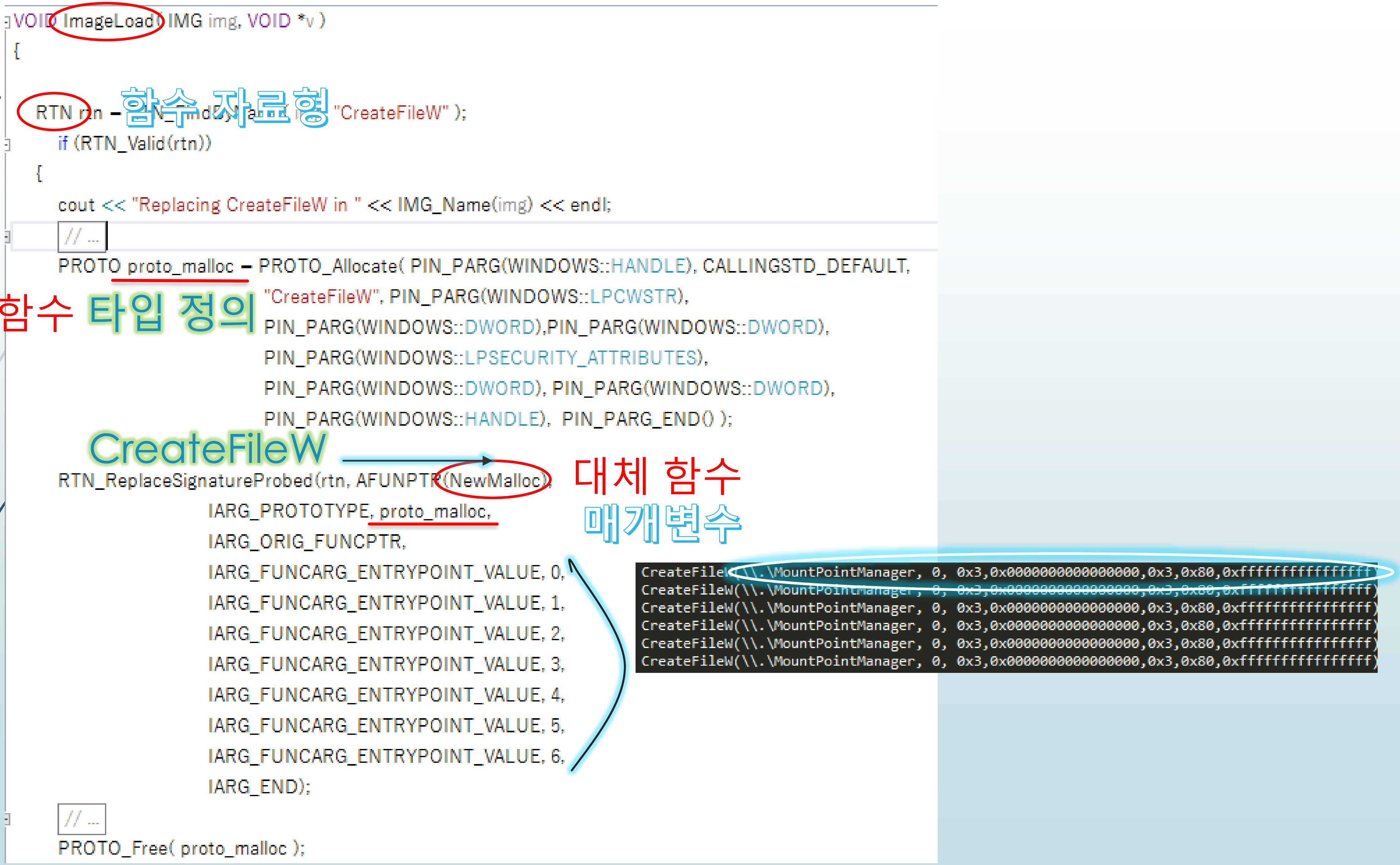
함수 포인터

Code 삽입

```
    // Start the program in probe mode, never returns
    //
    PIN_StartProgramProbed();
```

Pin 시작

```
    return 0;
}
```



```
VOID ImageLoad( IMG img, VOID *v )
{

    RTN rtn = RTN_FindByName( img, "CreateFileW" );
    if (RTN_Valid(rtn))
    {
        cout << "Replacing CreateFileW in " << IMG_Name(img) << endl;
        // ...
        PROTO proto_malloc = PROTO_Allocate( PIN_PARG(WINDOWS::HANDLE), CALLINGSTD_DEFAULT,
                                            "CreateFileW", PIN_PARG(WINDOWS::LPCWSTR),
                                            PIN_PARG(WINDOWS::DWORD),PIN_PARG(WINDOWS::DWORD),
                                            PIN_PARG(WINDOWS::LPSECURITY_ATTRIBUTES),
                                            PIN_PARG(WINDOWS::DWORD), PIN_PARG(WINDOWS::DWORD),
                                            PIN_PARG(WINDOWS::HANDLE),  PIN_PARG_END() );

        RTN_ReplaceSignatureProbed(rtn, AFUNPTR(NewMalloc),
                    IARG_PROTOTYPE, proto_malloc,
                    IARG_ORIG_FUNCPTR,
                    IARG_FUNCARG_ENTRYPOINT_VALUE, 0,
                    IARG_FUNCARG_ENTRYPOINT_VALUE, 1,
                    IARG_FUNCARG_ENTRYPOINT_VALUE, 2,
                    IARG_FUNCARG_ENTRYPOINT_VALUE, 3,
                    IARG_FUNCARG_ENTRYPOINT_VALUE, 4,
                    IARG_FUNCARG_ENTRYPOINT_VALUE, 5,
                    IARG_FUNCARG_ENTRYPOINT_VALUE, 6,
                    IARG_END);
        // ...
        PROTO_Free( proto_malloc );
```

함수 자료형

대체 함수 타입 정의

CreateFileW

대체 함수

매개변수

```
CreateFileW(\\.\MountPointManager, 0, 0x3,0x0000000000000000,0x3,0x80,0xffffffffffffffff
CreateFileW(\\.\MountPointManager, 0, 0x3,0x0000000000000000,0x3,0x80,0xffffffffffffffff
CreateFileW(\\.\MountPointManager, 0, 0x3,0x0000000000000000,0x3,0x80,0xffffffffffffffff
CreateFileW(\\.\MountPointManager, 0, 0x3,0x0000000000000000,0x3,0x80,0xffffffffffffffff
CreateFileW(\\.\MountPointManager, 0, 0x3,0x0000000000000000,0x3,0x80,0xffffffffffffffff
CreateFileW(\\.\MountPointManager, 0, 0x3,0x0000000000000000,0x3,0x80,0xffffffffffffffff
```

대체 함수

```
VOID * NewMalloc(FP_MALLOC orgFuncptr, WINDOWS::LPCWSTR  lpFileName,WINDOWS::DWORD dwDesiredAccess,
 WINDOWS::DWORD dwShareMode, WINDOWS::LPSECURITY_ATTRIBUTES  lpSecurityAttributes,
  WINDOWS::DWORD dwCreationDisposition , WINDOWS::DWORD dwFlagsAndAttributes  ,
  WINDOWS::HANDLE hTemplateFile)
{
    /* ... */
    CHAR* fileName = new CHAR[wcslen(lpFileName)];
    wcstombs(fileName , lpFileName , wcslen(lpFileName));
    cout << "CreateFileW"  << "(" << fileName  << ", " << hex   << dwDesiredAccess << ", " << dwShareMode << ","
        << lpSecurityAttributes << "," << dwCreationDisposition << ","  << dwFlagsAndAttributes << "," << hTemplateFile << ")" << dec << endl;
    VOID * v;
    WINDOWS::LPCWSTR str;
    if (wcsstr(lpFileName, L"hwang.txt") == NULL)
      v = orgFuncptr(lpFileName, dwDesiredAccess, dwShareMode, lpSecurityAttributes,
        dwCreationDisposition, dwFlagsAndAttributes, hTemplateFile);
    else
    {
      str = L"C:¥Users¥haha¥Desktop¥공부자료¥pin-3.0-76991-msvc-windows¥source¥tools¥malloctrace¥hwang.txt";
      v = orgFuncptr(str, dwDesiredAccess, dwShareMode, lpSecurityAttributes,
        dwCreationDisposition, dwFlagsAndAttributes, hTemplateFile);
    }
    return v;
}
```

유니코드 → 멀티바이트 코드

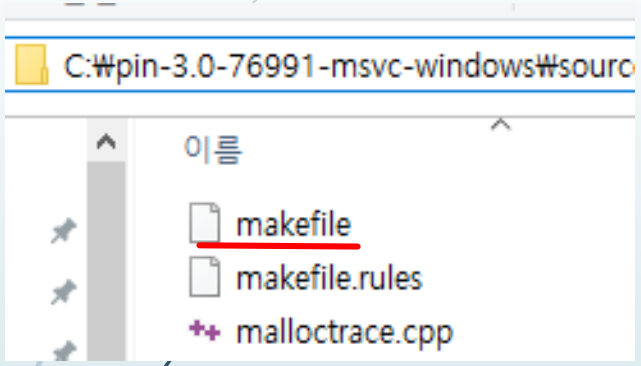
출력

```
CreateFileW(\\.\MountPointManager, 0, 0x3,0x0000000000000000,0x3,0x80,0xffffffffffffffff)
```

If SaveFileName == hwang.txt

저장경로 변경

향후 계획

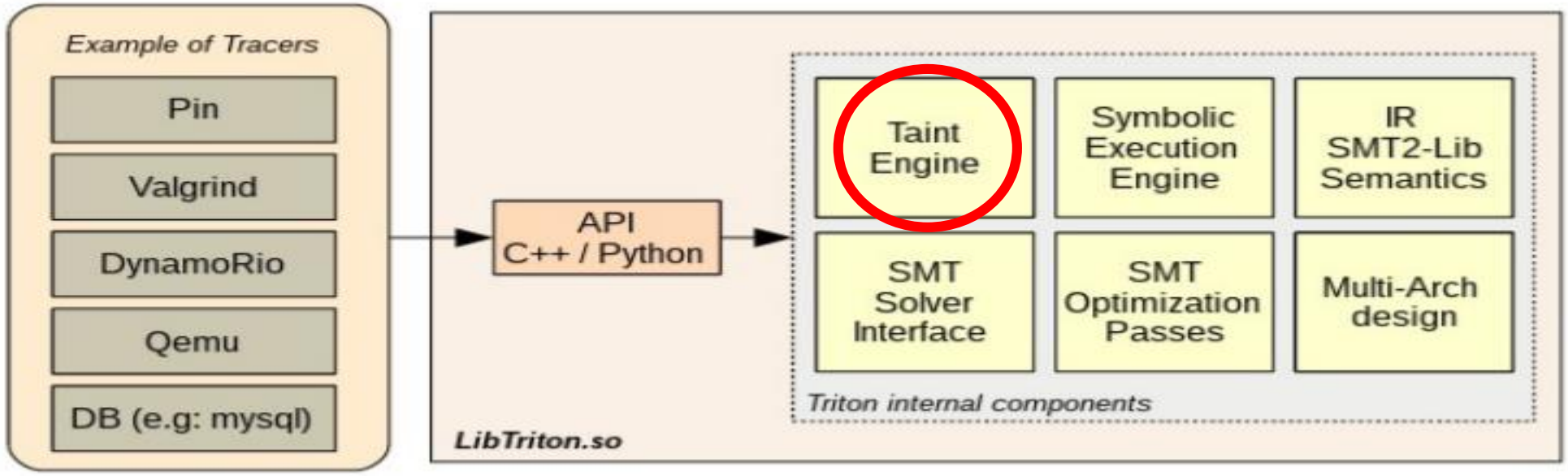


Gcc 컴파일러(**리눅스**) 및 Cygwin **사용 가능**

리눅스 + Visual studio(**windows**)

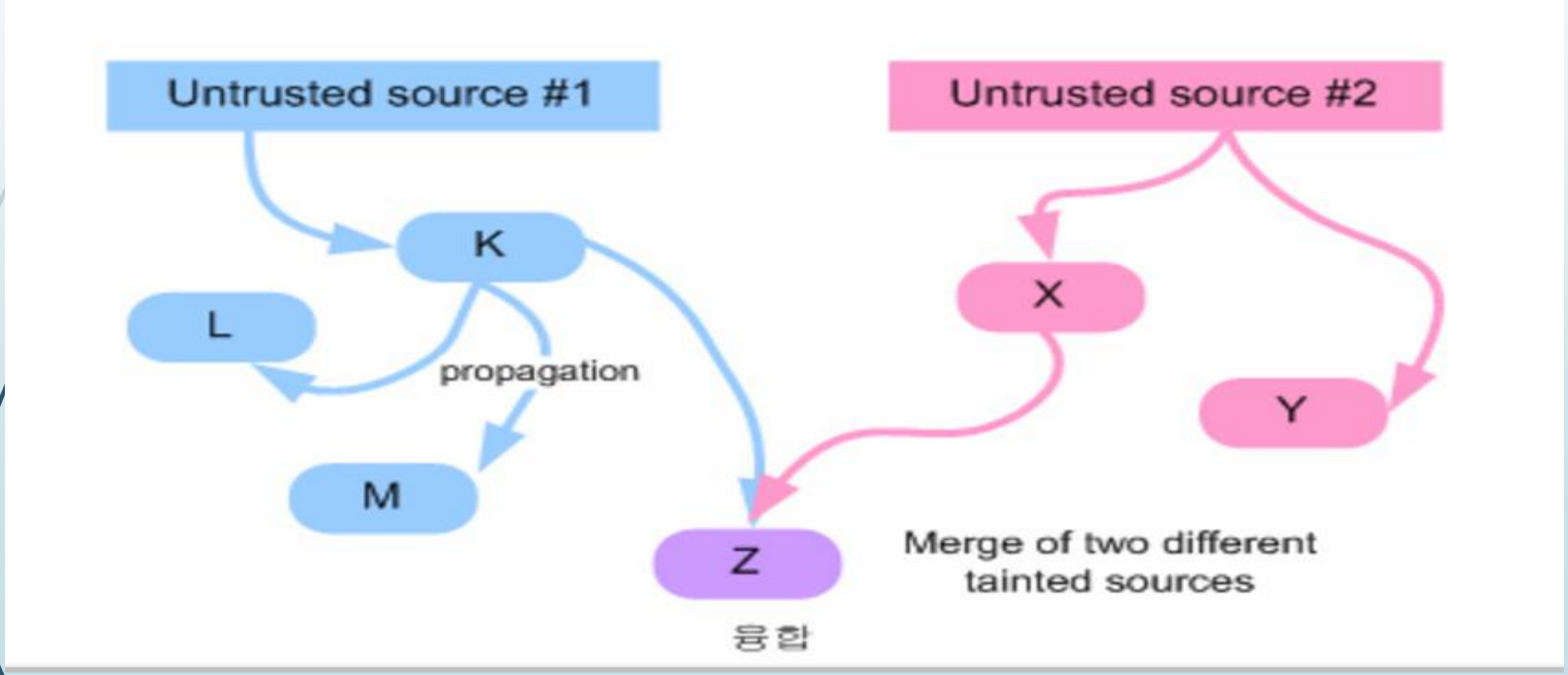
둘다 가능하도록 변환

Taint Analysis



Taint Analysis

➨**외부입력**으로부터 흐름을 파악 !

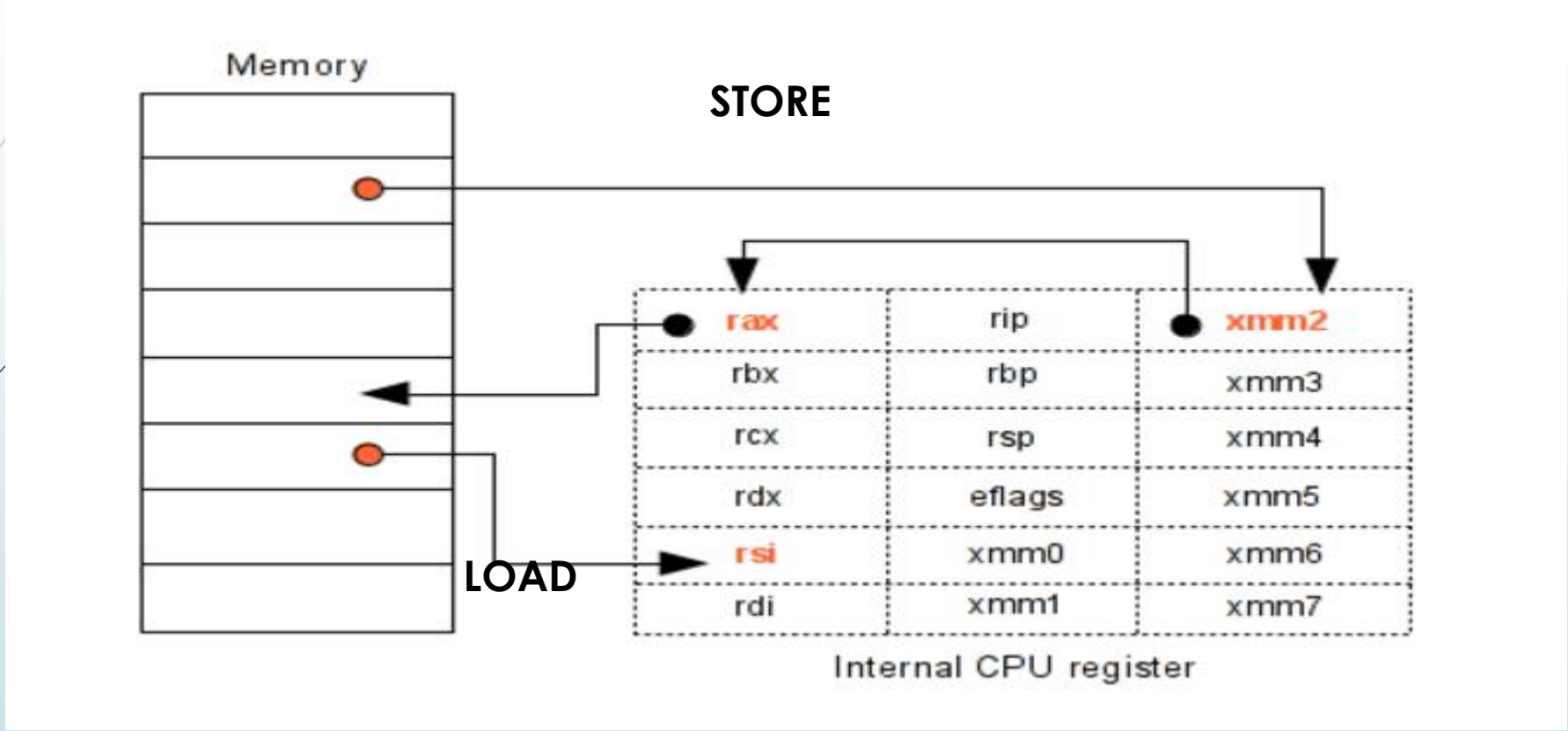


Taint Analysis

```c
void foo(char *buf)
{
  char a;

  a = buf[0];
  a = buf[4];
  a = buf[8];
  a = buf[10];
  buf[5]  = 't';
  buf[10] = 'e';
  buf[20] = 's';
  buf[30] = 't';
}

int main(int ac, char **av)
{
  int fd;
  char *buf;

  if (!(buf = malloc(256)))
    return -1;

  fd = open("./file.txt", O_RDONLY);
  read(fd, buf, 256), close(fd);
  foo(buf);
}
```

```
(angr)hwang@ubuntu:~/pin-3.2-81205-gcc-linux/source/tools/SimpleTai
[TAINT]                    bytes tainted from 0x2337010 to 0x2337110 (
[READ in 2337010]          400620: movzx eax, byte ptr [rax]
[READ in 2337014]          40062a: movzx eax, byte ptr [rax+0x4]
[READ in 2337018]          400635: movzx eax, byte ptr [rax+0x8]
[READ in 233701a]          400640: movzx eax, byte ptr [rax+0xa]
[WRITE in 2337015]         40064f: mov byte ptr [rax], 0x74
[WRITE in 233701a]         40065a: mov byte ptr [rax], 0x65
[WRITE in 2337024]         400665: mov byte ptr [rax], 0x73
[WRITE in 233702e]         400670: mov byte ptr [rax], 0x74
```

Taint Analysis



LOAD 및 **STORE** 명령어를 캐치하여 테인트를 (**Spread**)퍼뜨릴 수 있다

Taint Analysis

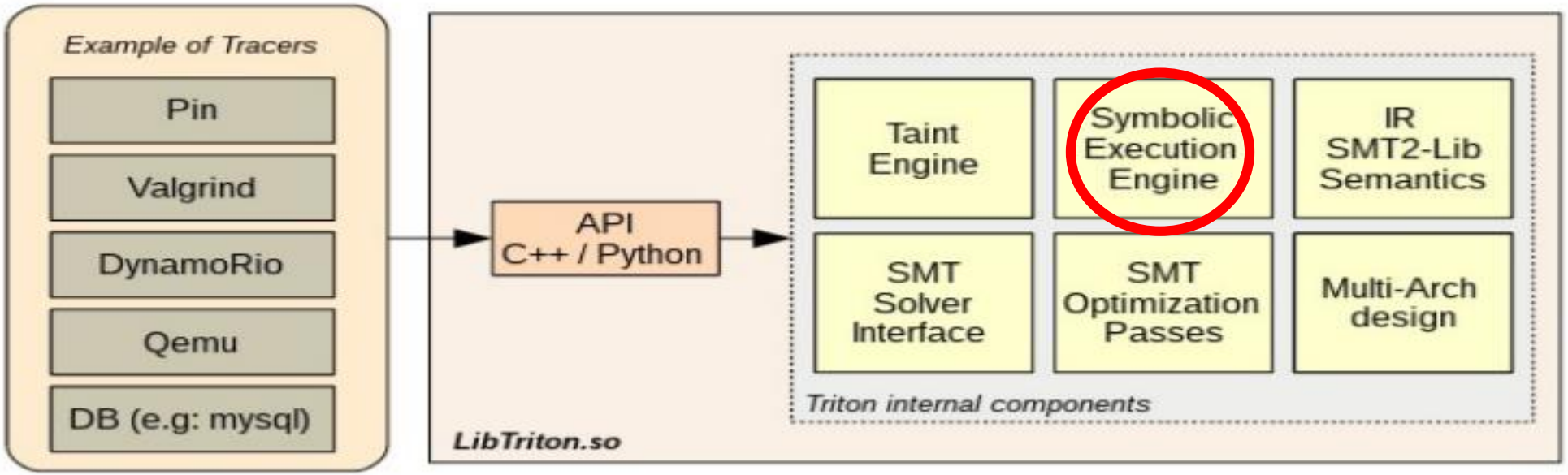
```
                                7ffe937b3f9e is now tainted
[READ in 18b4018]               40065a: movzx eax, byte ptr [rax+0x8]
                                eax is already tainted
[WRITE in 7ffe937b3f9d]         40065e: mov byte ptr [rbp-0x3], al
                                7ffe937b3f9d is now tainted
[READ in 7ffe937b3f9f]          400661: movsx edx, byte ptr [rbp-0x1]
                                edx is now tainted
[READ in 7ffe937b3f9e]          400665: movsx ecx, byte ptr [rbp-0x2]
                                ecx is now tainted
[READ in 7ffe937b3f9d]          400669: movsx eax, byte ptr [rbp-0x3]
                                eax is already tainted
[SPREAD]                        40066d: mov esi, ecx
                                output: esi | input: ecx
                                esi is now tainted
[SPREAD]                        40066f: mov edi, eax
                                output: edi | input: eax
                                edi is now tainted
[WRITE in 7ffe937b3f64]         40061c: mov byte ptr [rbp-0x14], dil
                                7ffe937b3f64 is now tainted
[WRITE in 7ffe937b3f60]         400620: mov byte ptr [rbp-0x18], cl
                                7ffe937b3f60 is now tainted
[WRITE in 7ffe937b3f5c]         400623: mov byte ptr [rbp-0x1c], al
                                7ffe937b3f5c is now tainted
[SPREAD]                        400632: mov eax, 0x0
                                output: eax | input: constant
                                eax is now freed
[SPREAD]                        7f13eb133f45: mov edi, eax
                                output: edi | input: eax
                                edi is now freed
[SPREAD]                        7f13eb14e1eb: mov edx, 0x1
                                output: edx | input: constant
                                edx is now freed
```

테인트 영역이 메모리로 **STRORE 될 때**

테인트 영역을 메모리가 **LOAD 할 때**

SPREAD가 진행된다.

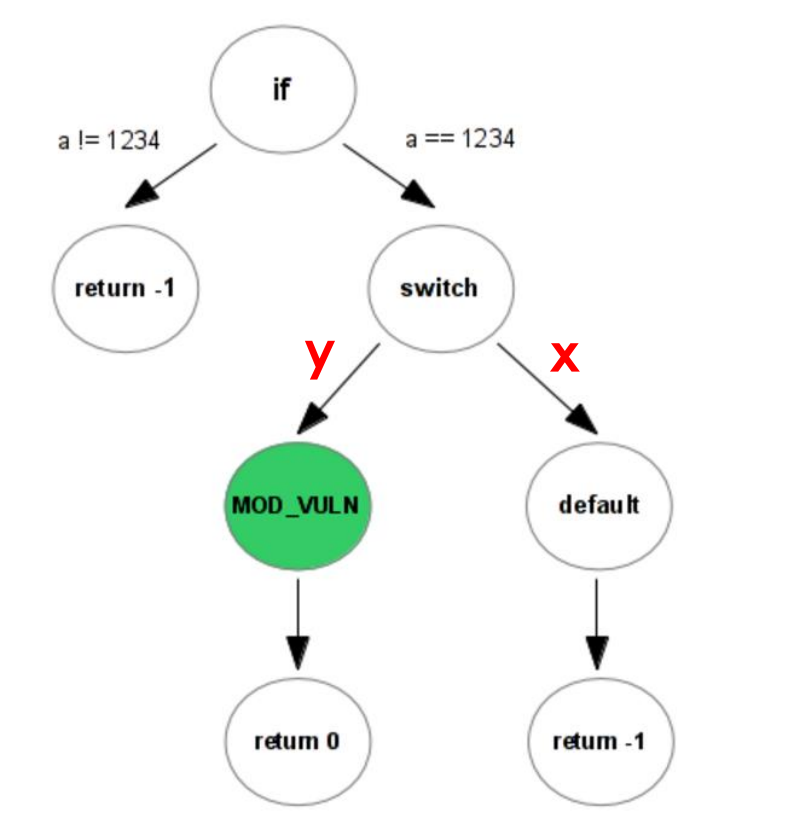
Symbolic Execution Engine



Symbolic Execution Engine

기본적으로 **하나의 input**으로 **하나의 경로**만 실행가능

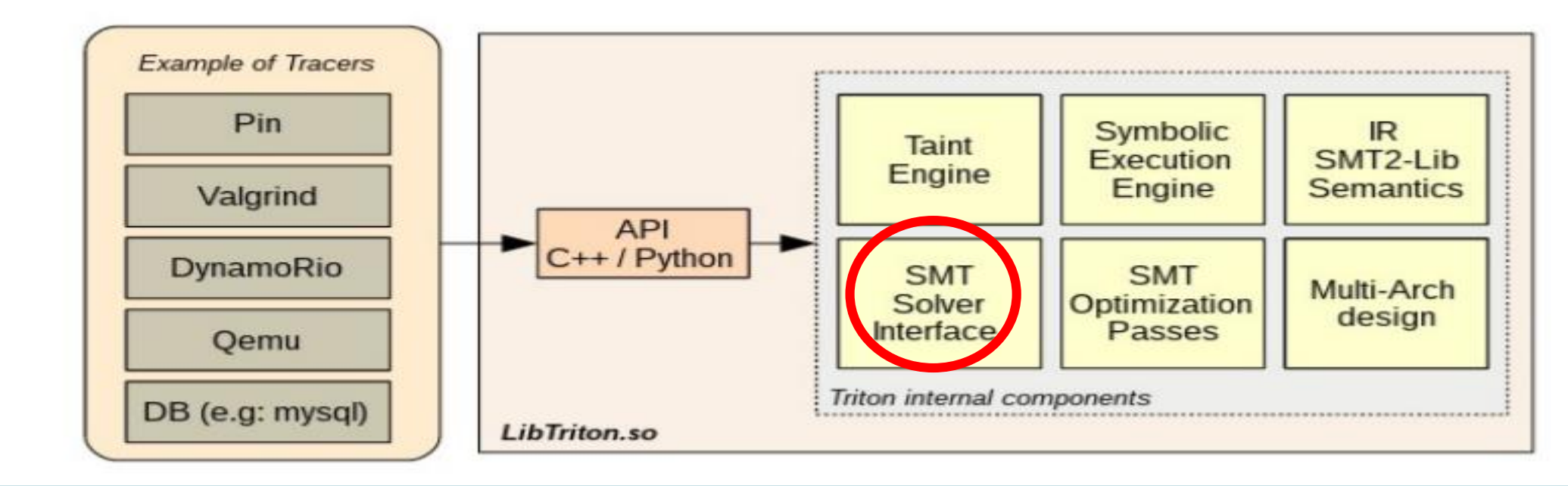
프로그램을 "**여러 갈래로**" 실행**가능** 하도록 만들어줌



(symbolic 기호를 사용)

SMT_Solver

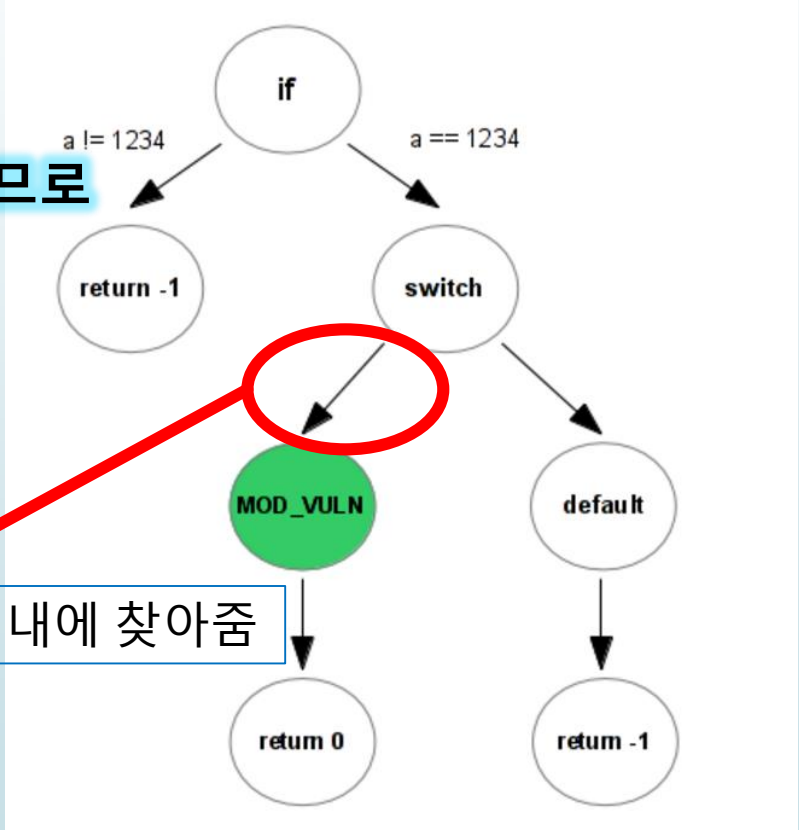
- MS에서 만든 **Z3**가 대표적임
- 자신이 원하는 경로의 해를 빠른 시간 내에 찾아줌



SMT_Solver

```
int foo(int a, char mod){
  if (a == 1234){
    switch (mod){
      case MOD_VULN:
        /* vulnerability here */
        return 0;
      default:
        return -1;
    }
  }
  return -1;
}
```

만족하는 해가 적으므로
경로를 놓치기 쉬움



경로의 해를 빠른 시간 내에 찾아줌

간단한 예제 (시리얼 키)

```
s2 = Int('s[2]')
s3 = Int('s[3]')
s4 = Int('s[4]')
s5 = Int('s[5]')
s6 = Int('s[6]')
s7 = Int('s[7]')
s8 = Int('s[8]')
s9 = Int('s[9]')
s10 = Int('s[10]')
s11 = Int('s[11]')
s12 = Int('s[12]')
s13 = Int('s[13]')
s14 = Int('s[14]')
s15 = Int('s[15]')
s16 = Int('s[16]')
s17 = Int('s[17]')
s18 = Int('s[18]')
s19 = Int('s[19]')
solver = Solver()
solver.add( s0 >= 0)
solver.add( s1 >= 0)
solver.add( s2 >= 0)
solver.add( s3 >= 0)
solver.add( s4 >= 0)
solver.add( s5 >= 0)
solver.add( s6 >= 0)
solver.add( s7 >= 0)
solver.add( s8 >= 0)
solver.add( s9 >= 0)
solver.add( s10 >= 0)
solver.add( s11 < 10)
solver.add( s12 < 10)
solver.add( s13 < 10)
solver.add( s14 < 10)
solver.add( s15 < 10)
solver.add( s16 < 10)
solver.add( s17 < 10)
solver.add( s18 < 10)
solver.add( s19 < 10)
```

```
solver.add( s17 < 10)
solver.add( s18 < 10)
solver.add( s19 < 10)

#add serial checking conditions
solver.add(s15 + s4 == 10)
solver.add(s1 * s18 == 2)
solver.add(s15 / s9 == 1)
solver.add(s17 - s0 == 4)
solver.add(s5 - s17 == -1)
solver.add(s15 - s1 == 5)
solver.add(s1 * s10 == 18)
solver.add(s8 + s13 == 14)
solver.add(s18 * s8 == 5)
solver.add(s4 * s11 == 0)
solver.add(s8 + s9 == 12 )
solver.add(s12 - s19 == 1)
solver.add(s9 % s17 == 7)
solver.add(s14 * s16 == 40)

solver.add(s7 - s4 == 1)

solver.add(s6 + s0 == 6)
solver.add(s2 - s16 == 0)

solver.add(s4 - s6 == 1)
solver.add(s0 % s5 == 4)
solver.add(s2 - s16 == 0)
solver.add(s5 * s11 == 0)

# s3 can't be 0 because of division by zero
solver.add(s3 != 0)
print('solving')
print(solver.check())
print(solver.model())
```

실행결과

```
:₩Users₩Hwang₩Desktop₩z3₩z3-4.3.2-x64-win₩bin>python 111.py
olving
at
s[8] = 5,
s[4] = 3,
s[19] = 8,
s[17] = 8,
s[16] = 8,
s[2] = 8,
s[9] = 7,
s[1] = 2,
s[3] = 1,
s[15] = 7,
s[11] = 0,
s[10] = 9,
s[12] = 9,
s[18] = 1,
s[0] = 4,
s[14] = 5,
s[7] = 4,
s[6] = 2,
s[5] = 7,
s[13] = 9]
```

```
080484E4
080484E4
080484E4 ; Attributes: bp-based frame
080484E4
080484E4 ; int __cdecl main(int argc, const char **argv, const char **envp)
080484E4 public main
080484E4 main proc near
080484E4
080484E4 argc= dword ptr  8
080484E4 argv= dword ptr  0Ch
080484E4 envp= dword ptr  10h
080484E4
080484E4 push    ebp
080484E5 mov     ebp, esp
080484E7 and     esp, 0FFFFFFF0h
080484EA sub     esp, 30h
080484ED mov     eax, large gs:14h
080484F3 mov     [esp+2Ch], eax
080484F7 xor     eax, eax
```

```
080484F9
080484F9 loc_80484F9:                    ; "Enter password: "
080484F9 mov     eax, offset format
080484FE mov     [esp], eax              ; format
08048501 call    _printf
08048506 mov     eax, offset aS          ; "%s"
0804850B lea     edx, [esp+13h]
0804850F mov     [esp+4], edx
08048513 mov     [esp], eax
08048516 call    ___isoc99_scanf
0804851B lea     eax, [esp+13h]
0804851F mov     [esp+4], eax            ; s2
08048523 mov     dword ptr [esp], offset pass_1685 ; "g00dJ0B!"
0804852A call    _strcmp
0804852F test    eax, eax
08048531 jnz     short loc_8048554
```

0848533

```
08048533 mov     dword ptr [esp], offset s ; "Congrats!"
0804853A call    _puts
0804853F nop
08048540 mov     eax, 0
08048545 mov     edx, [esp+2Ch]
08048549 xor     edx, large gs:14h
08048550 jz      short locret_8048567
```

08485354

```
08048554
08048554 loc_8048554:                    ; "Wrong!"
08048554 mov     dword ptr [esp], offset aWrong
0804855B call    _puts
08048560 jmp     short loc_80484F9
```

```
08048552 jmp     short loc_8048562
```

```
08048567
08048567 locret_8048567:
08048567 leave
08048568 retn
08048568 main endp
08048568
```

```
08048562
08048562 loc_8048562:
08048562 call    ___stack_chk_fail
```

```
import angr
def main():
    proj = angr.Project('./babyre',  load_options={'auto_load_libs': False})
    path_group = proj.factory.path_group(threads=4)

    path_group.explore(find = 0x4028C7, avoid = 0x4028c9)

    print path_group.found[0].state.posix.dumps(0)
    print path_group.found[0].state.posix.dumps(1)
main()
```

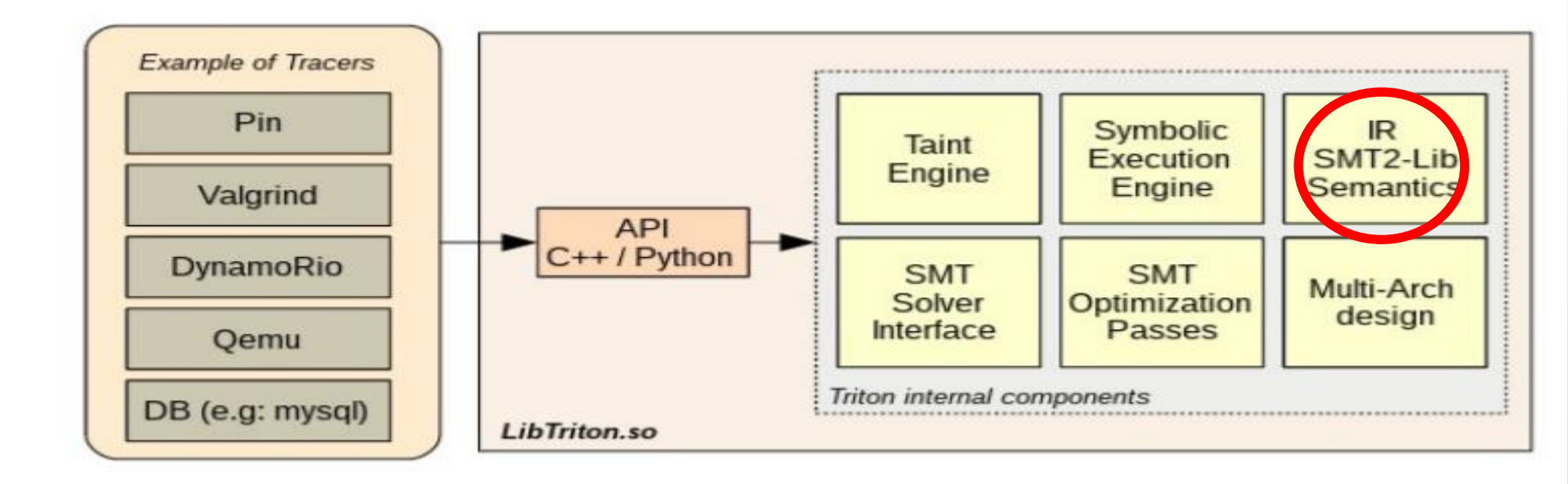
- Find 주소를 찾아가고
- Avoid 주소를 피하는
- 해를 구해다준다.

실행결과

```
(angr)hwang@ubuntu:~$ clear
(angr)hwang@ubuntu:~$ cd angr/
(angr)hwang@ubuntu:~/angr$ ls
=~      amadhj.py       babyre      babyre_ver2.py  z3_p.py~  z4.py    z5.py
amadhj  amadhj_hook.py  babyre.py   z3_p.py         z4        z4.py~   z5.py~
(angr)hwang@ubuntu:~/angr$ python babyre
babyre.py       babyre_ver2.py
(angr)hwang@ubuntu:~/angr$ python babyre
babyre.py       babyre_ver2.py
(angr)hwang@ubuntu:~/angr$ python babyre.py
WARNING | 2017-03-19 17:21:46,992 | simuvex.plugins.symbolic_memory | Concretizi
ng symbolic length. Much sad; think about implementing.
WARNING | 2017-03-19 17:21:51,322 | simuvex.plugins.symbolic_memory | Concretizi
ng symbolic length. Much sad; think about implementing.
WARNING | 2017-03-19 17:21:55,854 | simuvex.plugins.symbolic_memory | Concretizi
ng symbolic length. Much sad; think about implementing.
WARNING | 2017-03-19 17:22:03,661 | simuvex.plugins.symbolic_memory | Concretizi
ng symbolic length. Much sad; think about implementing.
WARNING | 2017-03-19 17:22:17,639 | simuvex.plugins.symbolic_memory | Concretizi
ng symbolic length. Much sad; think about implementing.
WARNING | 2017-03-19 17:22:36,090 | simuvex.plugins.symbolic_memory | Concretizi
ng symbolic length. Much sad; think about implementing.
WARNING | 2017-03-19 17:23:01,678 | simuvex.plugins.symbolic_memory | Concretizi
ng symbolic length. Much sad; think about implementing.
WARNING | 2017-03-19 17:23:38,811 | simuvex.plugins.symbolic_memory | Concretizi
ng symbolic length. Much sad; think about implementing.
WARNING | 2017-03-19 17:24:28,012 | simuvex.plugins.symbolic_memory | Concretizi
ng symbolic length. Much sad; think about implementing.
WARNING | 2017-03-19 17:25:45,744 | simuvex.plugins.symbolic_memory | Concretizi
ng symbolic length. Much sad; think about implementing.
WARNING | 2017-03-19 17:27:06,553 | simuvex.plugins.symbolic_memory | Concretizi
ng symbolic length. Much sad; think about implementing.
  WARNING | 2017-03-19 17:29:22,018 | simuvex.plugins.symbolic_memory | Concreti
zing symbolic length. Much sad; think about implementing.
  WARNING | 2017-03-19 17:32:12,605 | simuvex.plugins.symbolic_memory | Concreti
zing symbolic length. Much sad; think about implementing.
+000000077+000000097+000000116+000000104+000000032+000000105+000000115+000000032
+000000104+000000097+000000114+000000100+000000033

Var[0]: Var[1]: Var[2]: Var[3]: Var[4]: Var[5]: Var[6]: Var[7]: Var[8]: Var[9]:
Var[10]: Var[11]: Var[12]: The flag is: Math is hard!
```

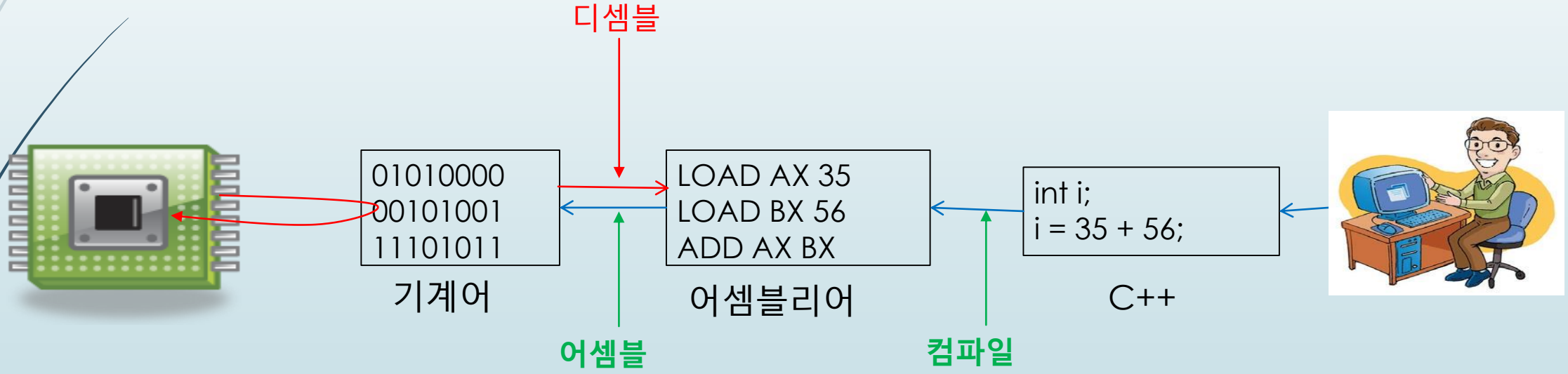
비밀번호 : Math is hard!

Intermediate Representation (IR)



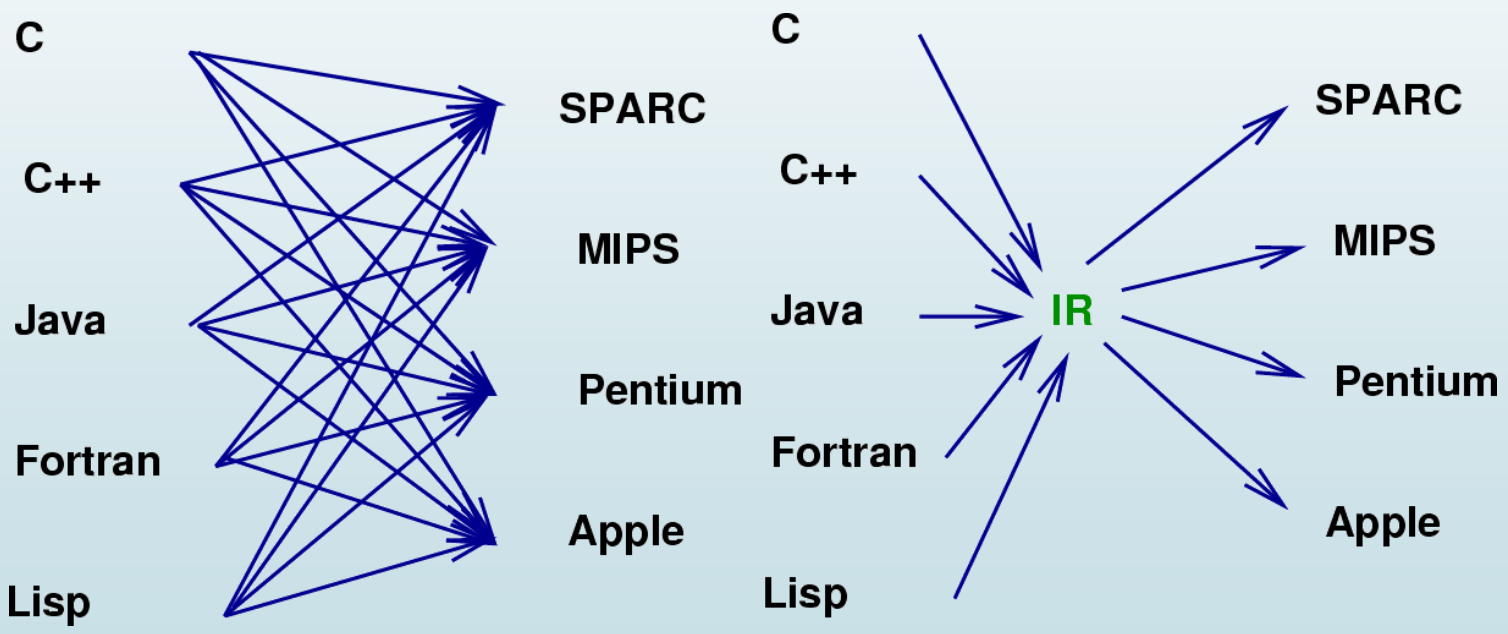
Intermediate Representation (IR)

각 CPU 마다 고유의 기계어와 어셈블리어를 가지고 있다.



Intermediate Representation (IR)

다양한 어셈블리어는 취약점 분석하는데 있어서 **장애물**이 된다.



Intermediate Representation (IR)

```
t3 = GET:I32(0)      # get %eax, a 32-bit integer (t3 = eax)
t2 = GET:I32(12)     # get %ebx, a 32-bit integer (t2 = ebx)
t1 = Add32(t3,t2)    # eger (t2 = ebx)
PUT(0) = t1          put %eax (eax = t1)
```

설명 사이트

1. http://sanguine.leaveret.kr/110 - (입문자용 설명 + 예제)
2. https://software.intel.com/sites/landingpage/pintool/docs/67254/Pin/html/index.html#WINDOWS_TOOLS (예제)
3. https://software.intel.com/sites/landingpage/pintool/docs/67254/Pin/html/group__API__REF.html (API설명서)


Q&A