2nd CodeEngn Seminar

정적 링크된 ELF 파일에서의 외부 심볼 정보 복구 기법

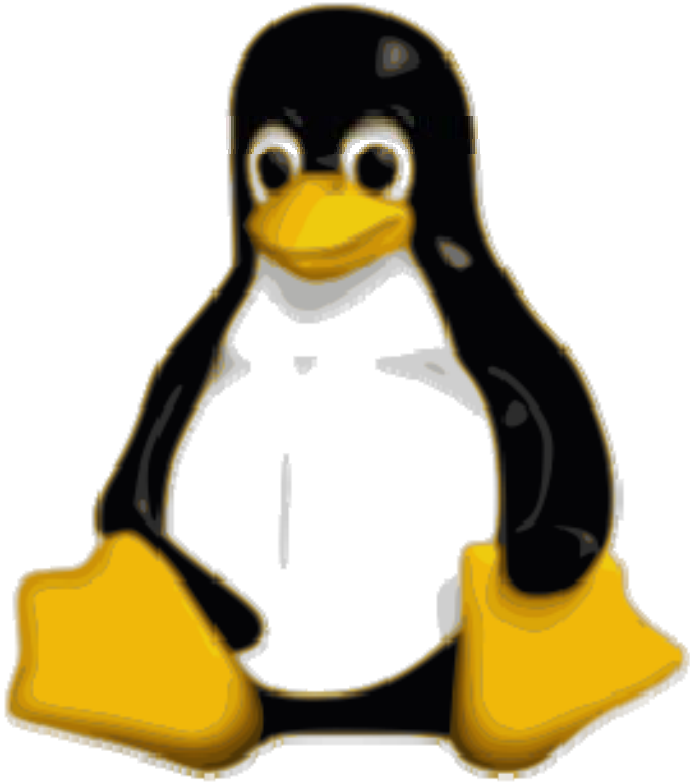
태 인 규 (graylynx@hackken.org)

CodeEngn
www.CodeEngn.com

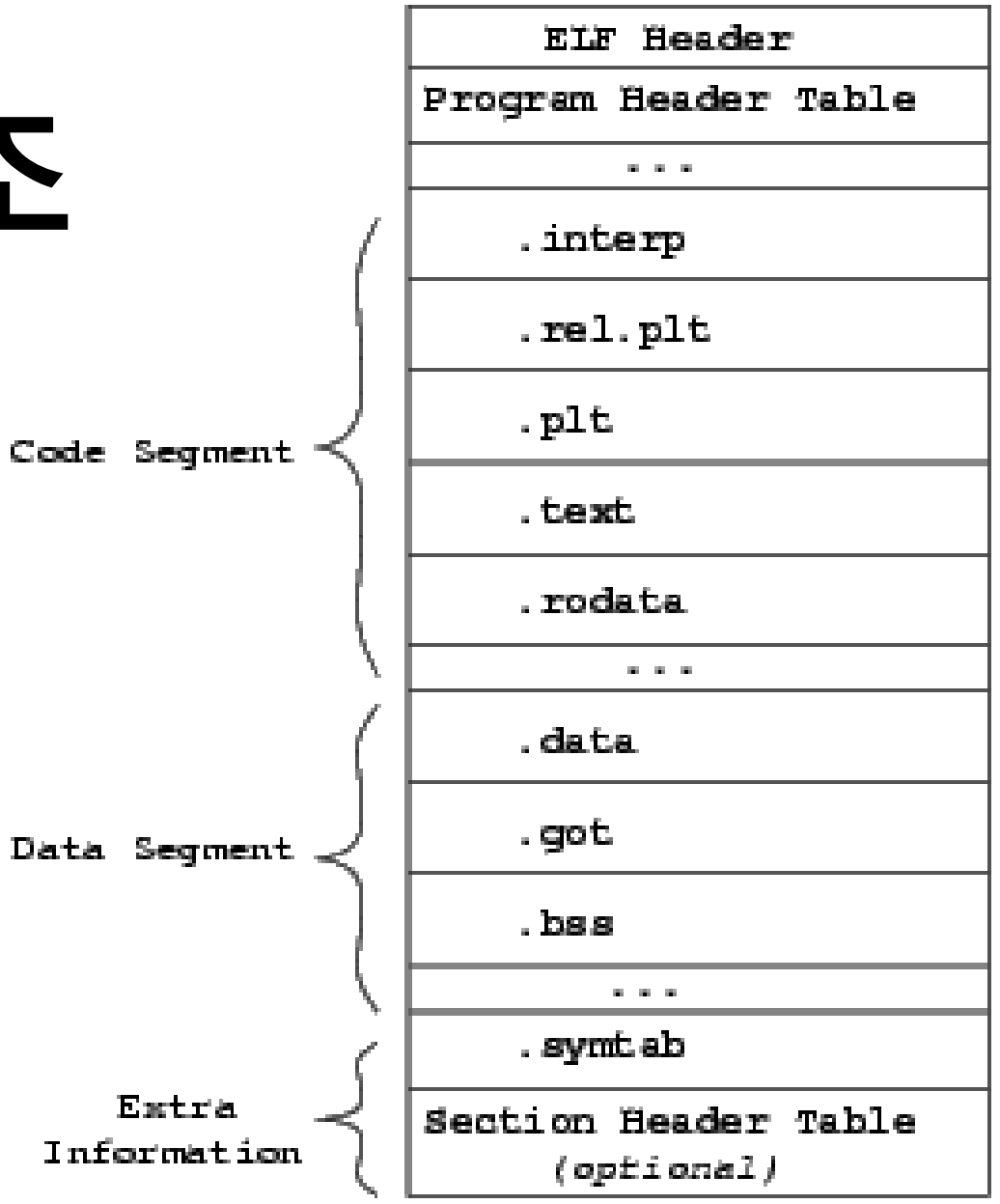
Executable
Linkable
Format

실행 가능한
링크 가능한
형식

공유라이브러리와
실행파일을
위한 기본 형식



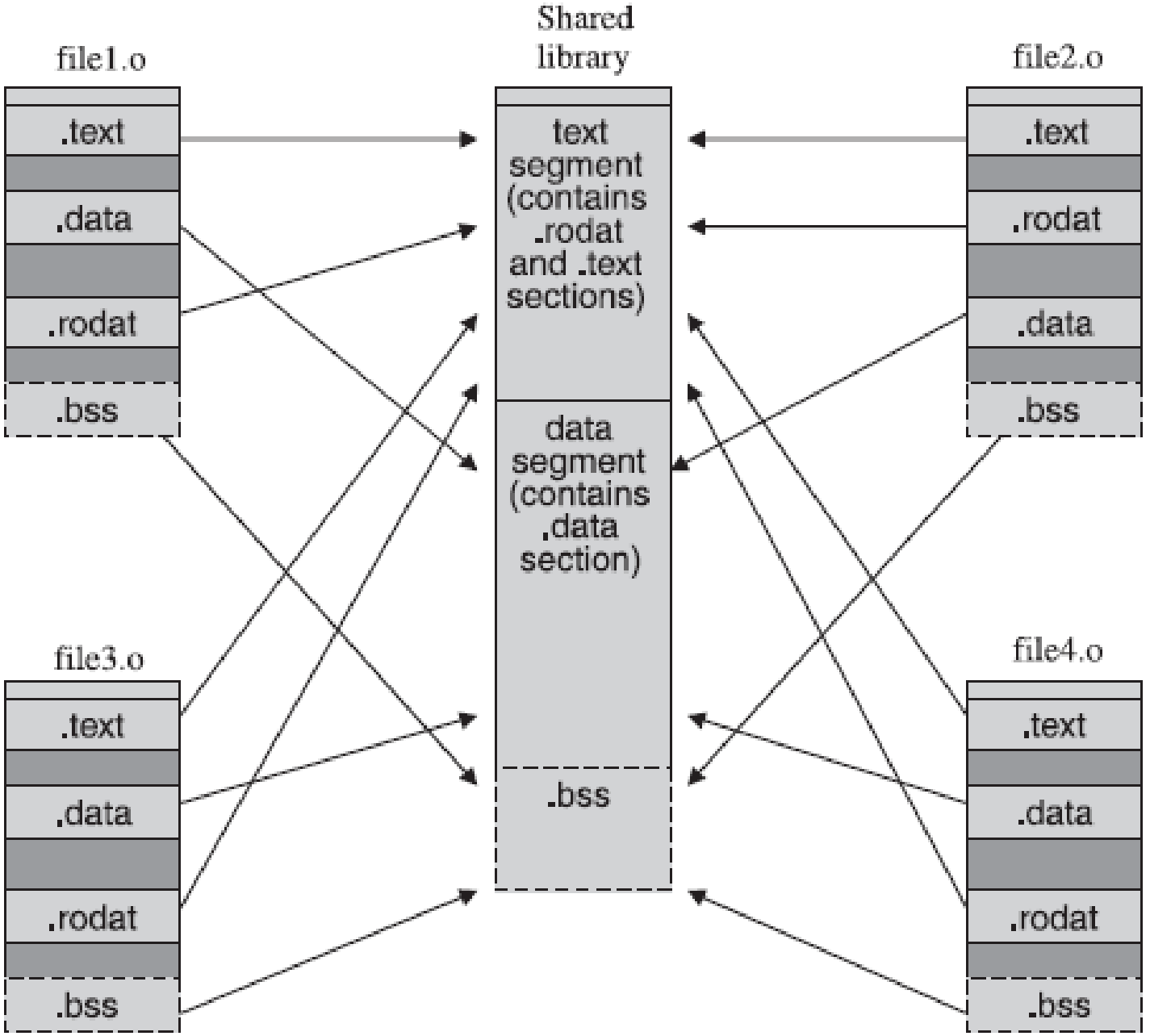
ELF 구조


ELF Header
Program Header Table
...
.interp
.rel.plt
.plt
.text
.rodata
...
.data
.got
.bss
...
.symtab
Section Header Table
(optional)
Code Segment
Data Segment
Extra Information

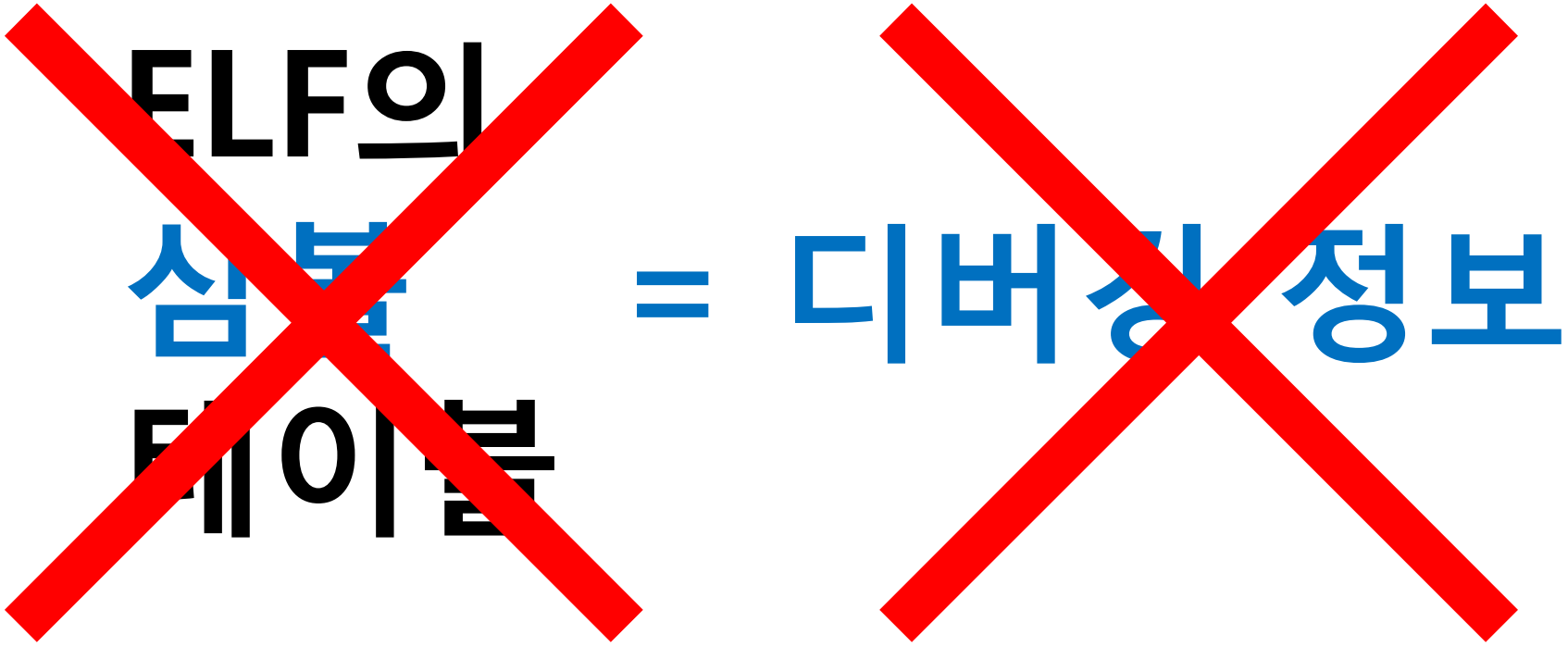
정적 링킹

VS

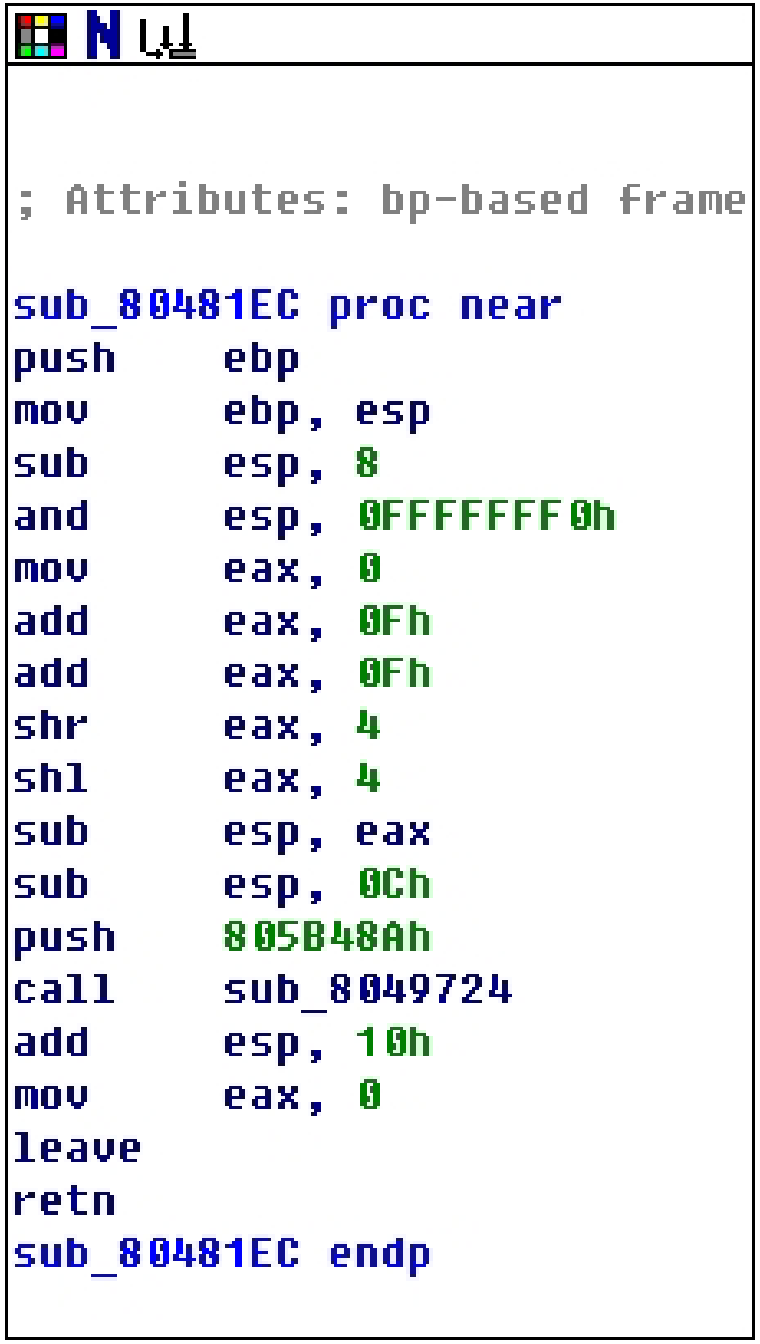
동적 링킹


Shared library
file1.o
.text
.data
.rodat
.bss
text segment (contains .rodat and .text sections)
data segment (contains .data section)
.bss
file2.o
.text
.rodat
.data
.bss
file3.o
.text
.data
.rodat
.bss
file4.o
.text
.data
.rodat
.bss


섹시
심볼


ELF의
심볼
테이블
=
디버깅 정보

QUIZ



```
; Attributes: bp-based frame

sub_80481EC proc near
push    ebp
mov     ebp, esp
sub     esp, 8
and     esp, 0FFFFFFF0h
mov     eax, 0
add     eax, 0Fh
add     eax, 0Fh
shr     eax, 4
shl     eax, 4
sub     esp, eax
sub     esp, 0Ch
push    805B48Ah
call    sub_8049724
add     esp, 10h
mov     eax, 0
leave
retn
sub_80481EC endp
```


WHY?


Stripped!


재개발이 가능할까?


실험

실행 파일에 복사된 공유 라이브러리 코드와 원본 공유 라이브러리 코드의 비교

```
/* example.c */
#include <stdio.h>
int main()
{
      printf("Hello, ELF!₩n");
      return 0;
}
```

비교대상 1
: 정적 링크, 심볼 테이블이 삭제된 ELF 실행 파일

```
$> gcc example.c -o example -static
$> cp example example_stripped
$> strip -S exmaple_stripped
```

비교대상 2
: 정적 공유라이브러리 내 목적 파일

```
$> cp /usr/lib/libc.a /tmp
$> ar /tmp/libc.a
```

비교대상 1 : 실행 파일에 복사된 공유 라이브러리 코드

```
00001720                        55 89 E5 8D  45 0C 83 EC 0C 50 FF 75
00001730  08 FF 35 3C ED 05 08 E8  18 3C 00 00 C9 C3
```

비교대상 2 : 원본 공유 라이브러리 코드

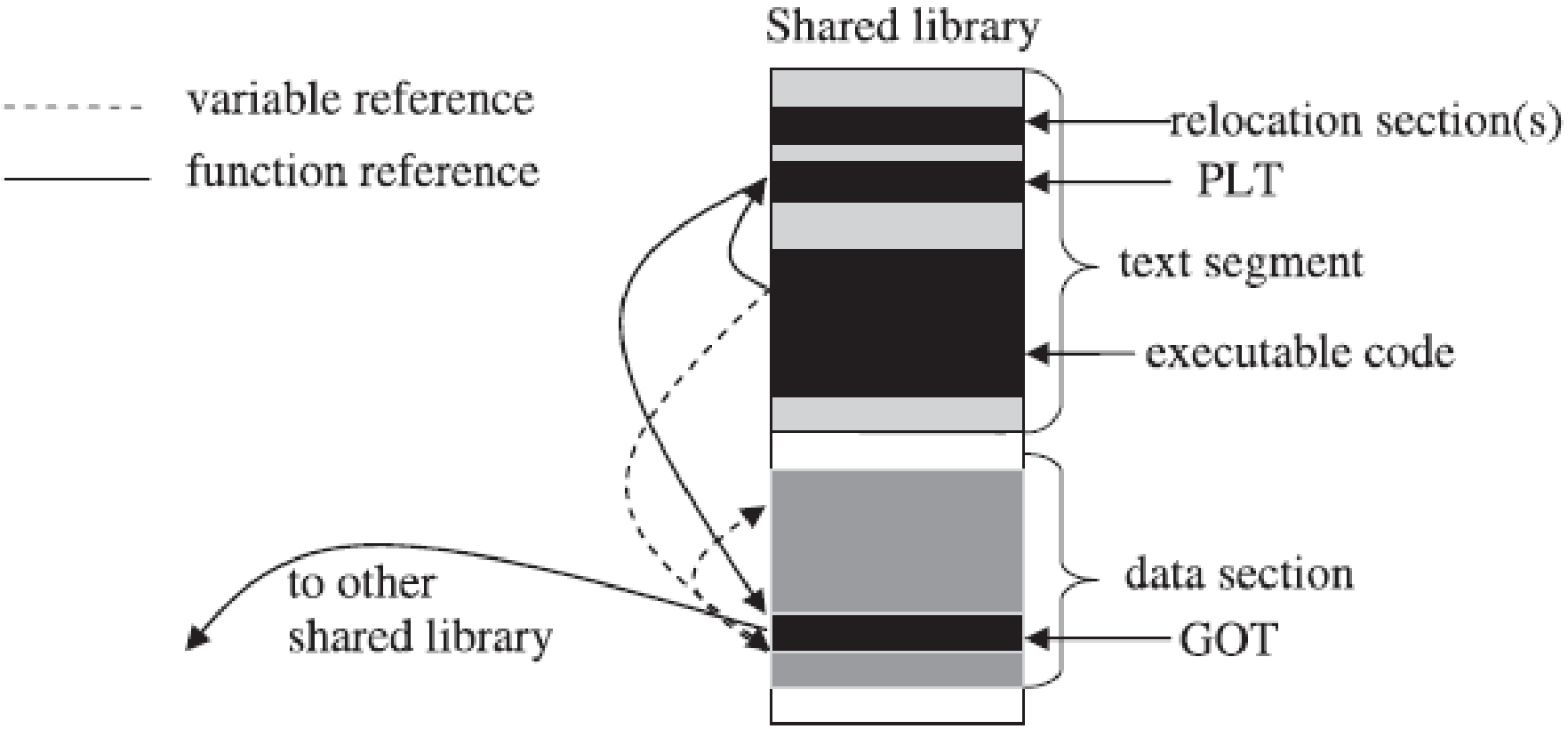
```
00000030                        55 89 E5 8D  45 0C 83 EC 0C 50 FF 75
00000040  08 FF 35 00 00 00 00 E8  FC FF FF FF C9 C3
```

```
0x8049724 <printf>:    push   %ebp
0x8049725 <printf+1>:  mov    %esp,%ebp
0x8049727 <printf+3>:  lea    0xc(%ebp),%eax
0x804972a <printf+6>:  sub    $0xc,%esp
0x804972d <printf+9>:  push   %eax
0x804972e <printf+10>: pushl  0x8(%ebp)
0x8049731 <printf+13>: pushl  0x805ed3c
0x8049737 <printf+19>: call   0x804d354 <vfprintf>
0x804973c <printf+24>: leave
0x804973d <printf+25>: ret
```

실행 파일에 복사된 printf() 함수 기계어 코드


재배치

재배치


Shared library
variable reference
function reference
relocation section(s)
PLT
text segment
executable code
data section
GOT
to other
shared library

재배치 테이블 조회

```
$> readelf -r printf.o
```

```
 Offset     Info    Type            Sym.Value  Sym. Name
0000000f   00000701 R_386_32        00000000   __stdoutp
00000014   00000802 R_386_PC32      00000000   vfprintf
```

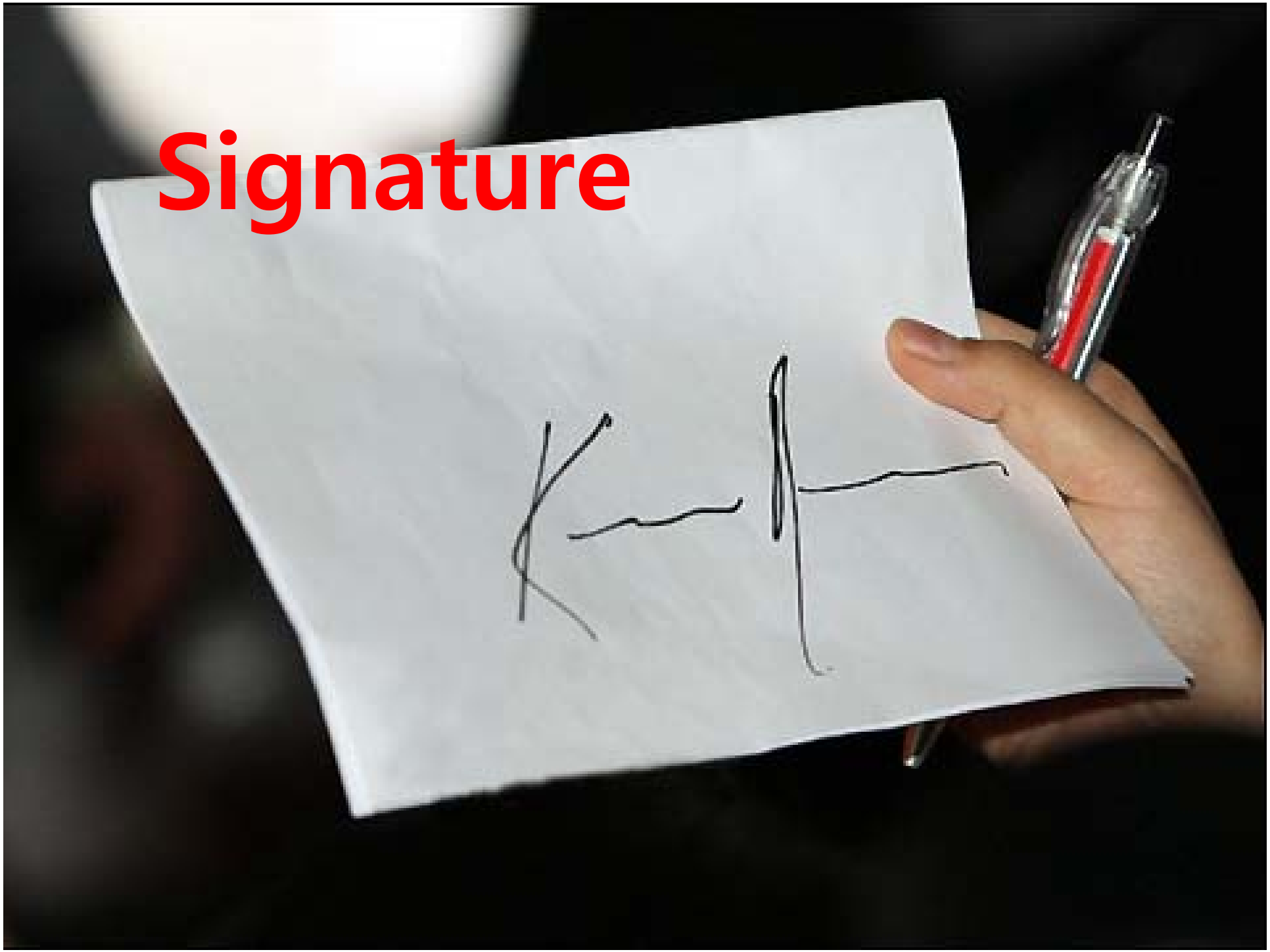
WHAT THE ??!!

비교대상 1

```
00001720                        55 89 E5 8D  45 0C 83 EC 0C 50 FF 75
00001730  08 FF 35 3C ED 05 08 E8  18 3C 00 00 C9 C3
```

비교대상 2

```
00000030                        55 89 E5 8D  45 0C 83 EC 0C 50 FF 75
00000040  08 FF 35 00 00 00 00 E8  FC FF FF FF C9 C3
```


Signature


꿈은 ★ 이루어진다

grayResolve

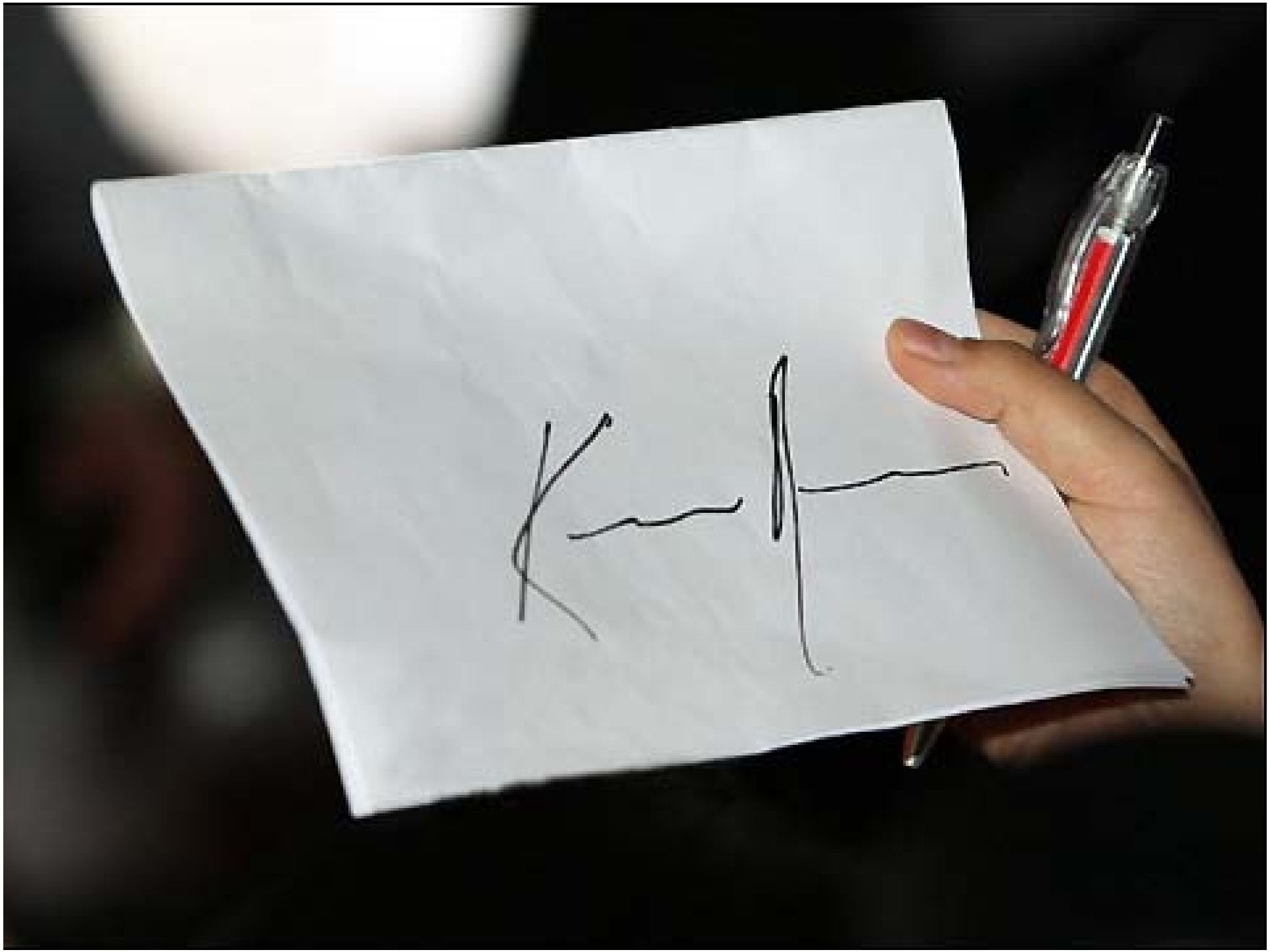
- ✓ IDA Pro Plug-In
- ✓ C / C++ / STL
- ✓ Archive Parser
- ✓ ELF Analyzer
- ✓ Signature Searching
- ✓ Name Resolving

Demo

Q & A

QUIZ

본 발표에서 사용된 삽화 중 사인하는 장면의 주인공은 누구인가?



2nd CodeEngn Seminar

http://blog.hackken.org/grayTools

graylynx@hackken.org

CodeEngn

www.CodeEngn.com