# PLT GOT

- 소스 파일 -> 실행파일

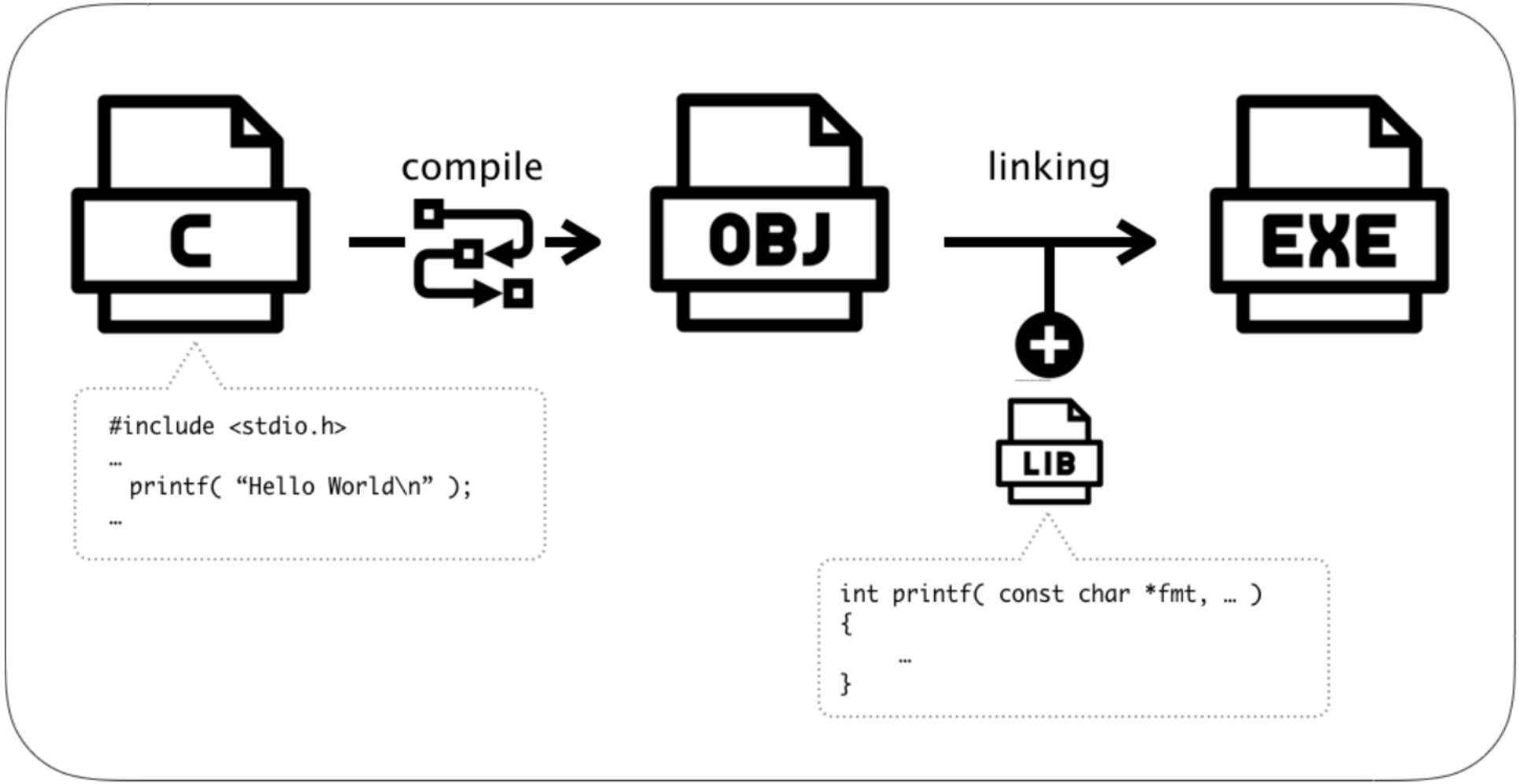



그림 출처 : https://bpsecblog.wordpress.com/2016/03/07/about_got_plt_1/

- Dynamic linking

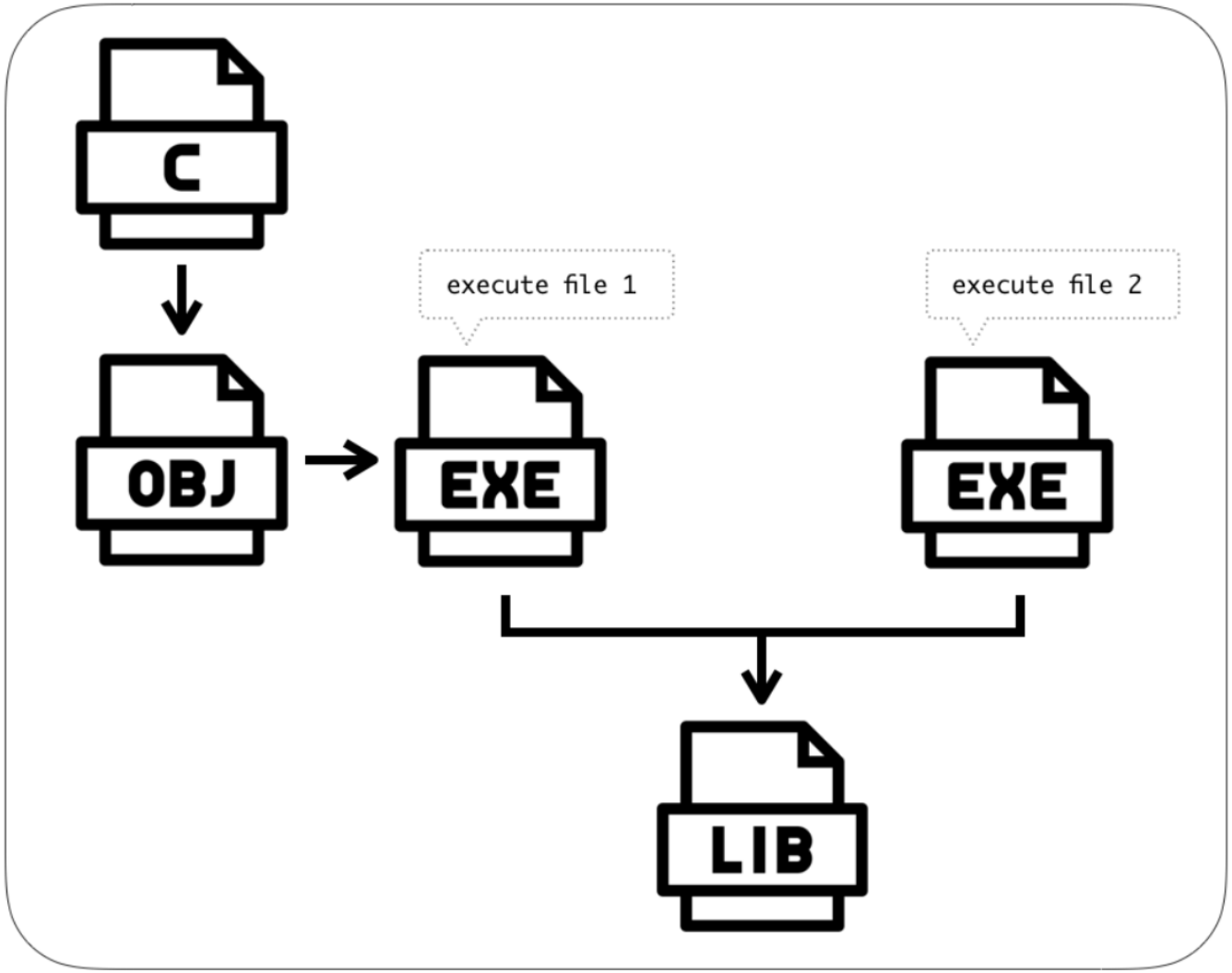



그림 출처 : https://bpsecblog.wordpress.com/2016/03/07/about_got_plt_1/

- Static Linking

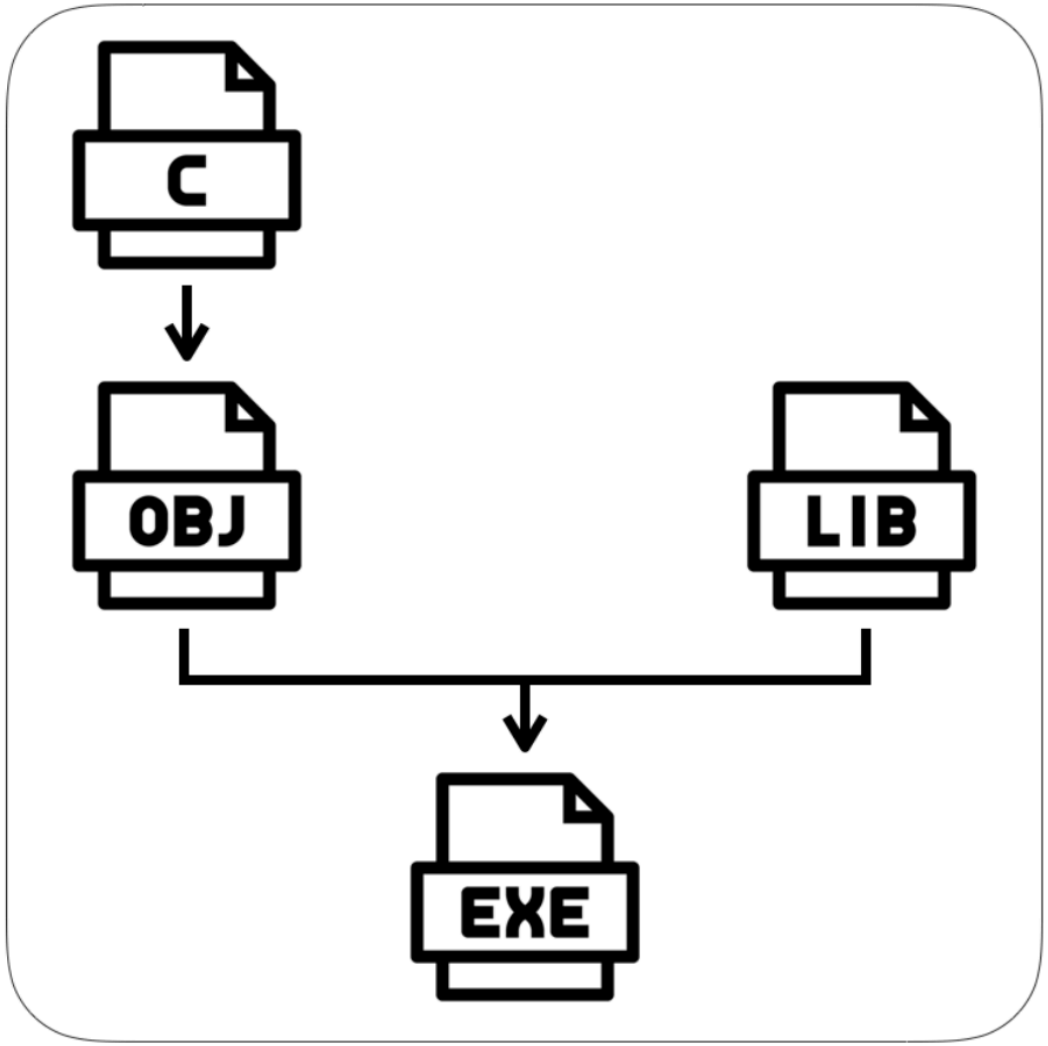
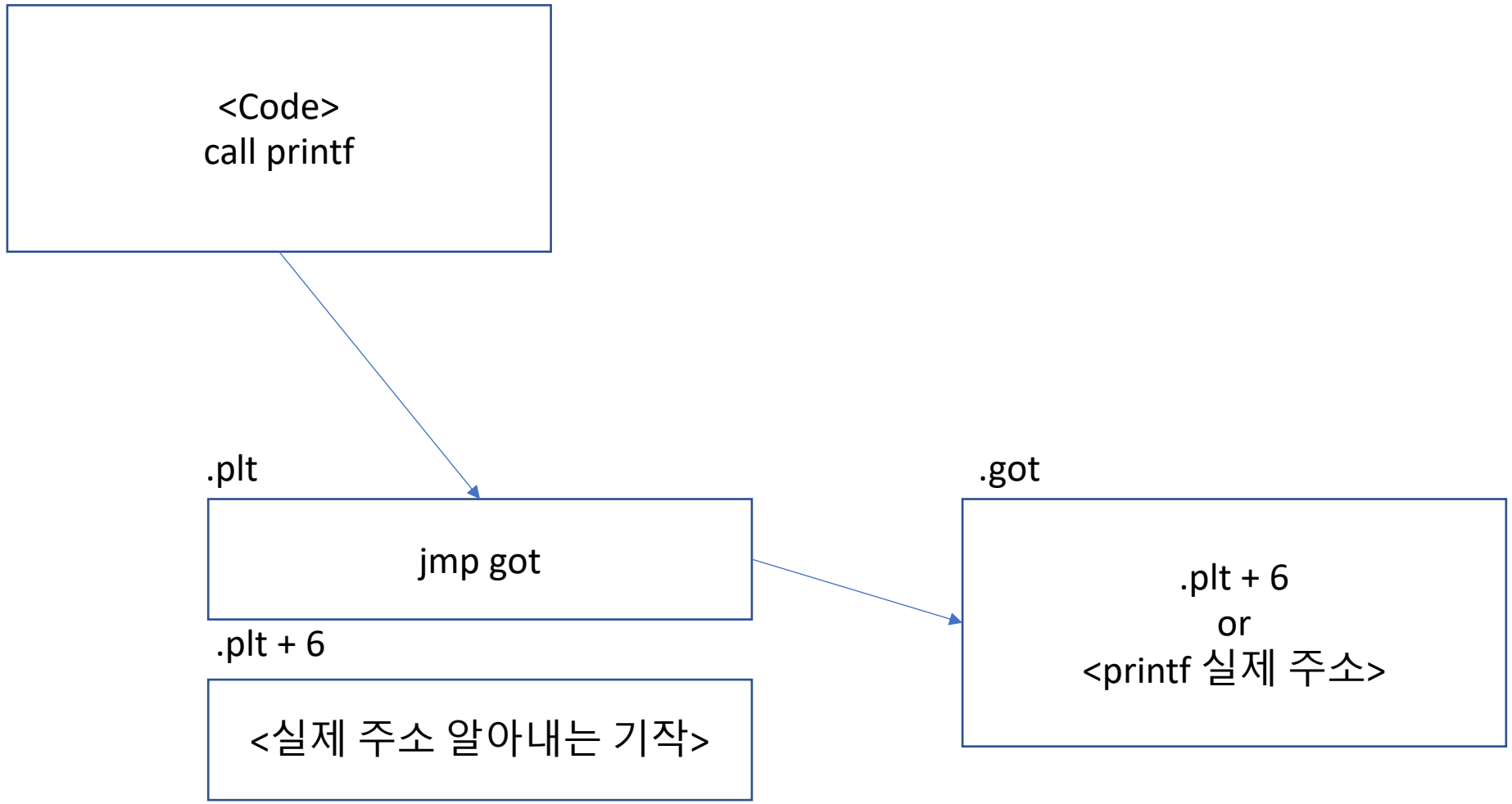
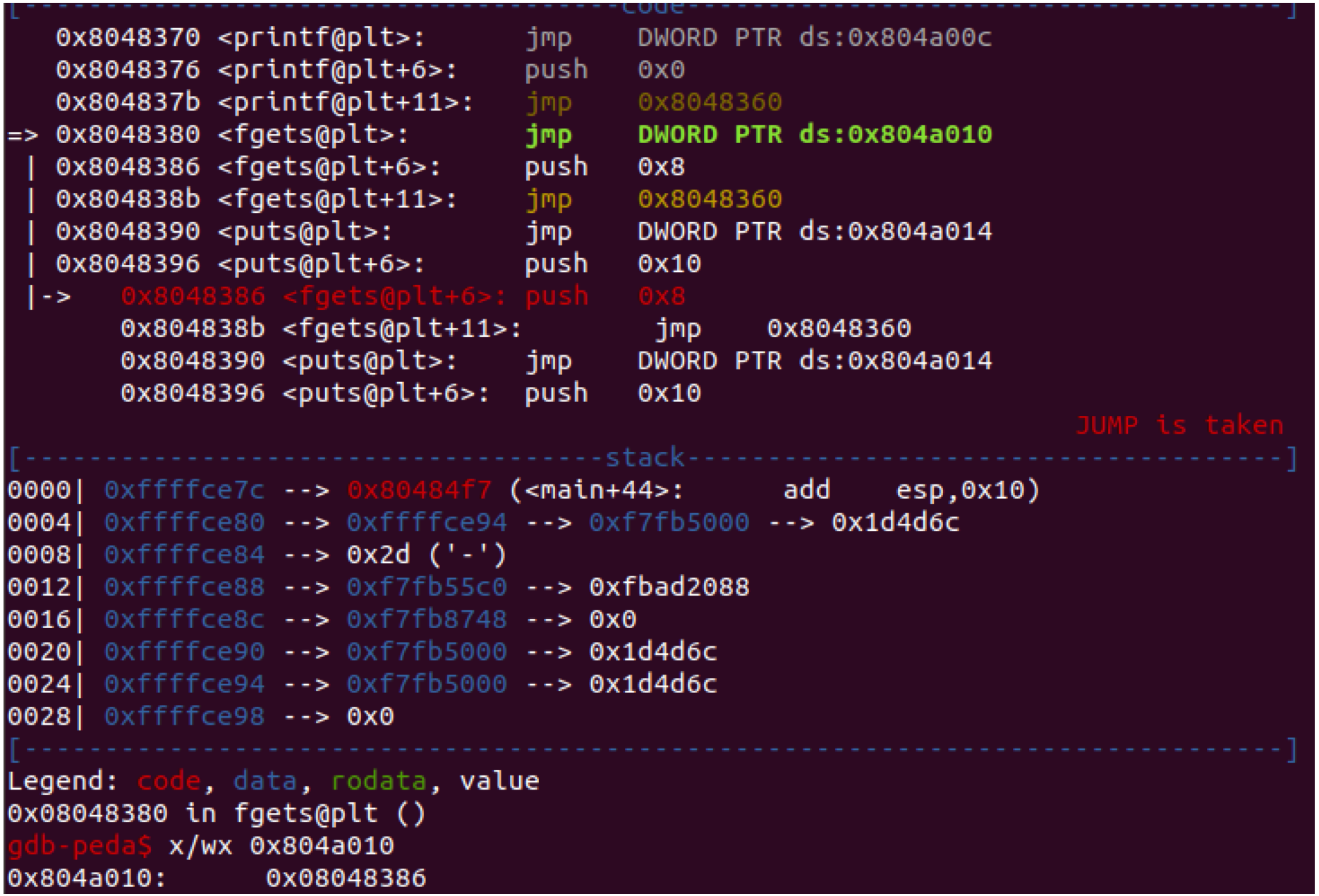



그림 출처 : https://bpsecblog.wordpress.com/2016/03/07/about_got_plt_1/


<Code>
call printf
.plt
jmp got
.plt + 6
<실제 주소 알아내는 기작>
.got
.plt + 6
or
<printf 실제 주소>


0x80484ec <main+33>: push 0x2d
0x80484ee <main+35>: lea eax,[ebp-0x34]
0x80484f1 <main+38>: push eax
=> 0x80484f2 <main+39>: call 0x8048380 <fgets@plt>
0x80484f7 <main+44>: add esp,0x10
0x80484fa <main+47>: sub esp,0x8
0x80484fd <main+50>: lea eax,[ebp-0x34]
0x8048500 <main+53>: push eax
Guessed arguments:
arg[0]: 0xffffce94 --> 0xf7fb5000 --> 0x1d4d6c
arg[1]: 0x2d ('-')
arg[2]: 0xf7fb55c0 --> 0xfbad2088
arg[3]: 0xf7fb8748 --> 0x0
arg[4]: 0xf7fb5000 --> 0x1d4d6c
arg[5]: 0xf7fb5000 --> 0x1d4d6c
stack
0000| 0xffffce80 --> 0xffffce94 --> 0xf7fb5000 --> 0x1d4d6c
0004| 0xffffce84 --> 0x2d ('-')
0008| 0xffffce88 --> 0xf7fb55c0 --> 0xfbad2088
0012| 0xffffce8c --> 0xf7fb8748 --> 0x0
0016| 0xffffce90 --> 0xf7fb5000 --> 0x1d4d6c
0020| 0xffffce94 --> 0xf7fb5000 --> 0x1d4d6c
0024| 0xffffce98 --> 0x0
0028| 0xffffce9c --> 0xf7e1020b (add esp,0x10)
Legend: code, data, rodata, value
0x080484f2 in main ()
gdb-peda$

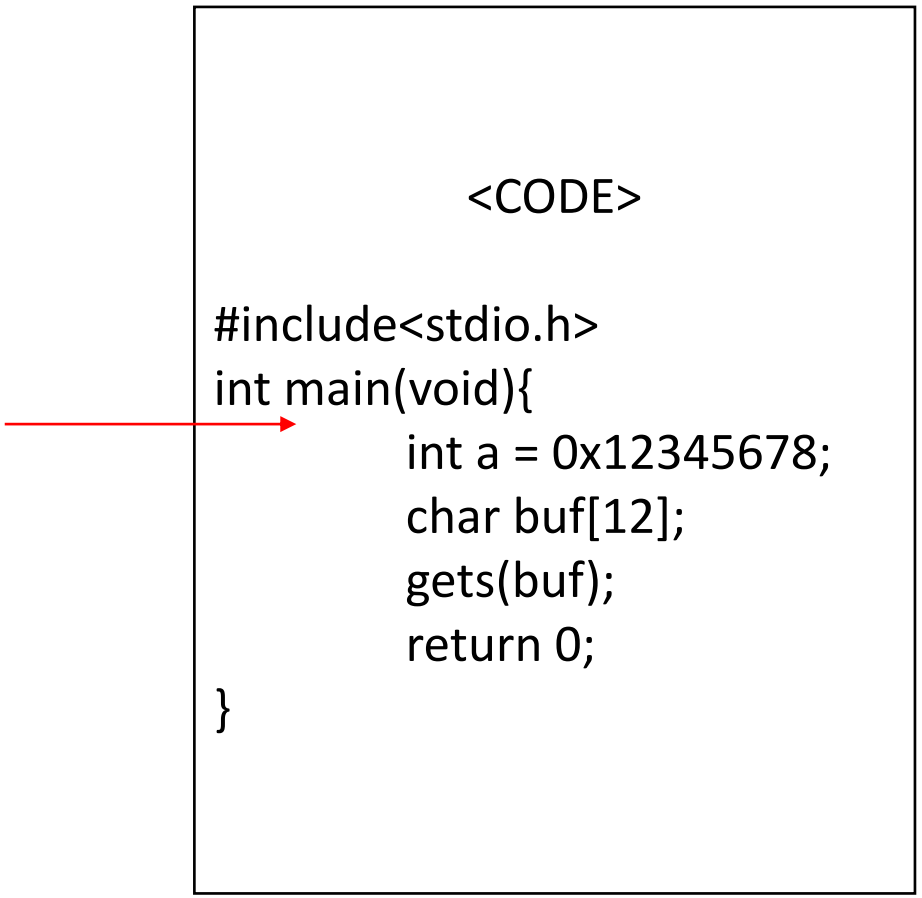

[----------code----------]
0x8048370 <printf@plt>: jmp DWORD PTR ds:0x804a00c
0x8048376 <printf@plt+6>: push 0x0
0x804837b <printf@plt+11>: jmp 0x8048360
=> 0x8048380 <fgets@plt>: jmp DWORD PTR ds:0x804a010
| 0x8048386 <fgets@plt+6>: push 0x8
| 0x804838b <fgets@plt+11>: jmp 0x8048360
| 0x8048390 <puts@plt>: jmp DWORD PTR ds:0x804a014
| 0x8048396 <puts@plt+6>: push 0x10
|-> 0x8048386 <fgets@plt+6>: push 0x8
0x804838b <fgets@plt+11>: jmp 0x8048360
0x8048390 <puts@plt>: jmp DWORD PTR ds:0x804a014
0x8048396 <puts@plt+6>: push 0x10
JUMP is taken
[----------stack----------]
0000| 0xffffce7c --> 0x80484f7 (<main+44>: add esp,0x10)
0004| 0xffffce80 --> 0xffffce94 --> 0xf7fb5000 --> 0x1d4d6c
0008| 0xffffce84 --> 0x2d ('-')
0012| 0xffffce88 --> 0xf7fb55c0 --> 0xfbad2088
0016| 0xffffce8c --> 0xf7fb8748 --> 0x0
0020| 0xffffce90 --> 0xf7fb5000 --> 0x1d4d6c
0024| 0xffffce94 --> 0xf7fb5000 --> 0x1d4d6c
0028| 0xffffce98 --> 0x0
[----------]
Legend: code, data, rodata, value
0x08048380 in fgets@plt ()
gdb-peda$ x/wx 0x804a010
0x804a010: 0x08048386

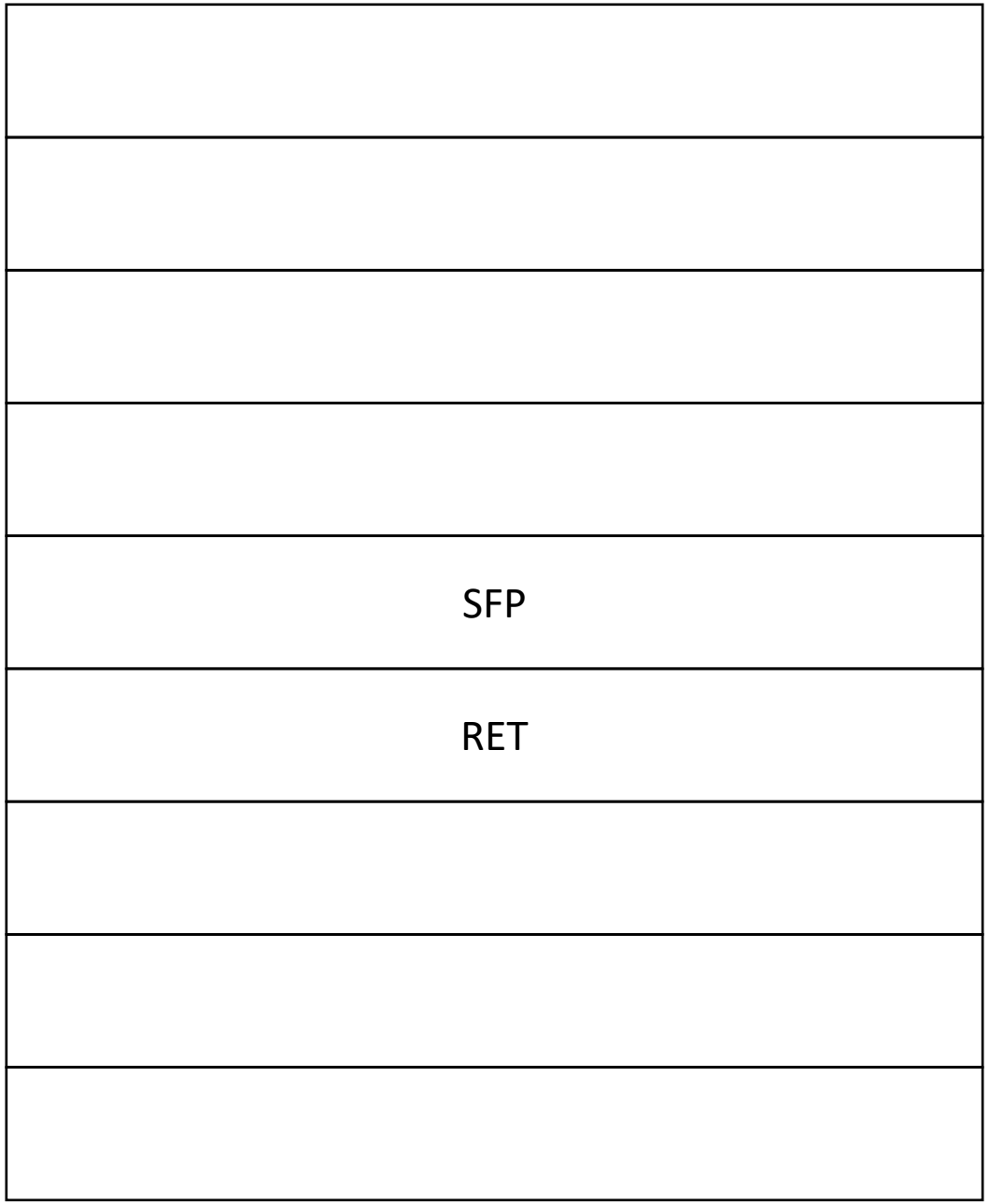
- PLT => jmp dword ptr [got]

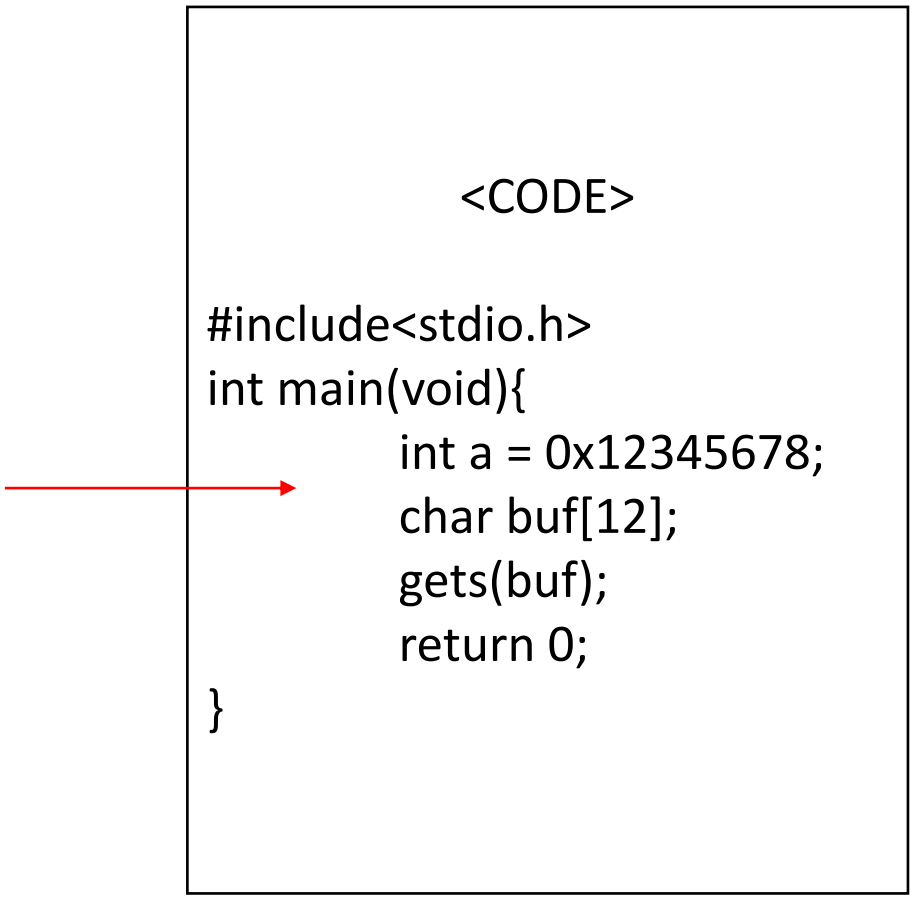
  즉, ' 함수를 실행하는 코드'를 가리킨다.

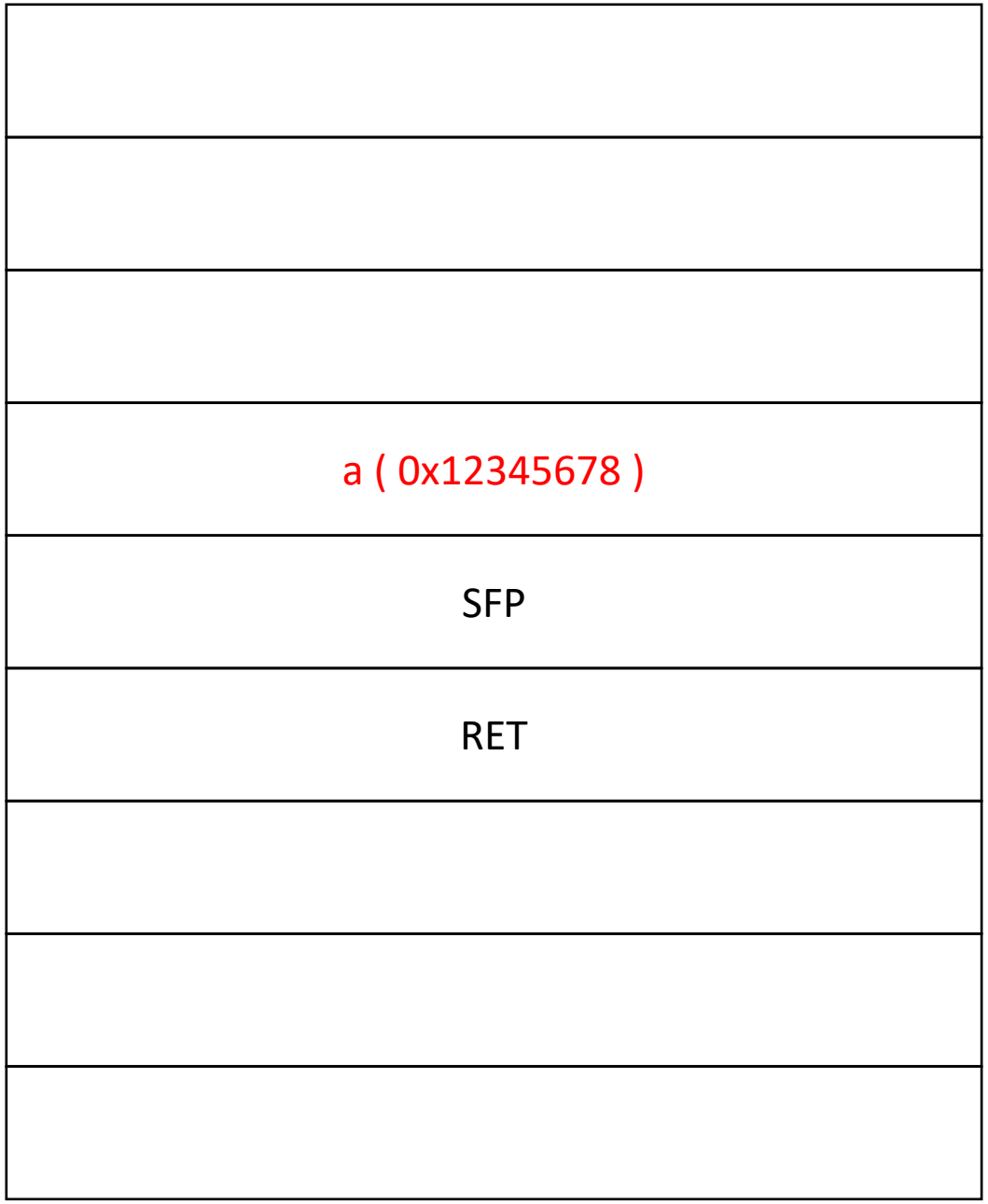
- GOT => 실제 함수 주소.

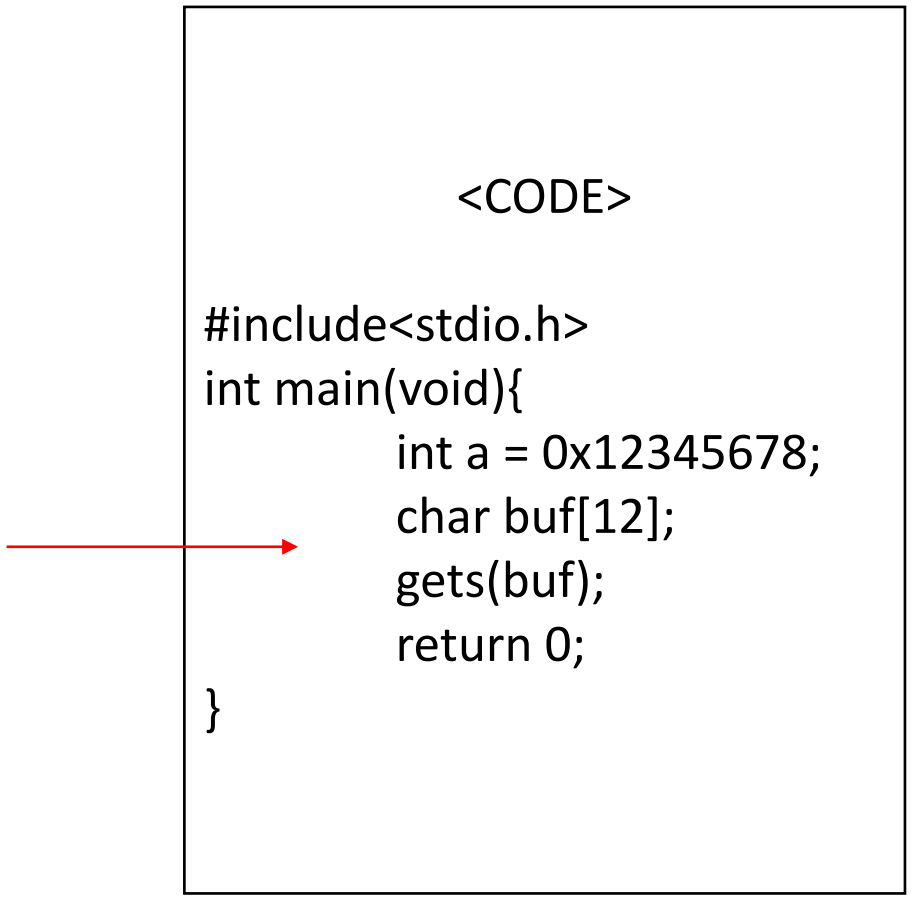
# Return to Library ( RTL )

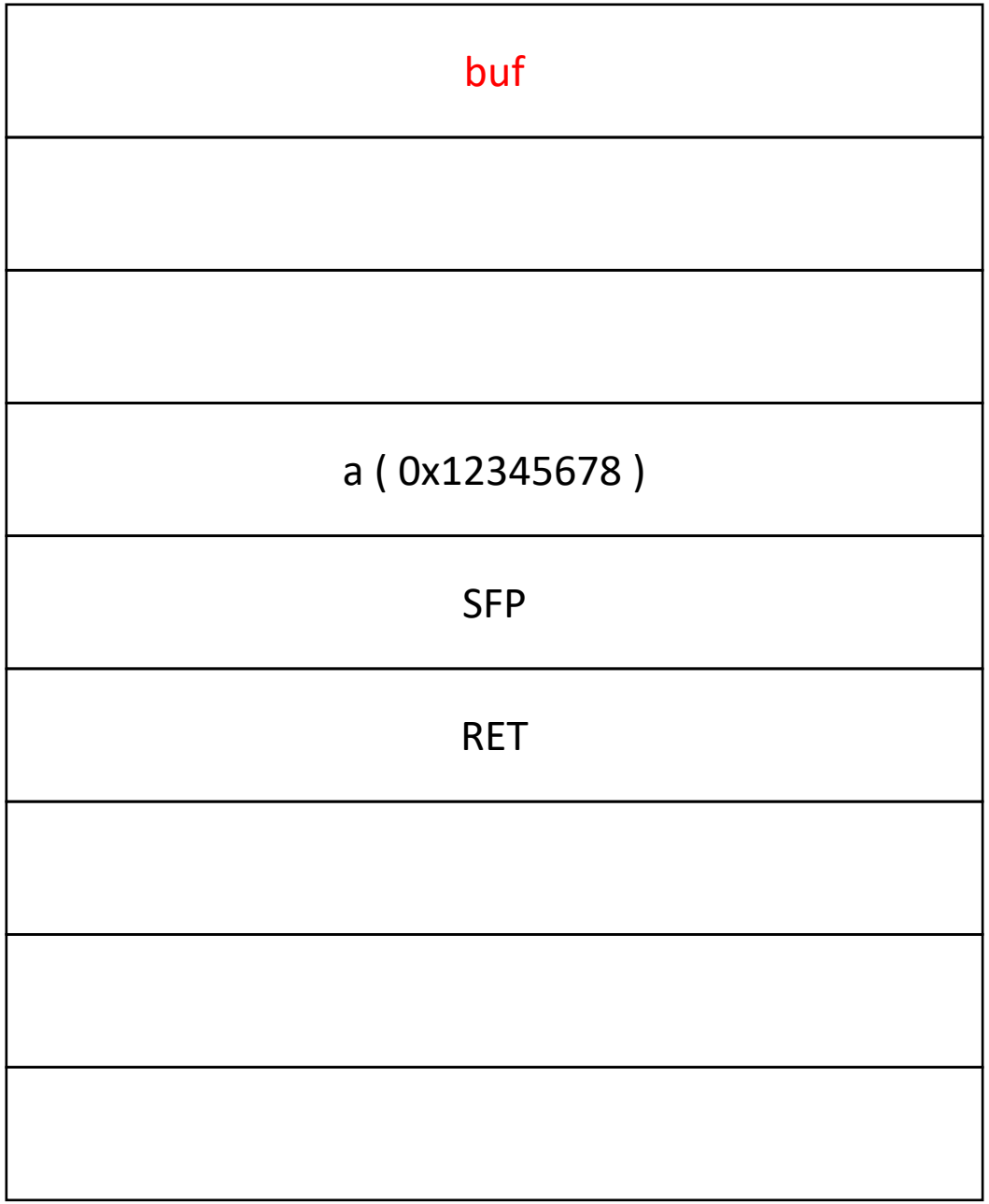

<CODE>
#include<stdio.h>
int main(void){
int a = 0x12345678;
char buf[12];
gets(buf);
return 0;
}


SFP
RET


<CODE>
#include<stdio.h>
int main(void){
int a = 0x12345678;
char buf[12];
gets(buf);
return 0;
}


a ( 0x12345678 )
SFP
RET


<CODE>
#include<stdio.h>
int main(void){
int a = 0x12345678;
char buf[12];
gets(buf);
return 0;
}


buf
a ( 0x12345678 )
SFP
RET

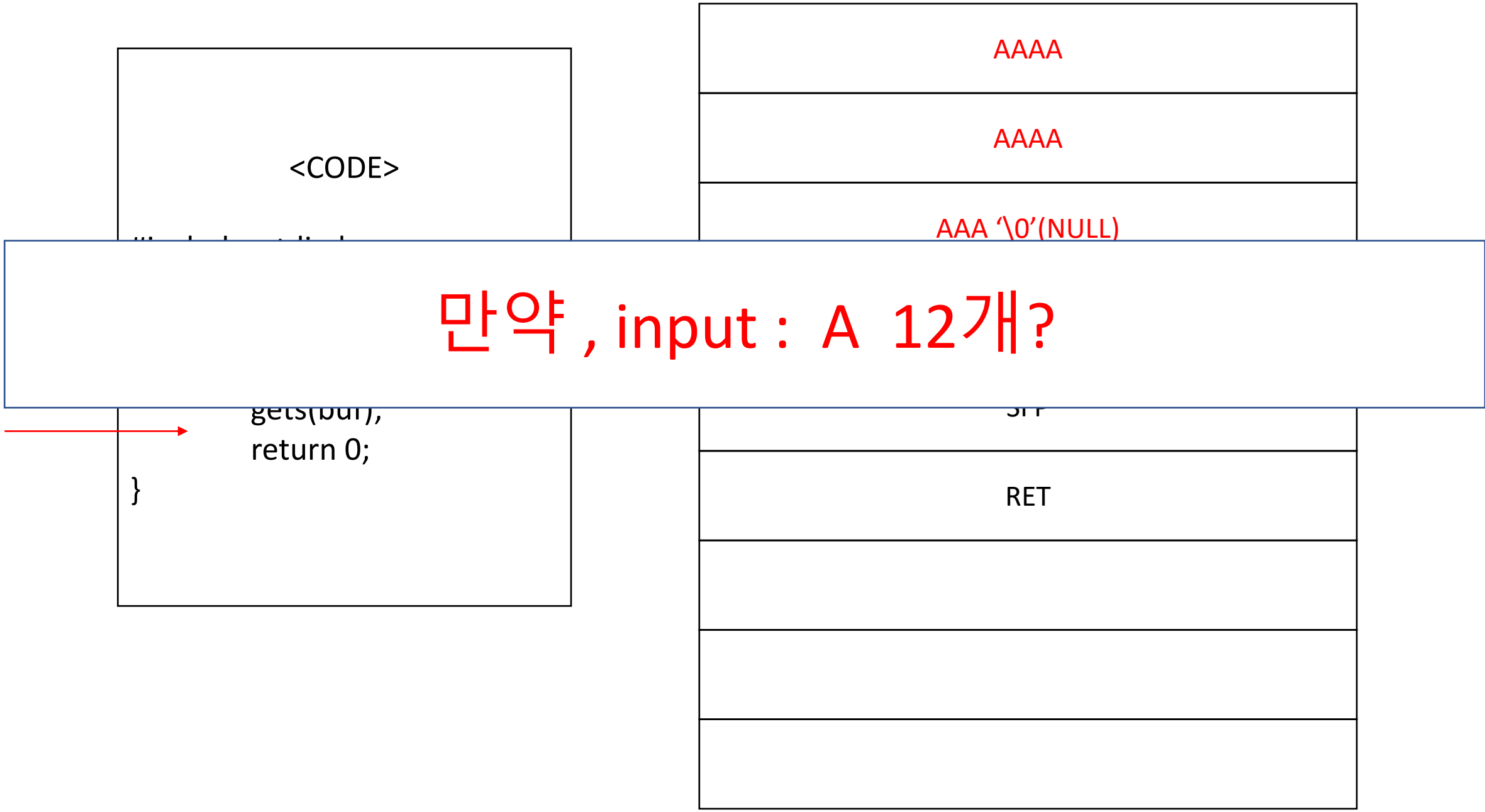
<CODE>

```
#include<stdio.h>
int main(void){
        int a = 0x12345678;
        char buf[12];
        gets(buf);
        return 0;
}
```



input : A 11개

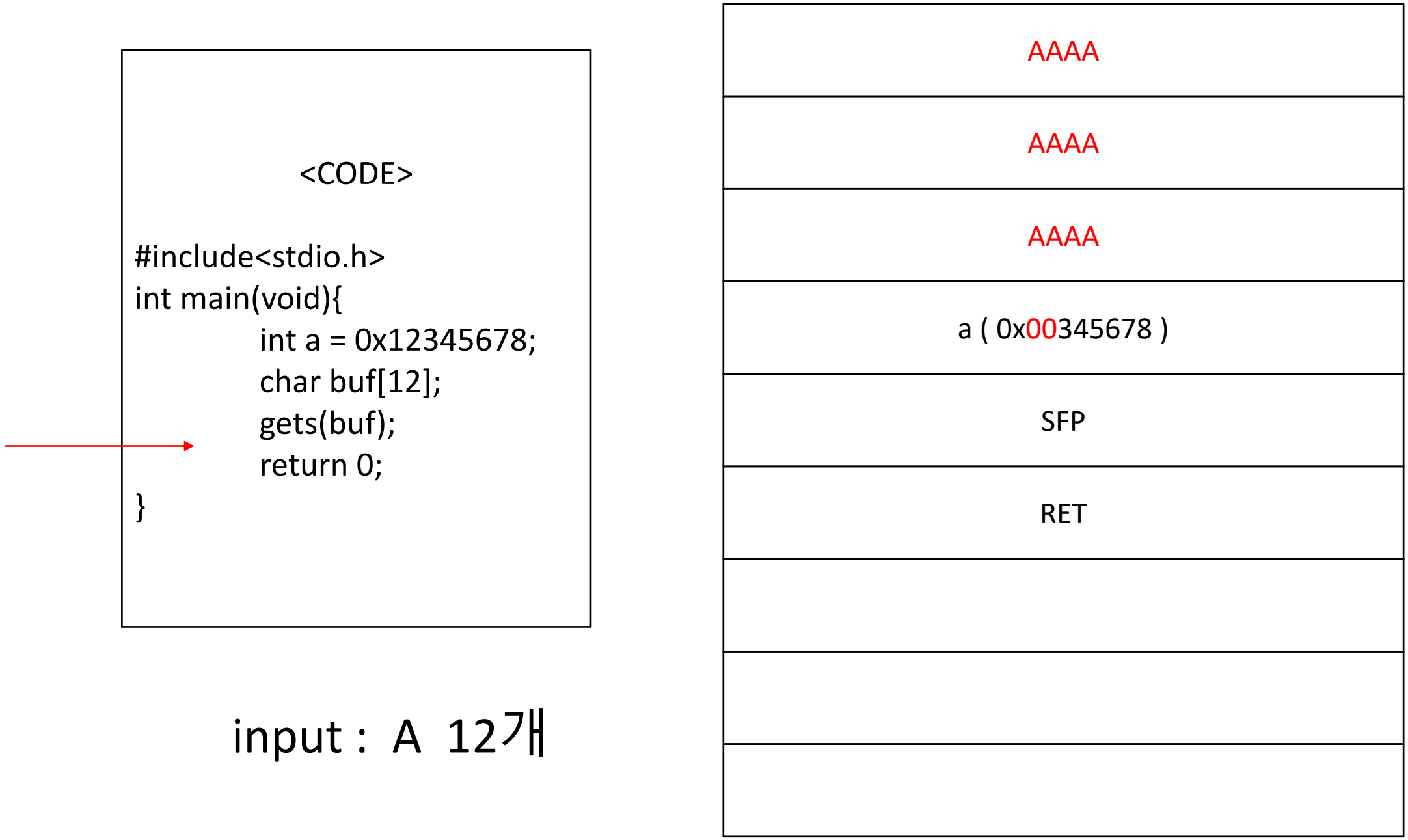
| |
|---|
| AAAA |
| AAAA |
| AAA ‘\0’(NULL) |
| a ( 0x12345678 ) |
| SFP |
| RET |
| |
| |
| |


<CODE>
gets(buf);
return 0;
}
AAAA
AAAA
AAA '\0'(NULL)
RET
만약 , input : A 12개?

<CODE>

```
#include<stdio.h>
int main(void){
        int a = 0x12345678;
        char buf[12];
        gets(buf);
        return 0;
}
```

input : A 12개



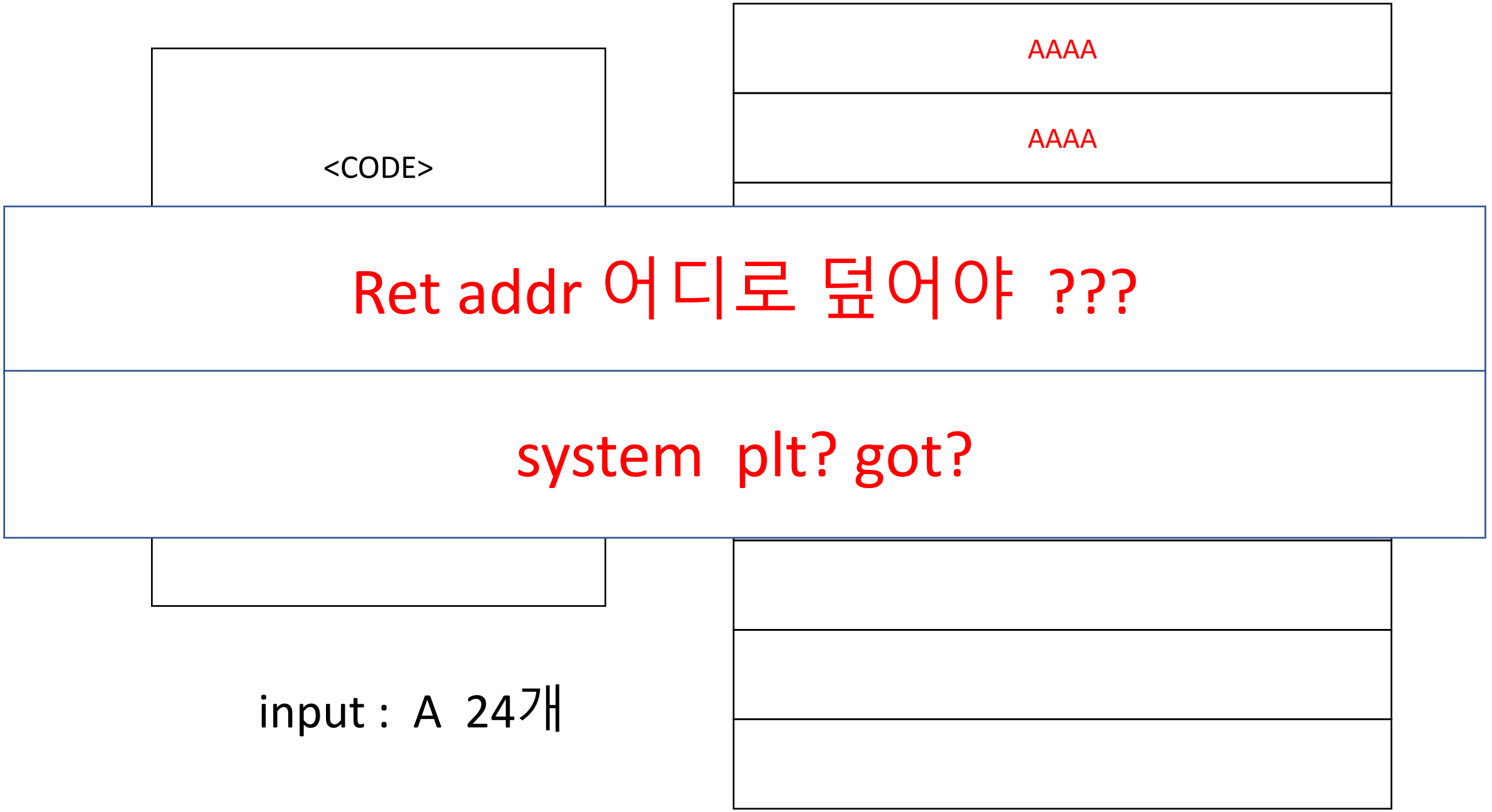
<CODE>

```
#include<stdio.h>
int main(void){
        int a = 0x12345678;
        char buf[12];
        gets(buf);
        return 0;
}
```



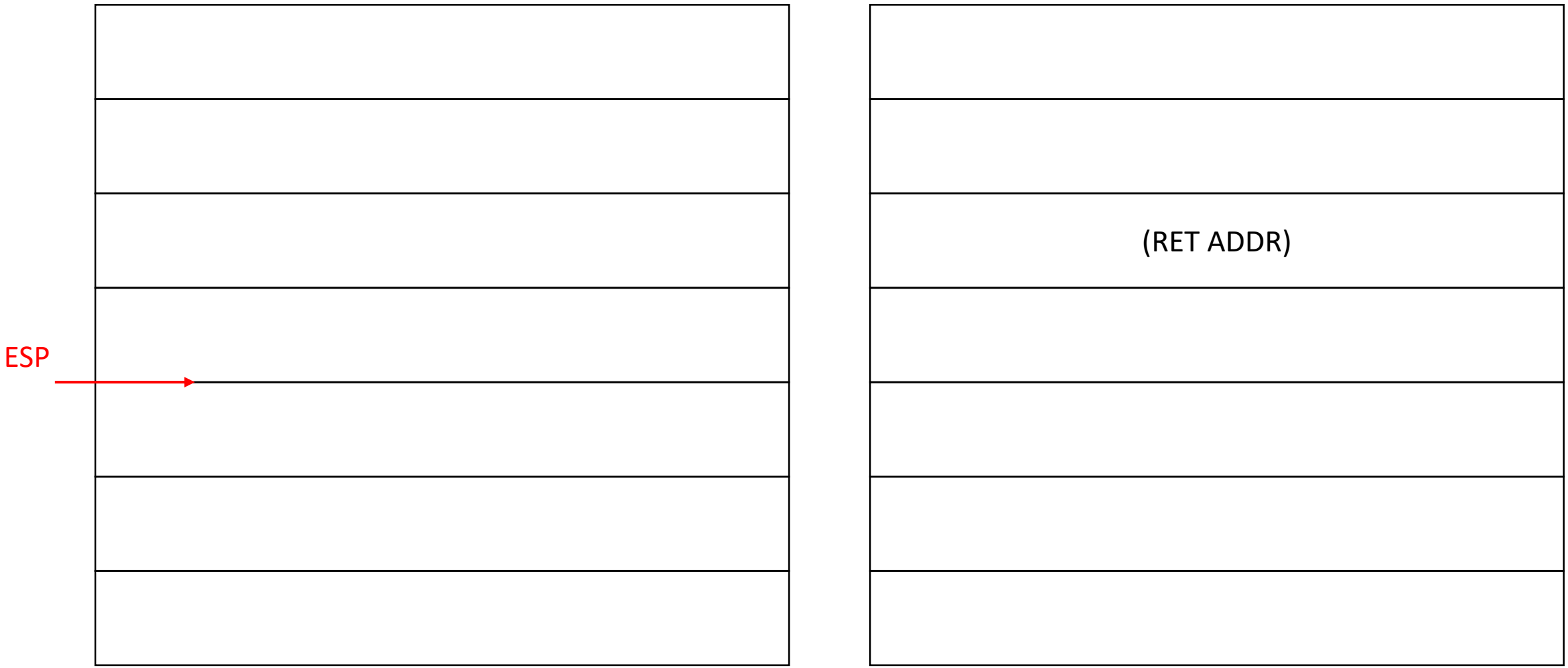
input : A 24개

| AAAA |
| --- |
| AAAA |
| AAAA |
| AAAA |
| AAAA |
| AAAA ( < - Ret addr 부분 덮어씀!!) |
| |
| |
| |

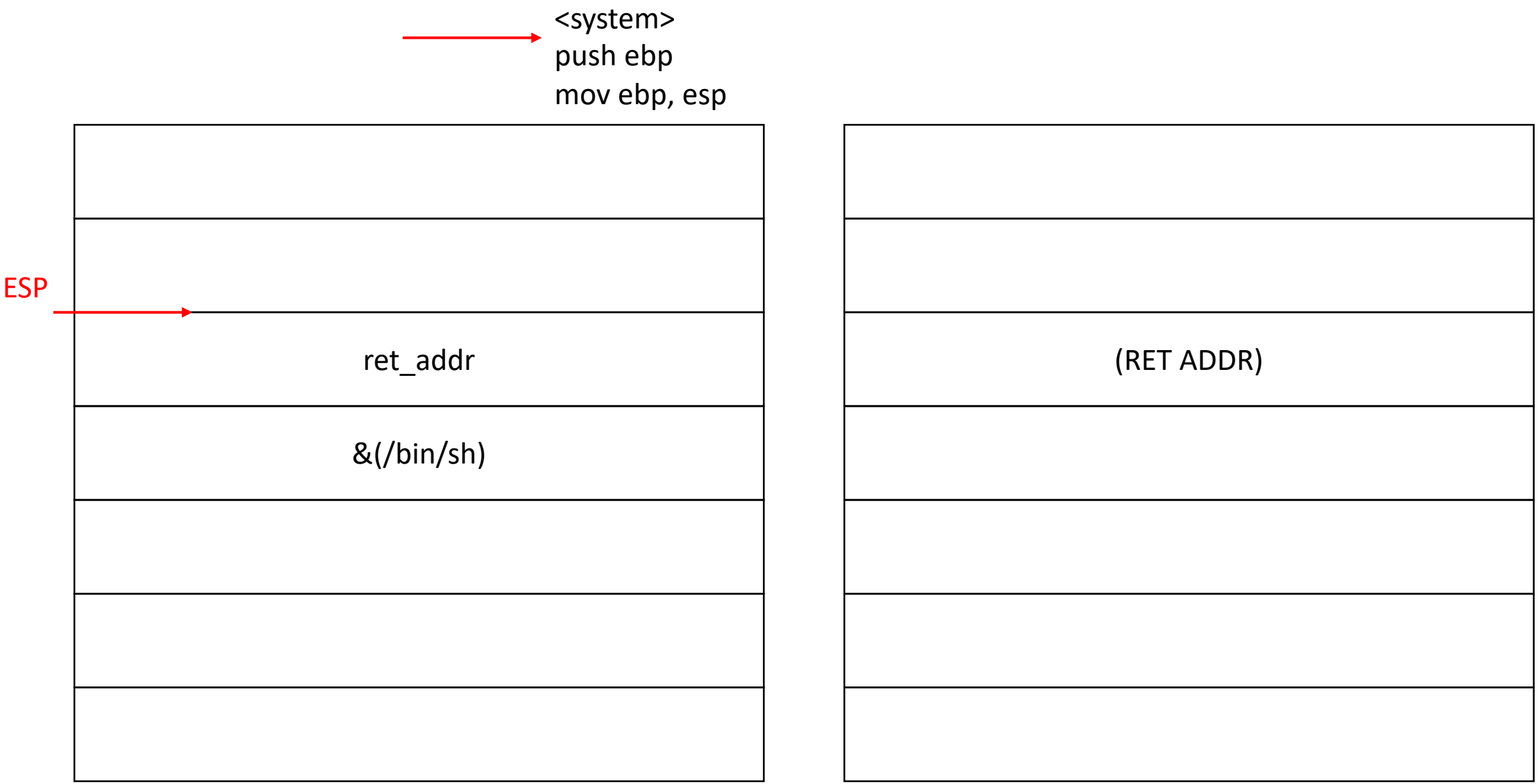

<CODE>
AAAA
AAAA
Ret addr 어디로 덮어야 ???
system plt? got?
input : A 24개

<CODE>
system(“/bin/sh”)

<system>
push ebp
mov ebp, esp



<CODE>
system(“/bin/sh”)



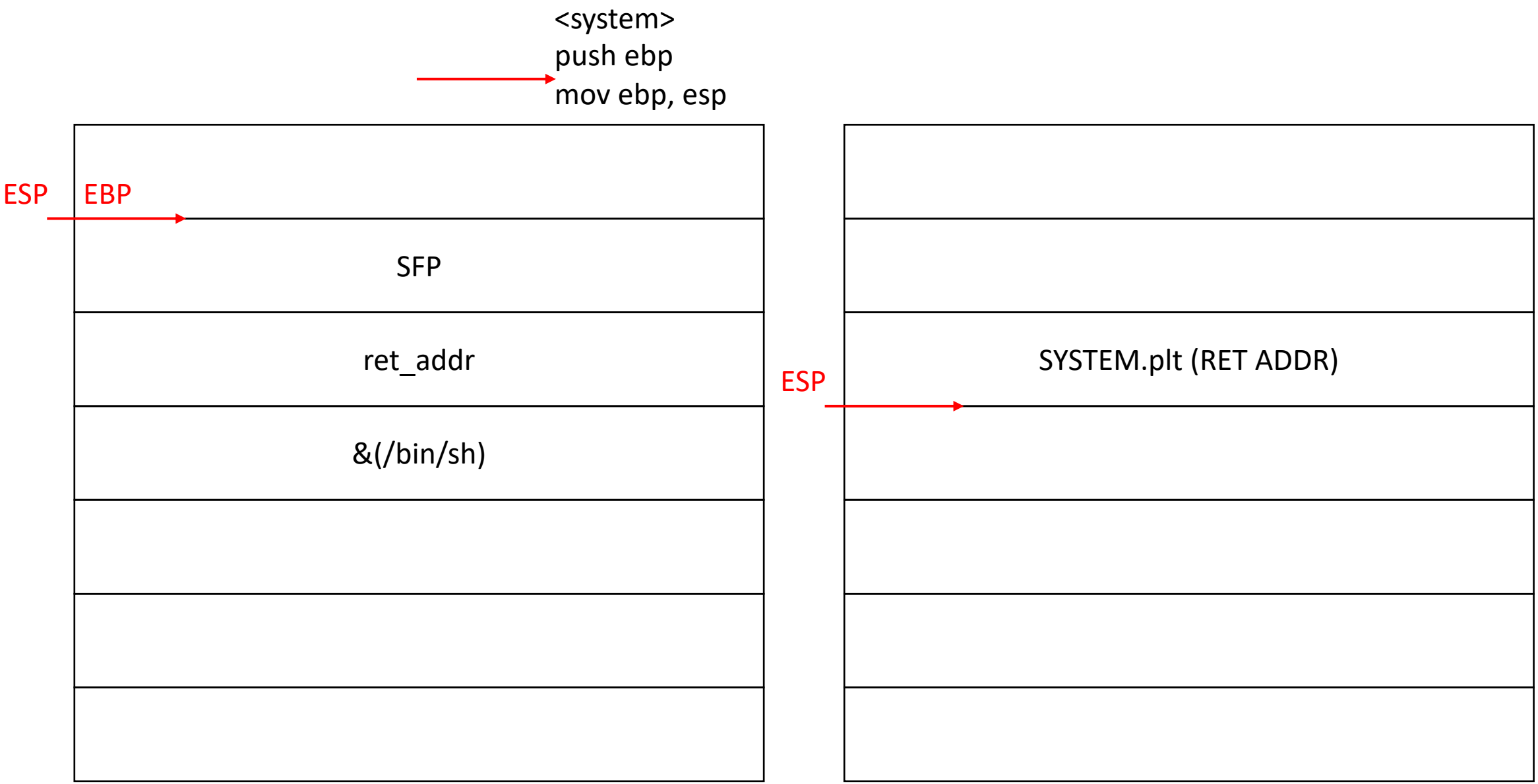
<CODE>
system("/bin/sh")

<system>
push ebp
mov ebp, esp

<CODE>
system("/bin/sh")

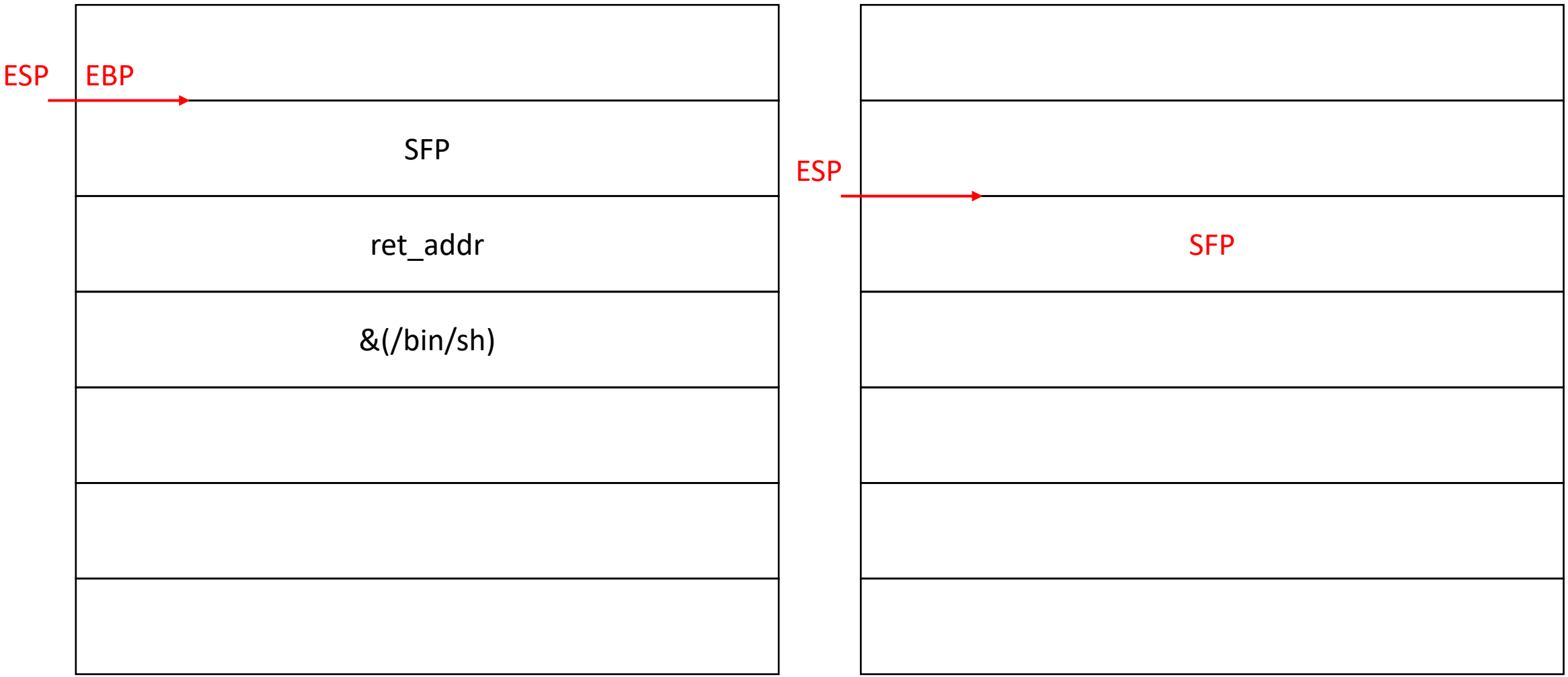
<system>
push ebp
mov ebp, esp

<CODE>
system("/bin/sh")



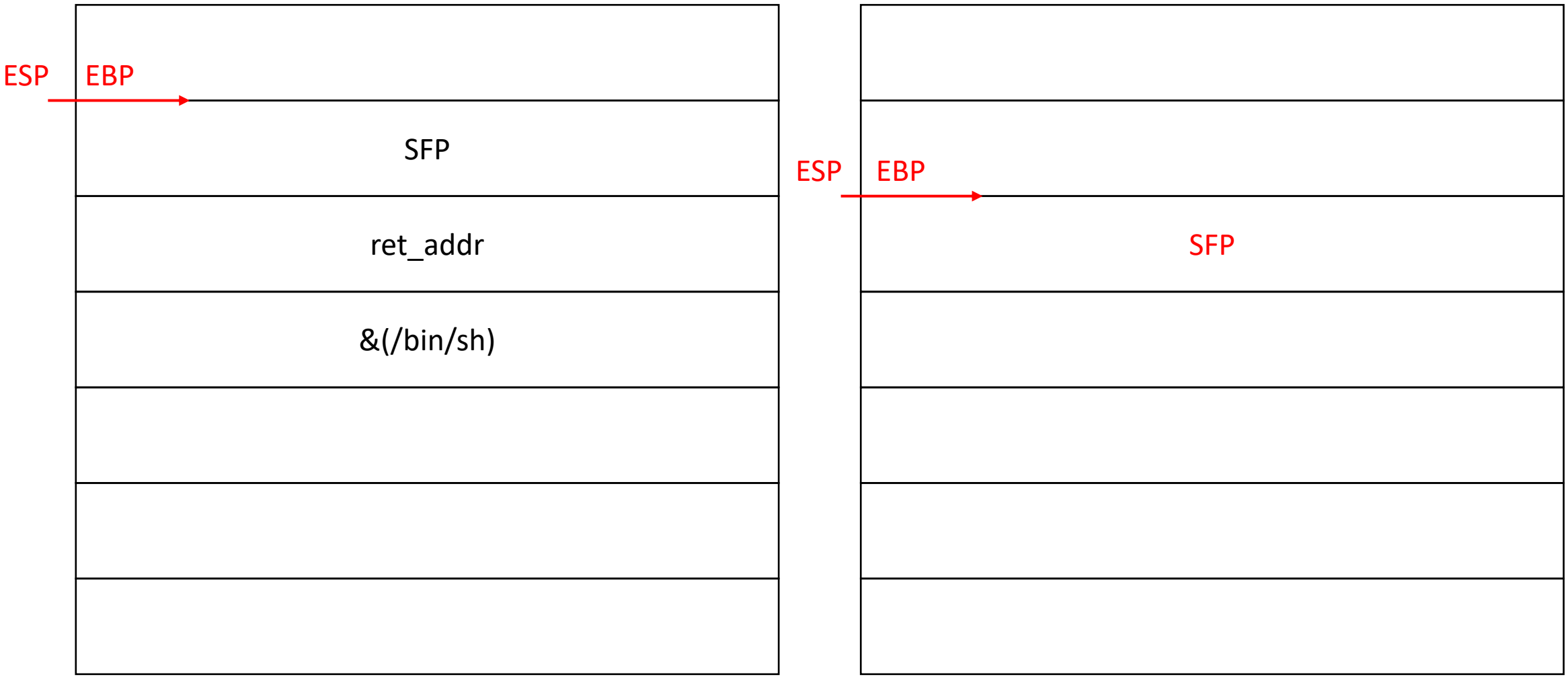
<CODE>
system("/bin/sh")

<system>
push ebp
mov ebp, esp



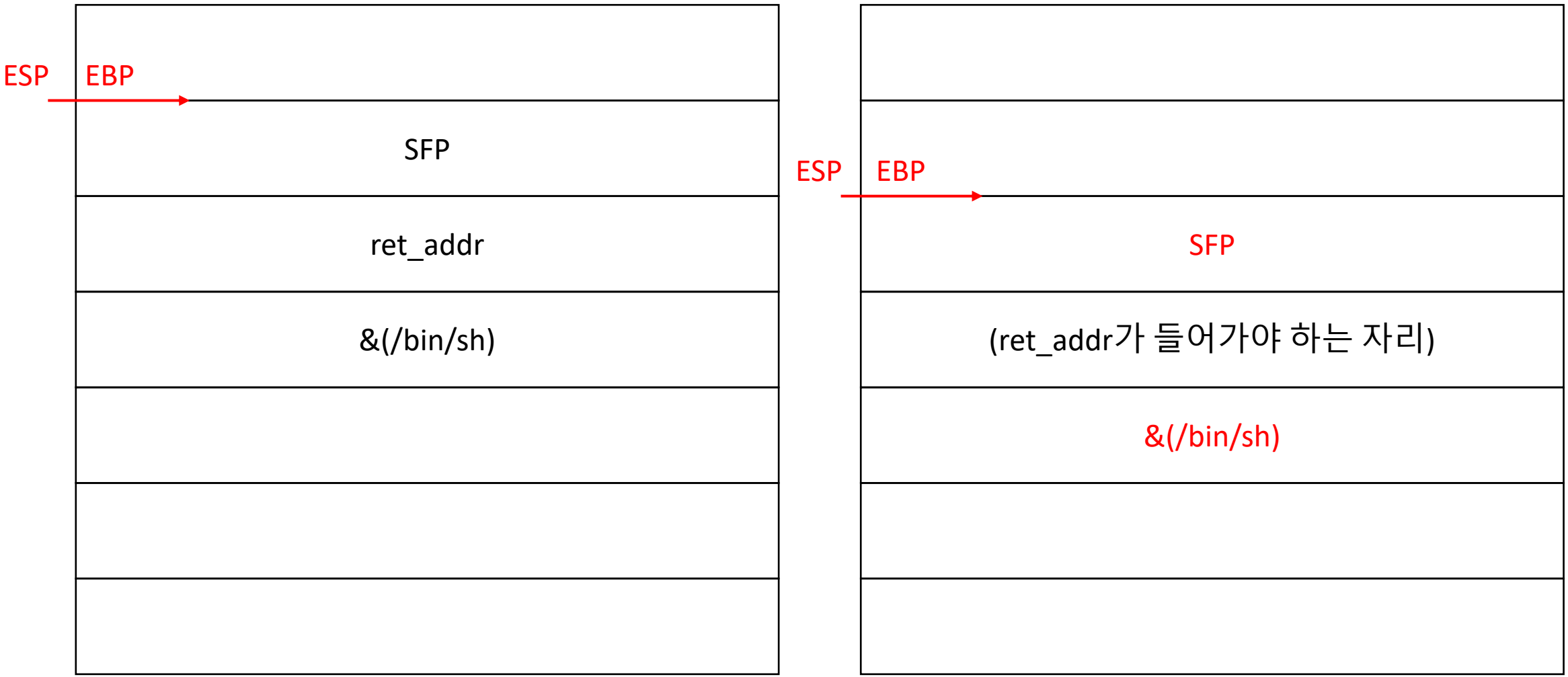
<CODE>
system(“/bin/sh”)

<system>
push ebp
mov ebp, esp



<CODE>
system(“/bin/sh”)

<system>
push ebp
mov ebp, esp

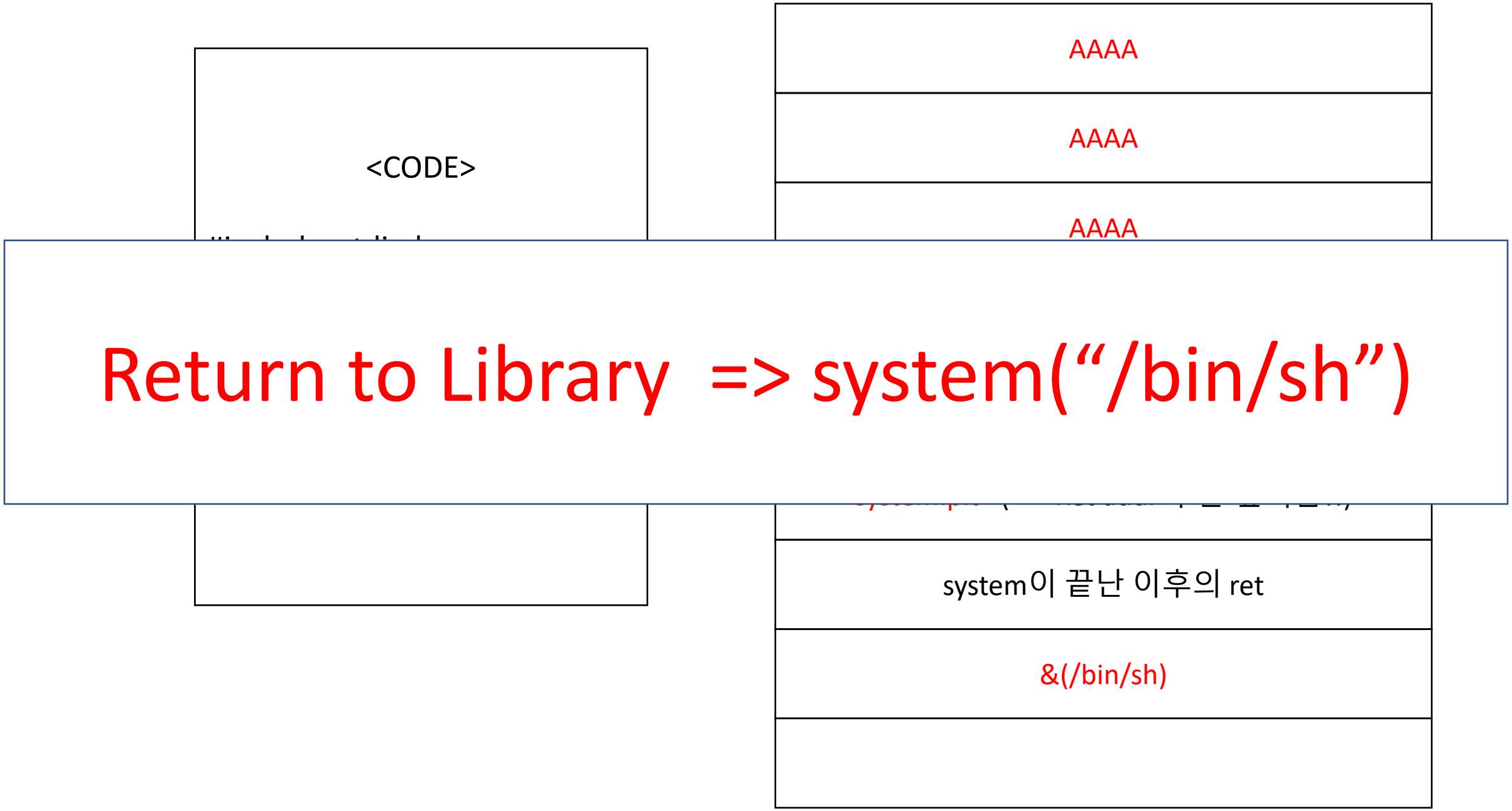


<CODE>

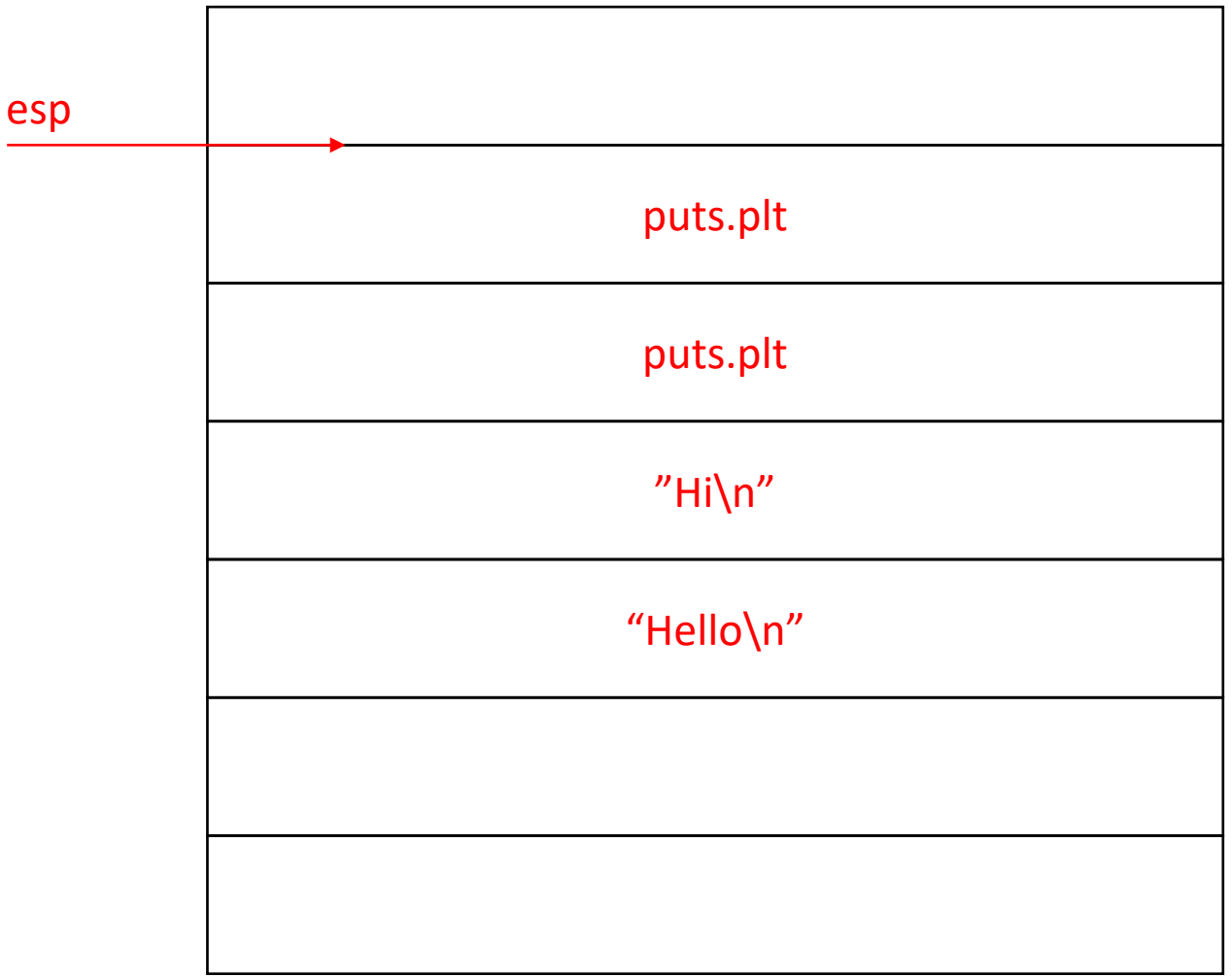
```
#include<stdio.h>
int main(void){
        int a = 0x12345678;
        char buf[12];
        gets(buf);
        return 0;
}
```



| |
|---|
| AAAA |
| AAAA |
| AAAA |
| AAAA |
| AAAA |
| system.plt ( < - Ret addr 부분 덮어씀!!) |
| system이 끝난 이후의 ret |
| &(/bin/sh) |
| |

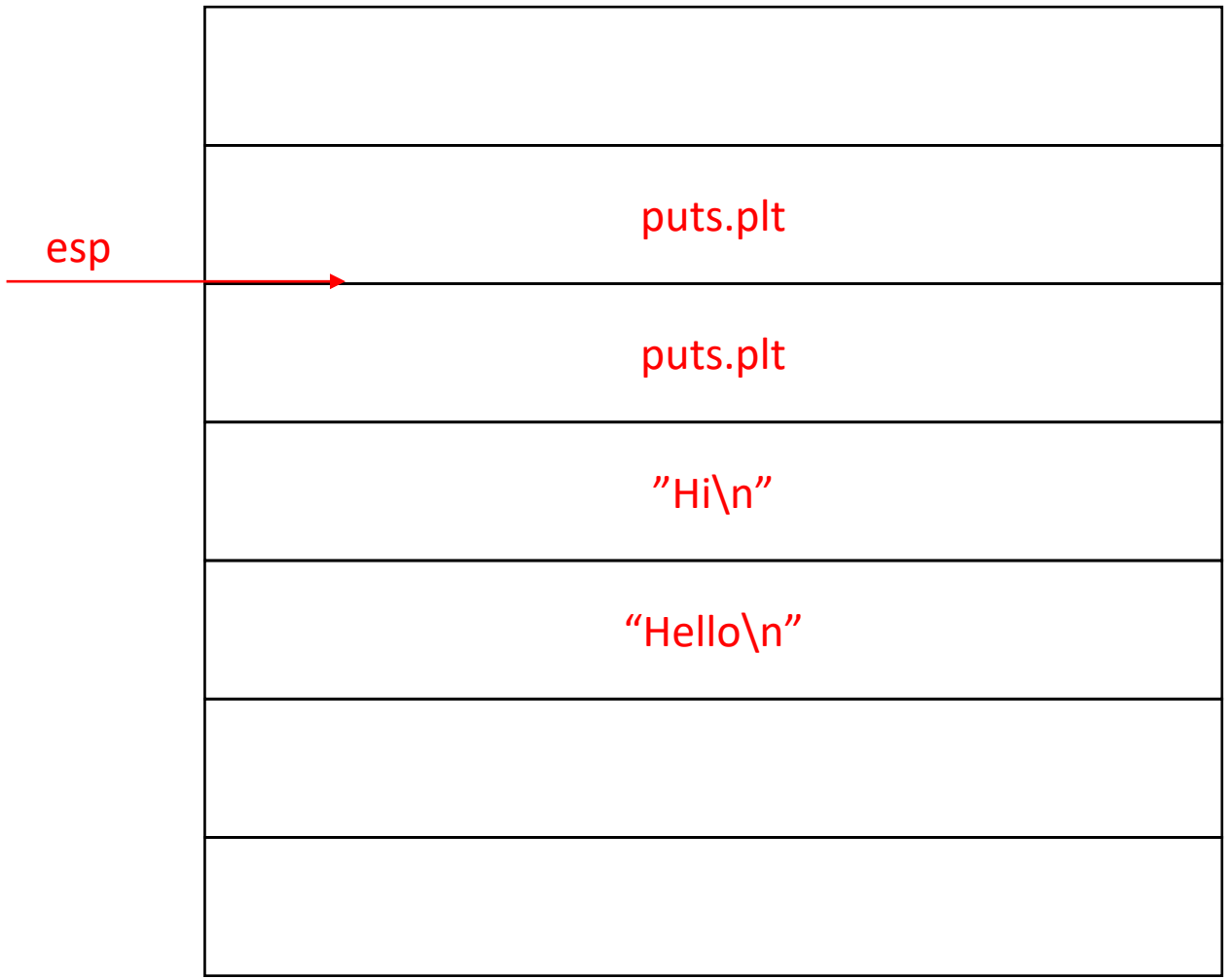

<CODE>
AAAA
AAAA
AAAA
system이 끝난 이후의 ret
&(/bin/sh)
Return to Library => system("/bin/sh")

<출력>



<출력>

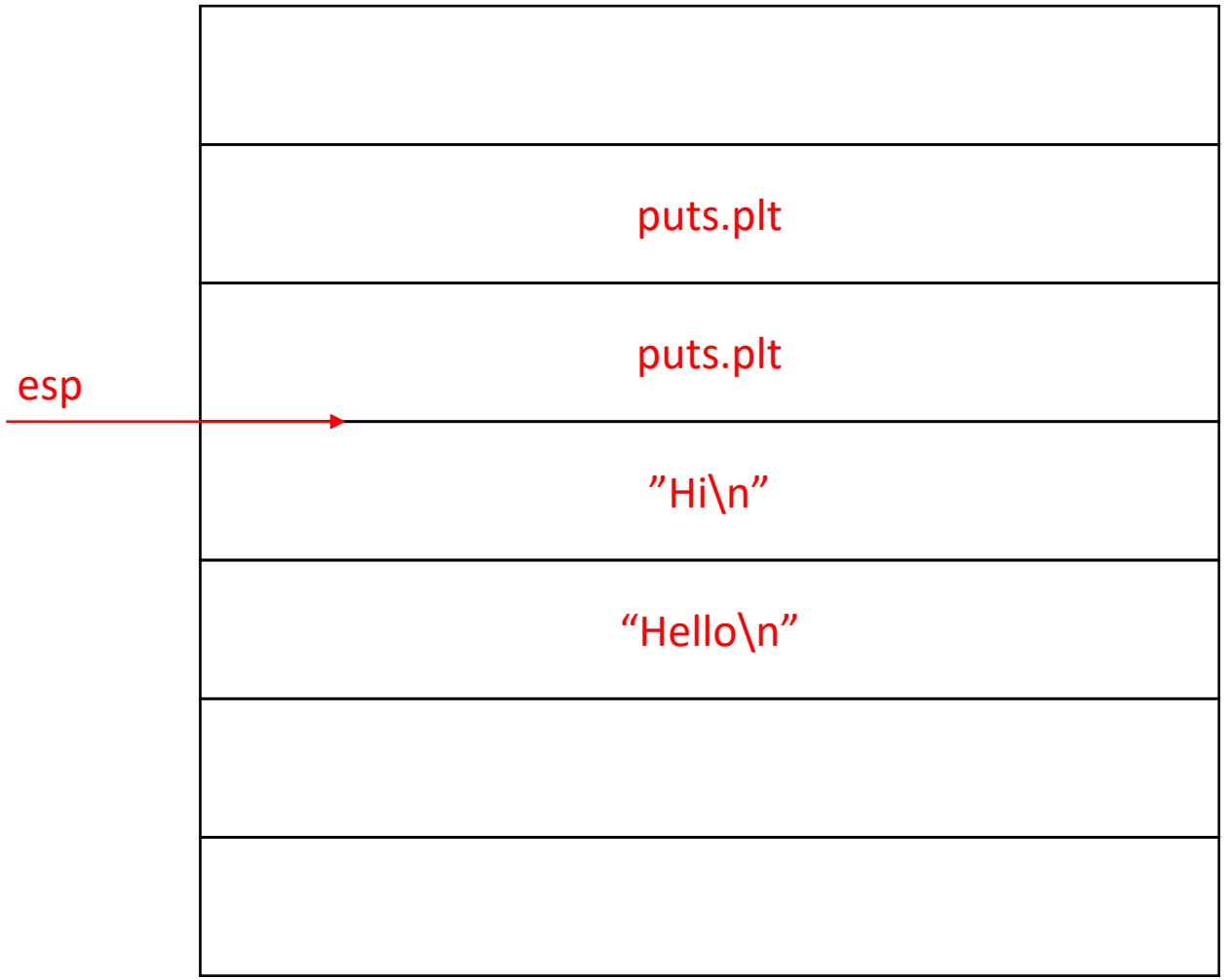
Hi



<출력>

Hi

Hello

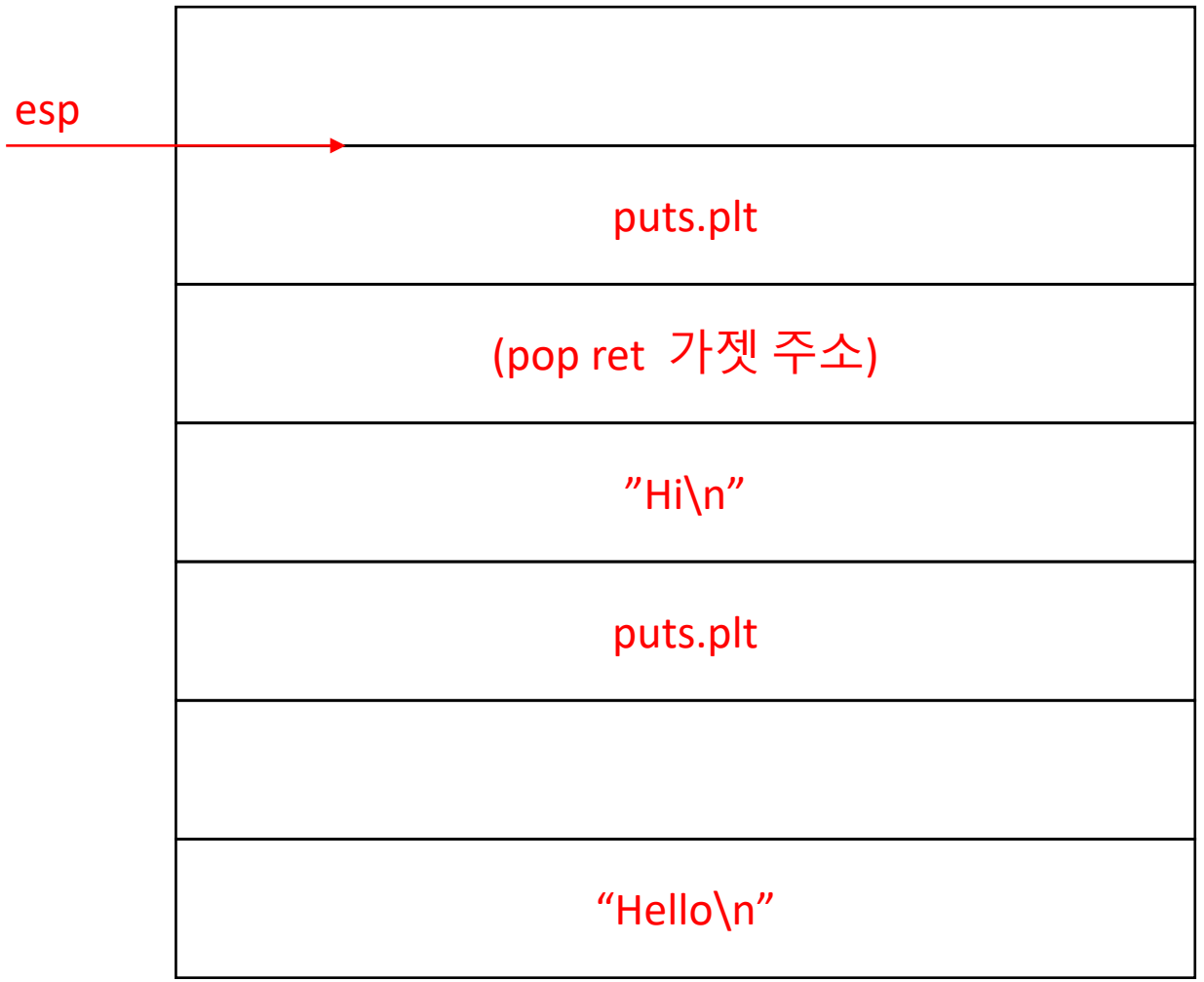


<출력>

3개는 못하려나??

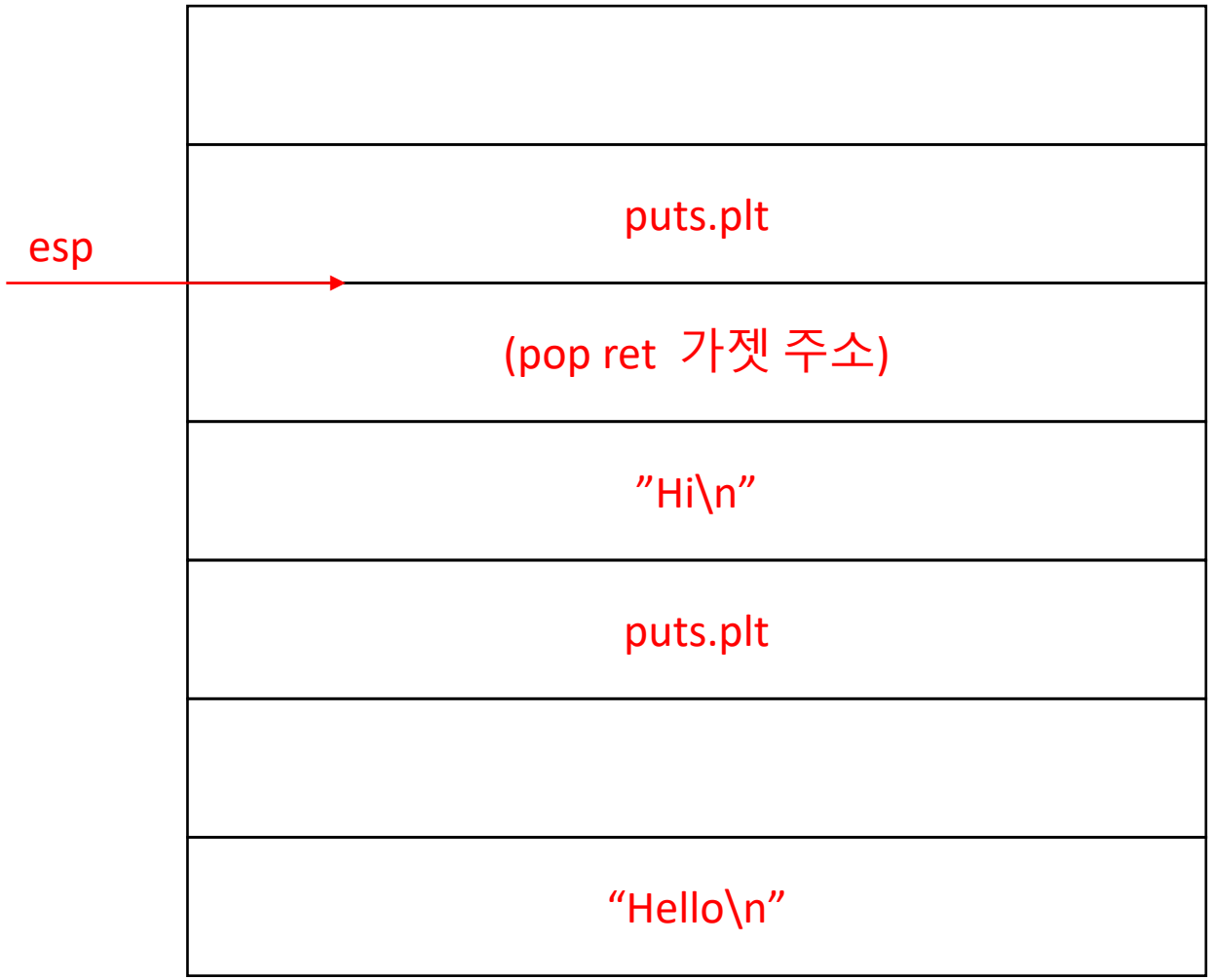
“Hello\n”

<출력>



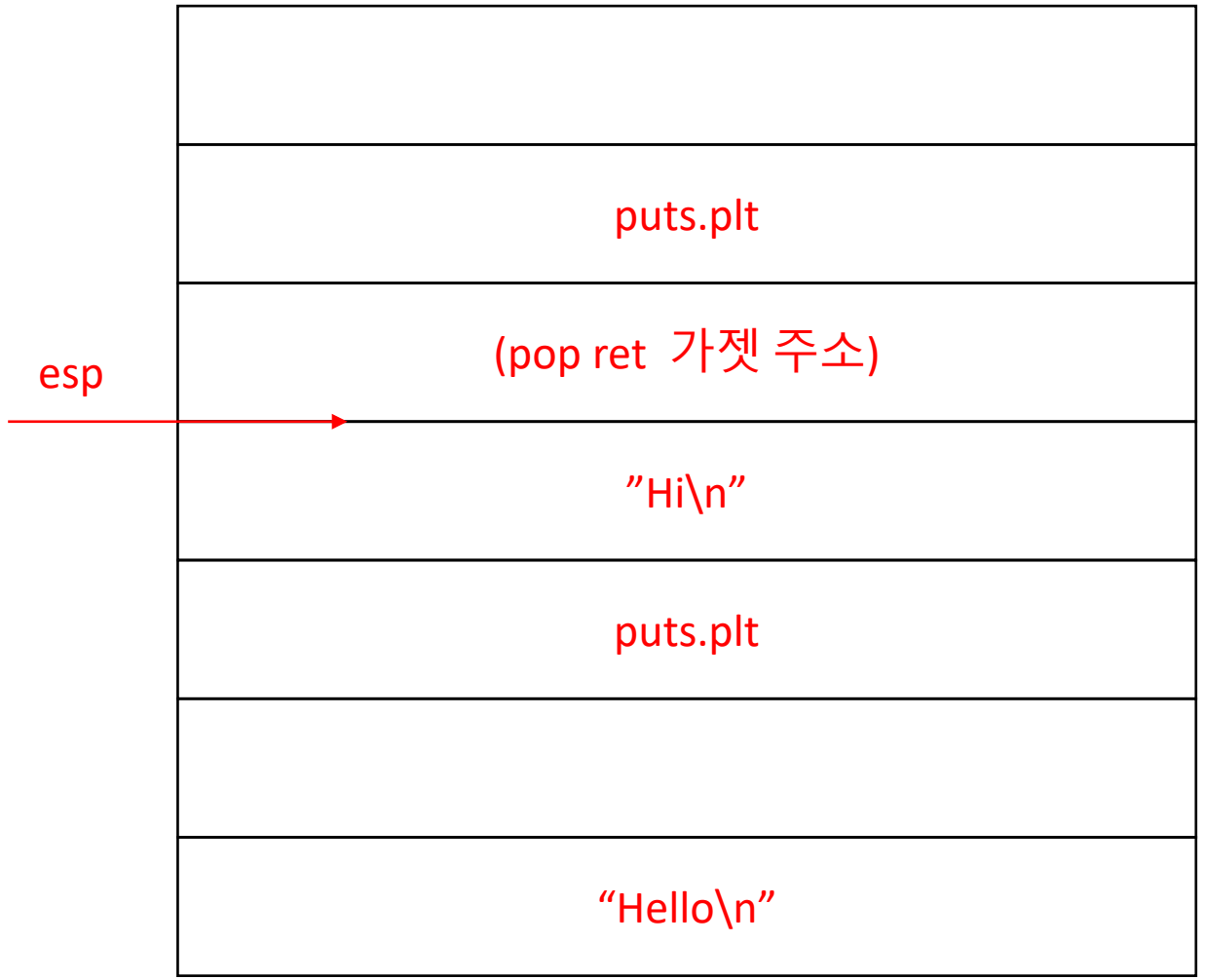
<출력>

Hi



<출력>

Hi




eip (pop ret 가젯)

pop

ret

<출력>

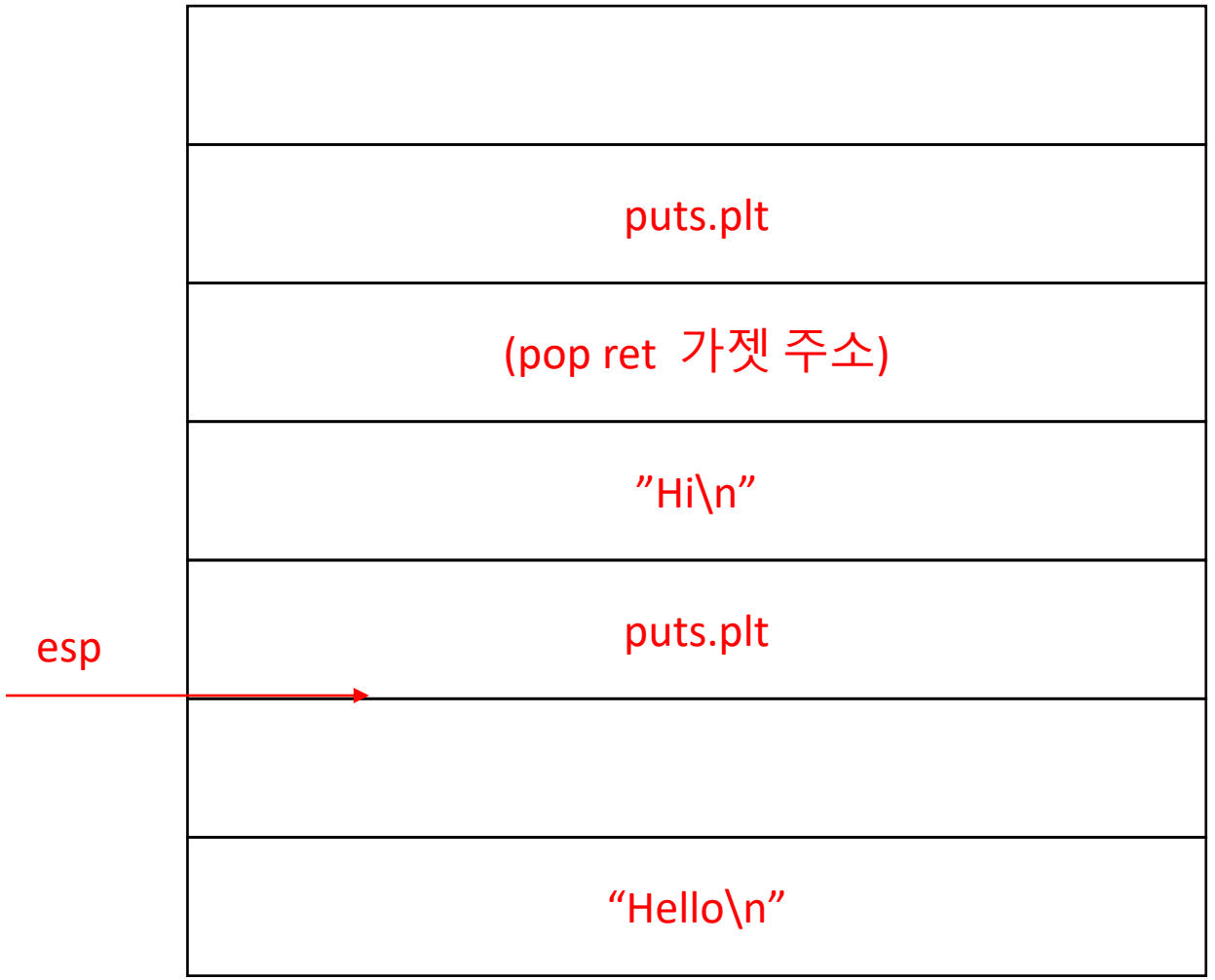
Hi


eip

(pop ret 가젯)
pop
ret

<출력>

Hi



<출력>

# RTL chaining