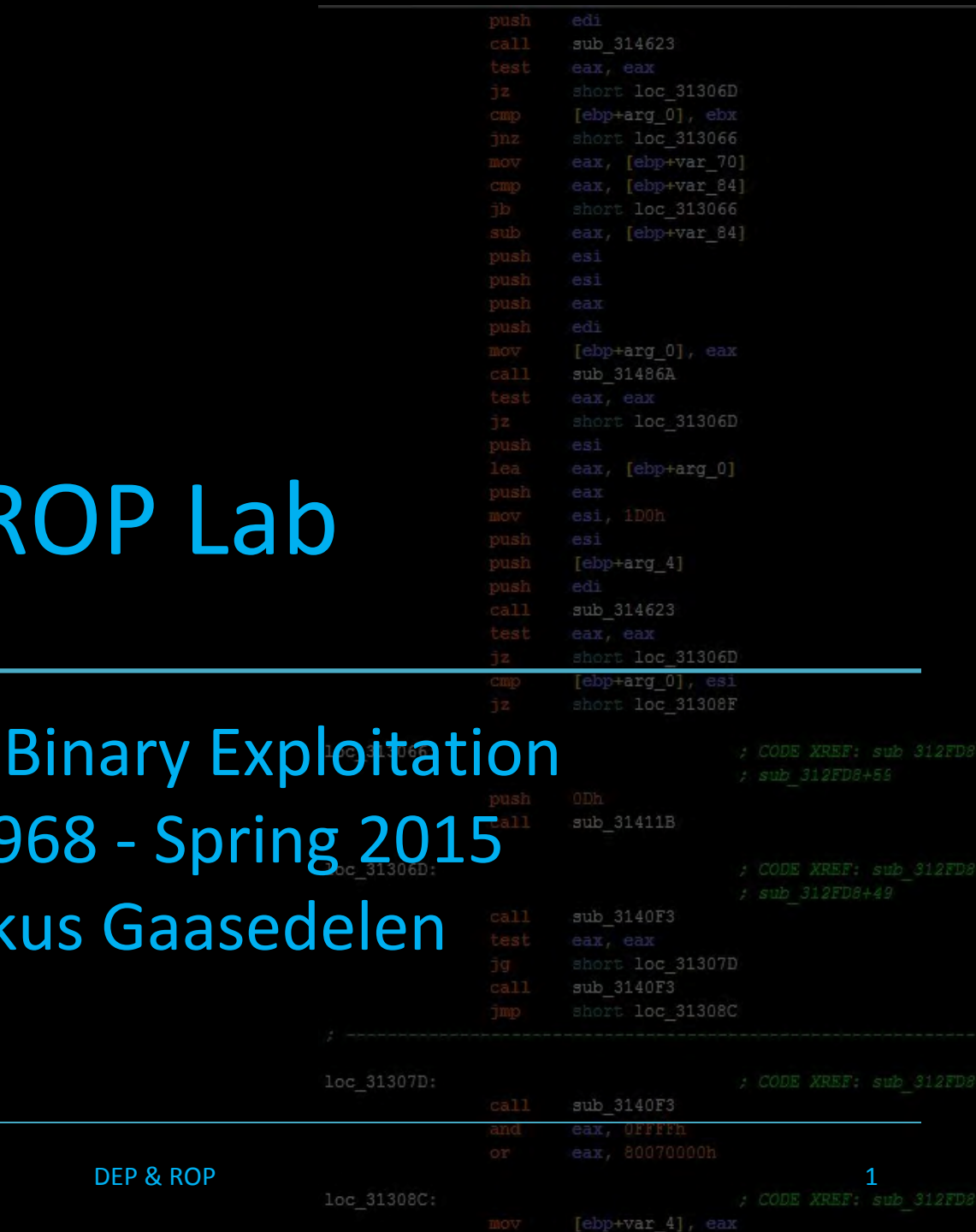
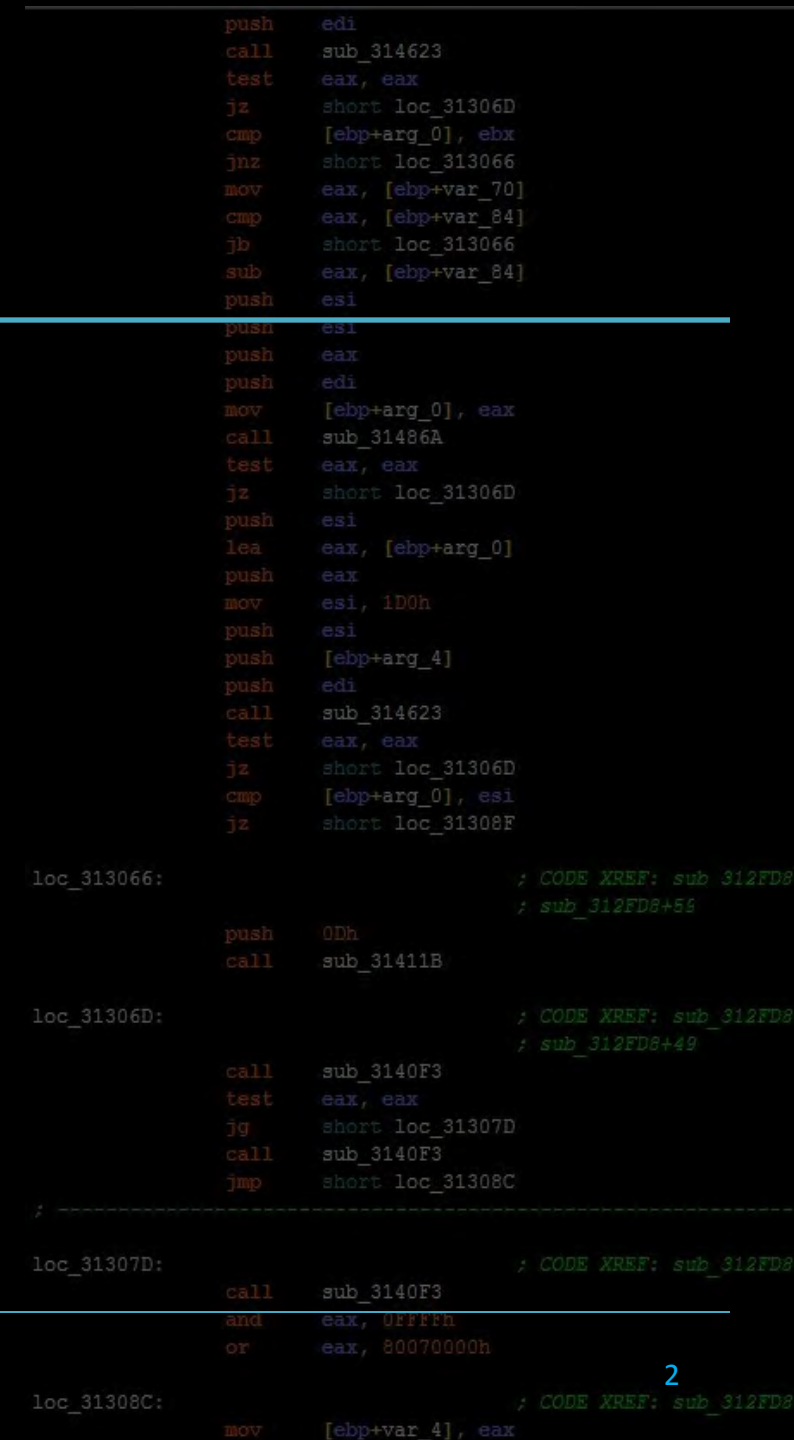
ROP Lab

Modern Binary Exploitation
CSCI 4968 - Spring 2015
Markus Gaasedelen



Final notes on DEP

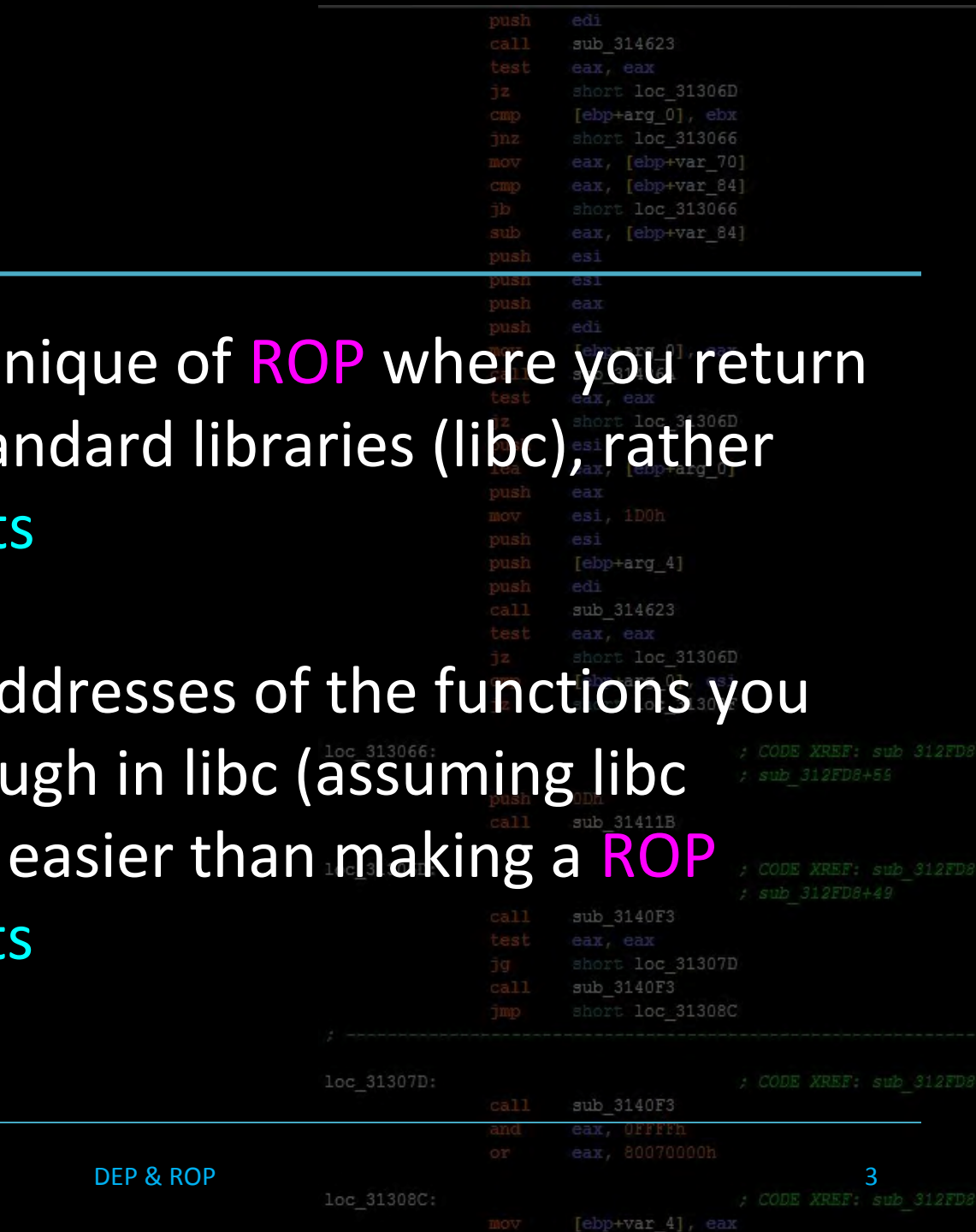
- One last concept
  - ret2libc



ret2libc

- ‘ret2libc’ is a technique of ROP where you return to functions in standard libraries (libc), rather than using gadgets

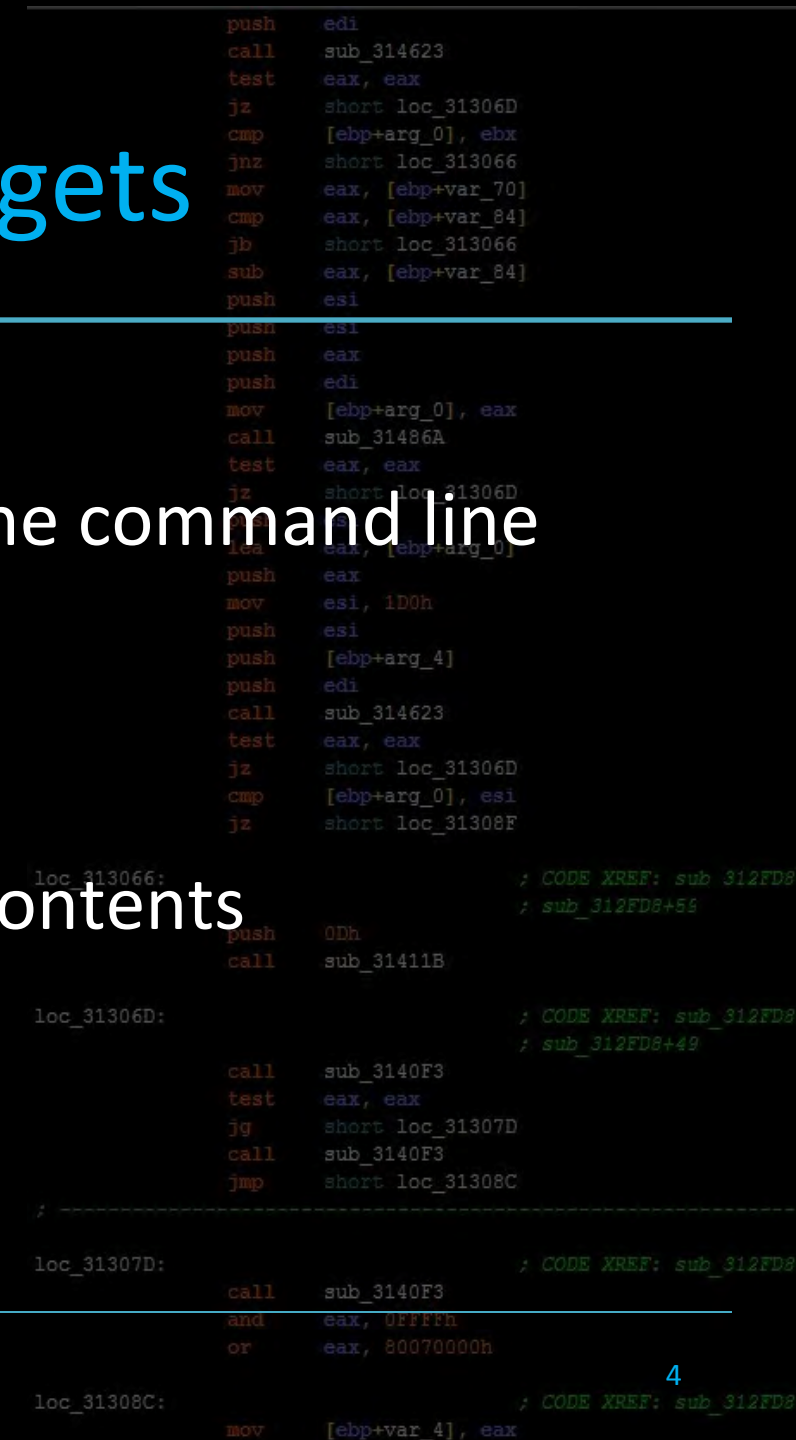
- If you know the addresses of the functions you want to ROP through in libc (assuming libc exists), ret2libc is easier than making a ROP chain with gadgets



Common ret2libc Targets

- system()
  - Executes something on the command line
  - system("cat flag.txt");

- (f) open() / read() / write()
  - Open/Read/Write a file contents



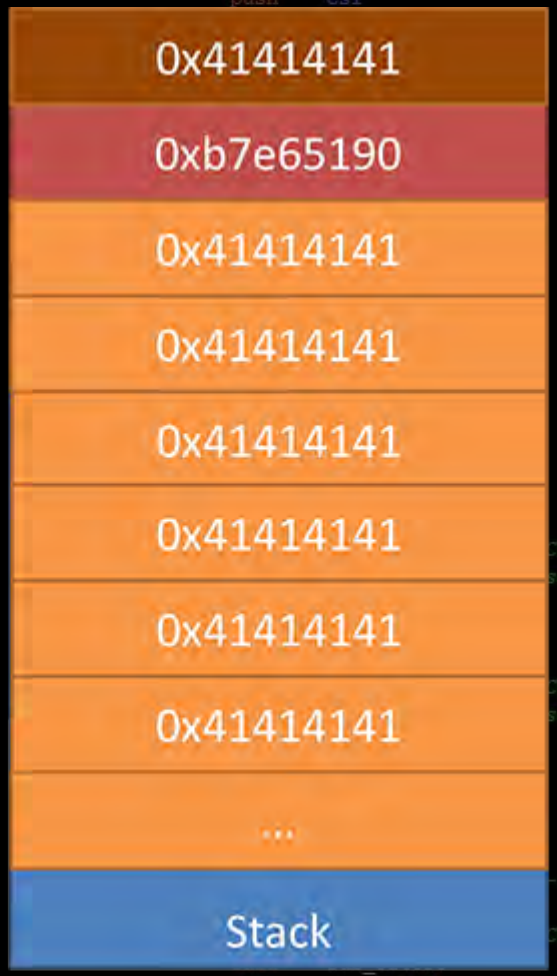
ret2libc example

```
0x08045430:   ret
```

EIP

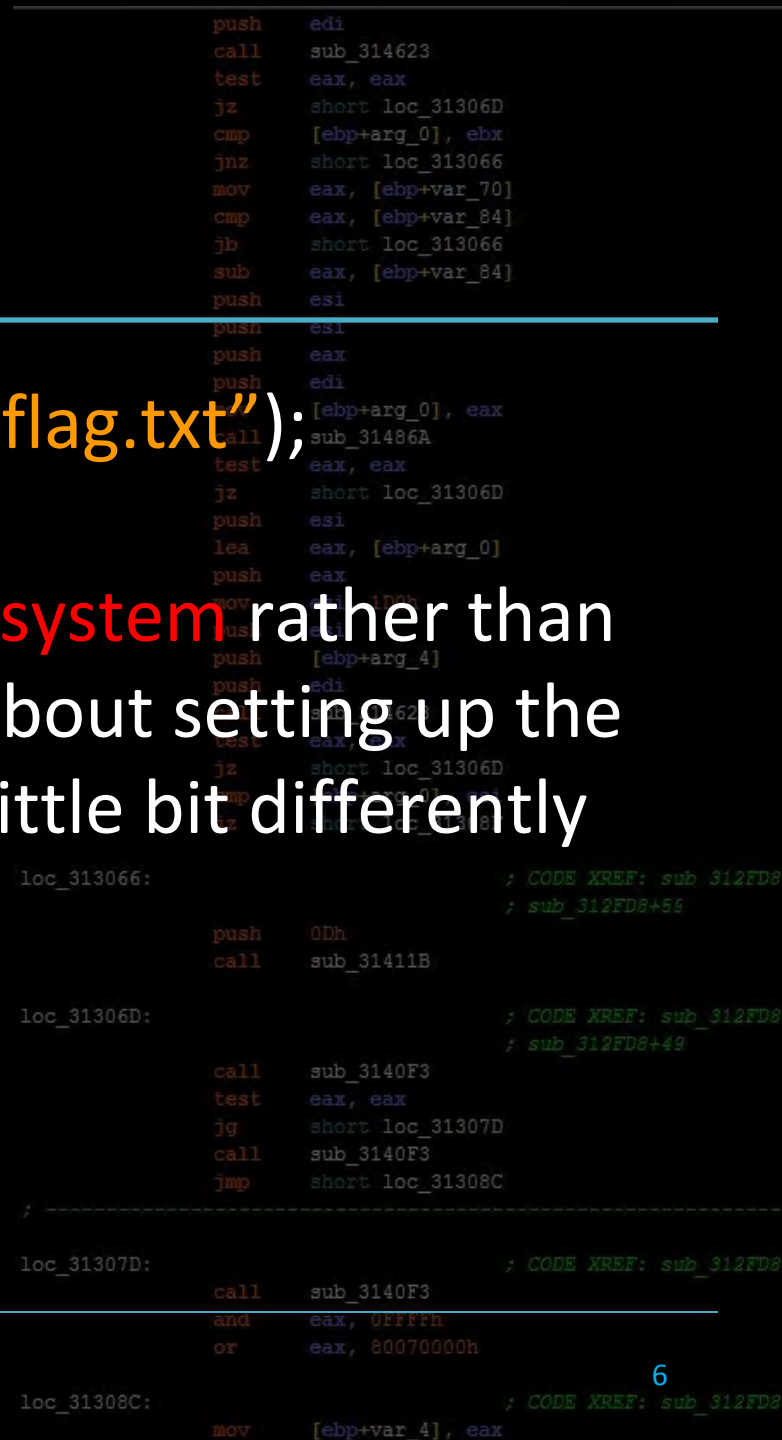
```
.....................................
system()
0xb7e65190:   push    ebx
0xb7e65191:   sub     esp, 8
0xb7e65194:   mov     eax, DWORD PTR
  [esp+0x10]
...
```

system() --->



Returning to System

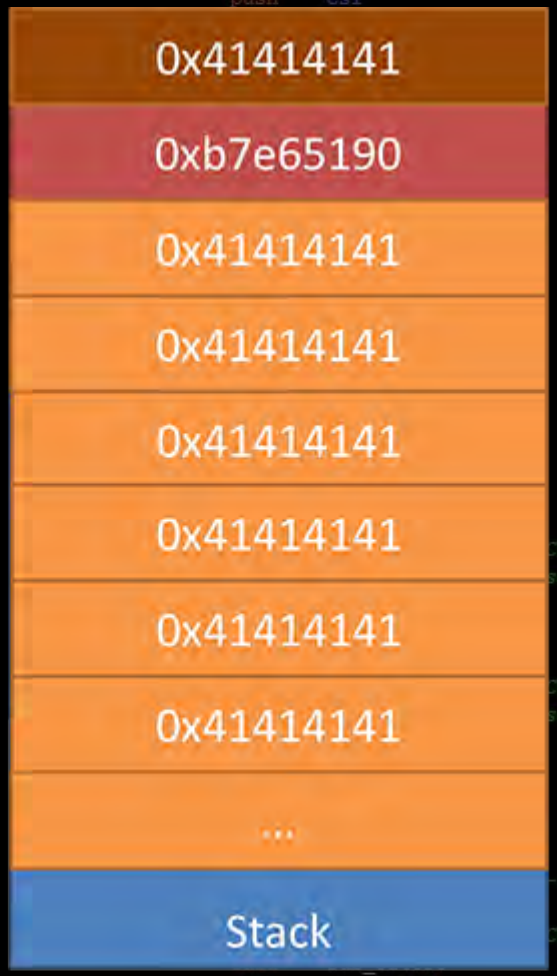
- We want to call system(“cat flag.txt”);
- Because we are ROPing into system rather than calling it, you have to think about setting up the stack (to pass arguments) a little bit differently



ret2libc example

```
0x08045430:   ret        <-- EIP
.....................................
system()
0xb7e65190:   push    ebx
0xb7e65191:   sub     esp, 8
0xb7e65194:   mov     eax, DWORD PTR
  [esp+0x10]
...
```

system() --->

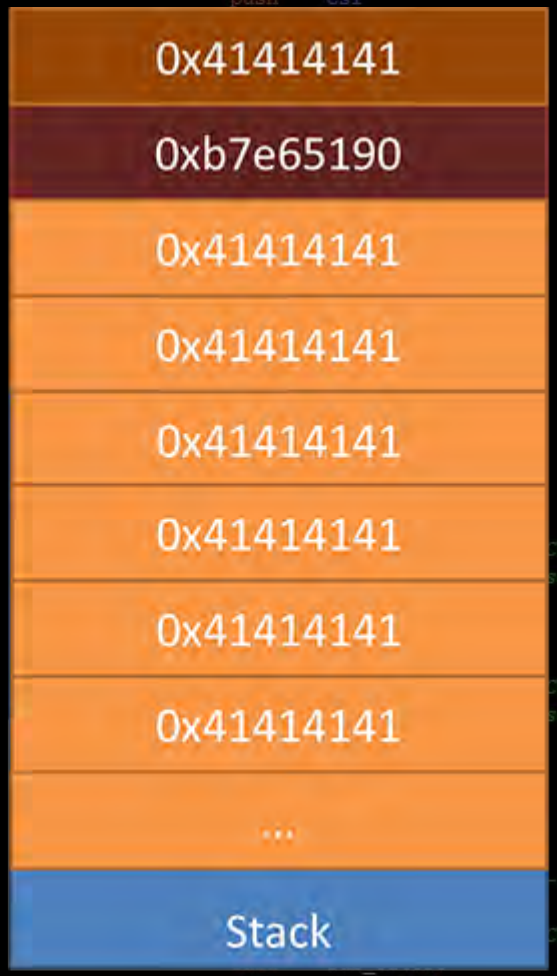


ret2libc example

```
system() --->
????? --->
????? --->
```

```
0x08045430:  ret

.....................................
system()
0xb7e65190:  push   ebx
0xb7e65191:  sub    esp, 8
0xb7e65194:  mov    eax, DWORD PTR
  [esp+0x10]
...
```

EIP



ret2libc example

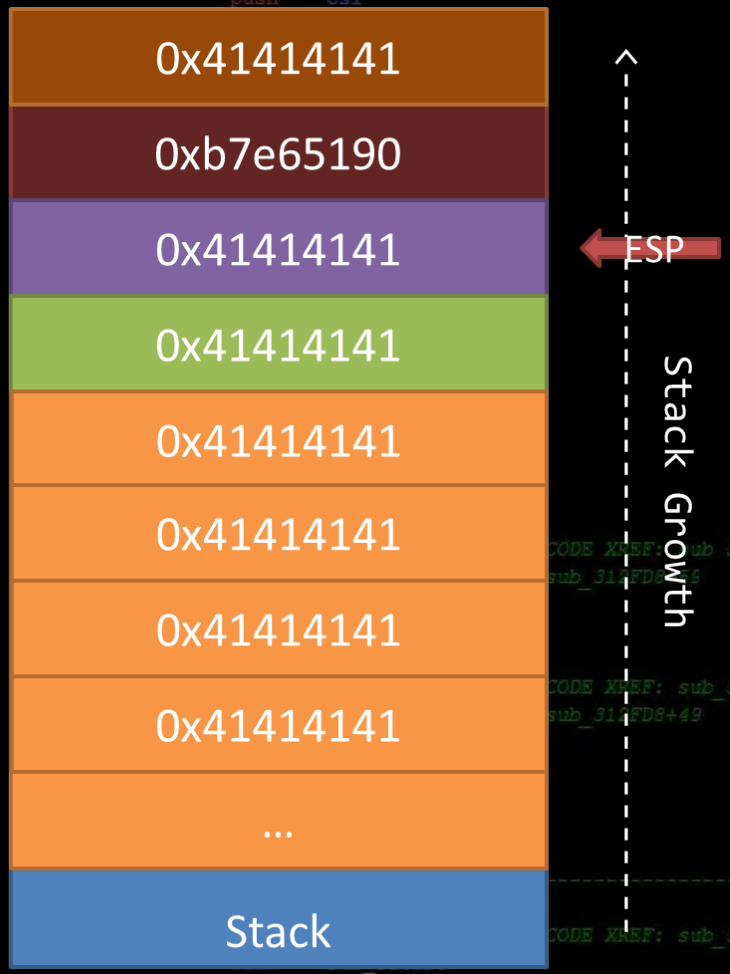
```
0x08045430:  ret

.....................................
system()
0xb7e65190:  push   ebx
0xb7e65191:  sub    esp, 8
0xb7e65194:  mov    eax, DWORD PTR
  [esp+0x10]
...
```

EIP

system() --->

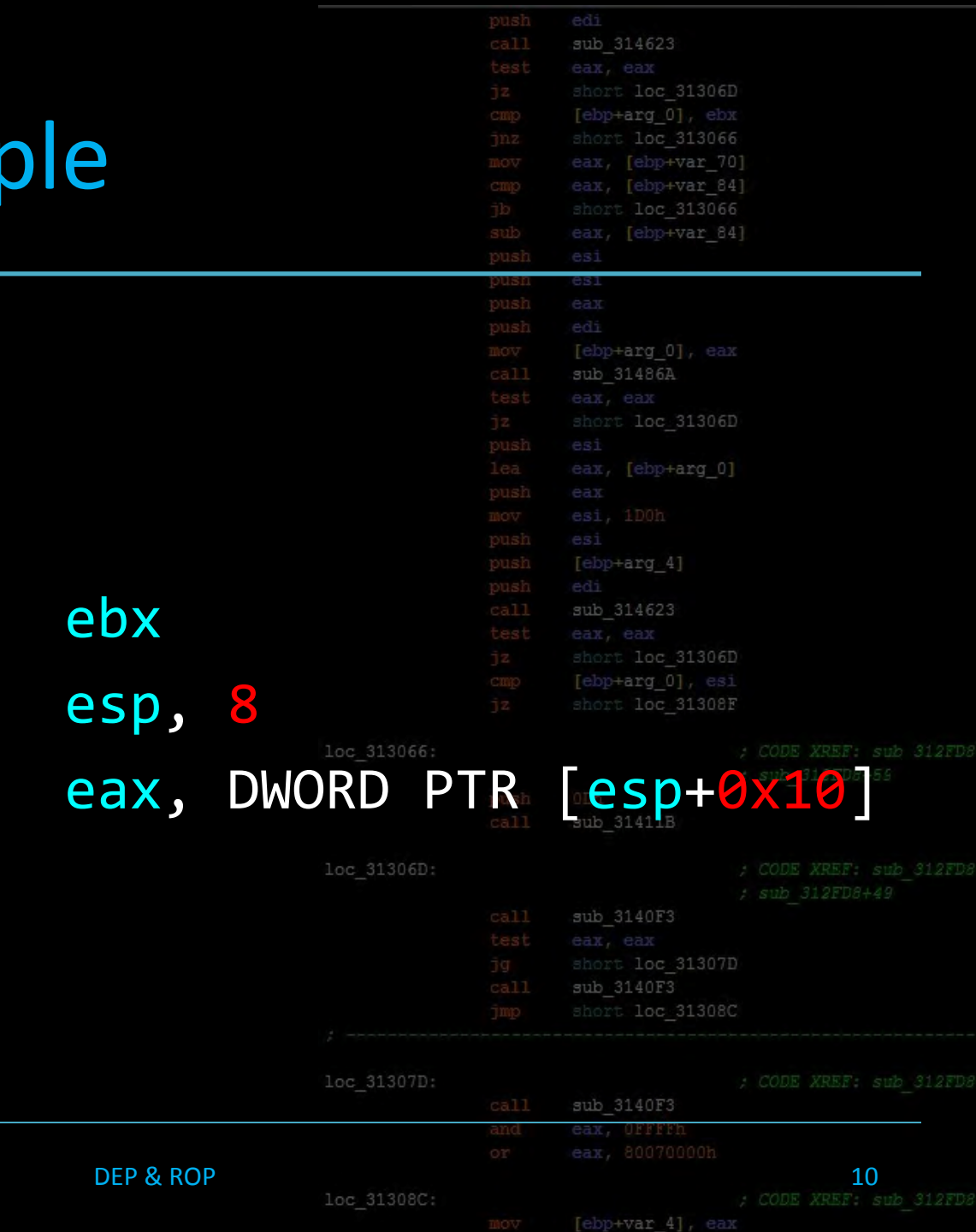
ret address --->

first arg --->



ret2libc example

```
system()
0xb7e65190:   push    ebx
0xb7e65191:   sub     esp, 8
0xb7e65194:   mov     eax, DWORD PTR [esp+0x10]
...
```



ret2libc example

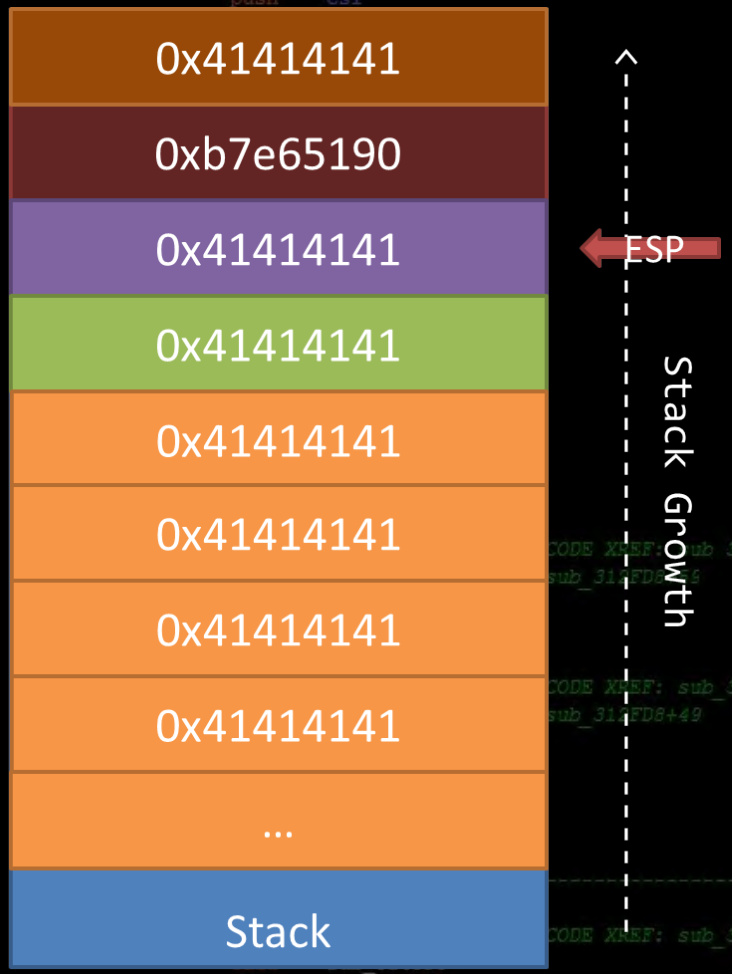
```
0x08045430:  ret

.....................................
system()
0xb7e65190:  push   ebx
0xb7e65191:  sub    esp, 8
0xb7e65194:  mov    eax, DWORD PTR
  [esp+0x10]
...
```

EIP

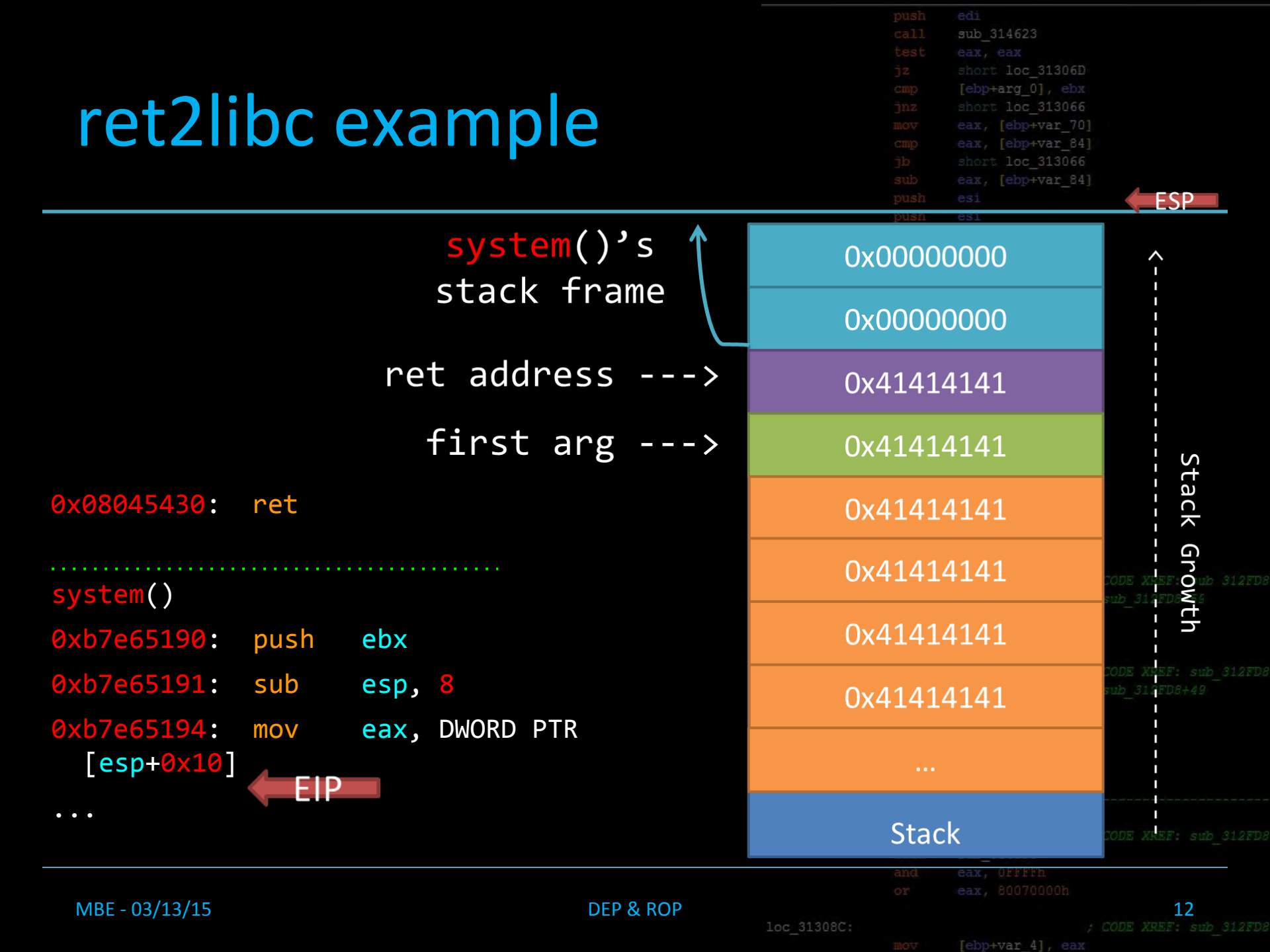
system() --->

ret address --->

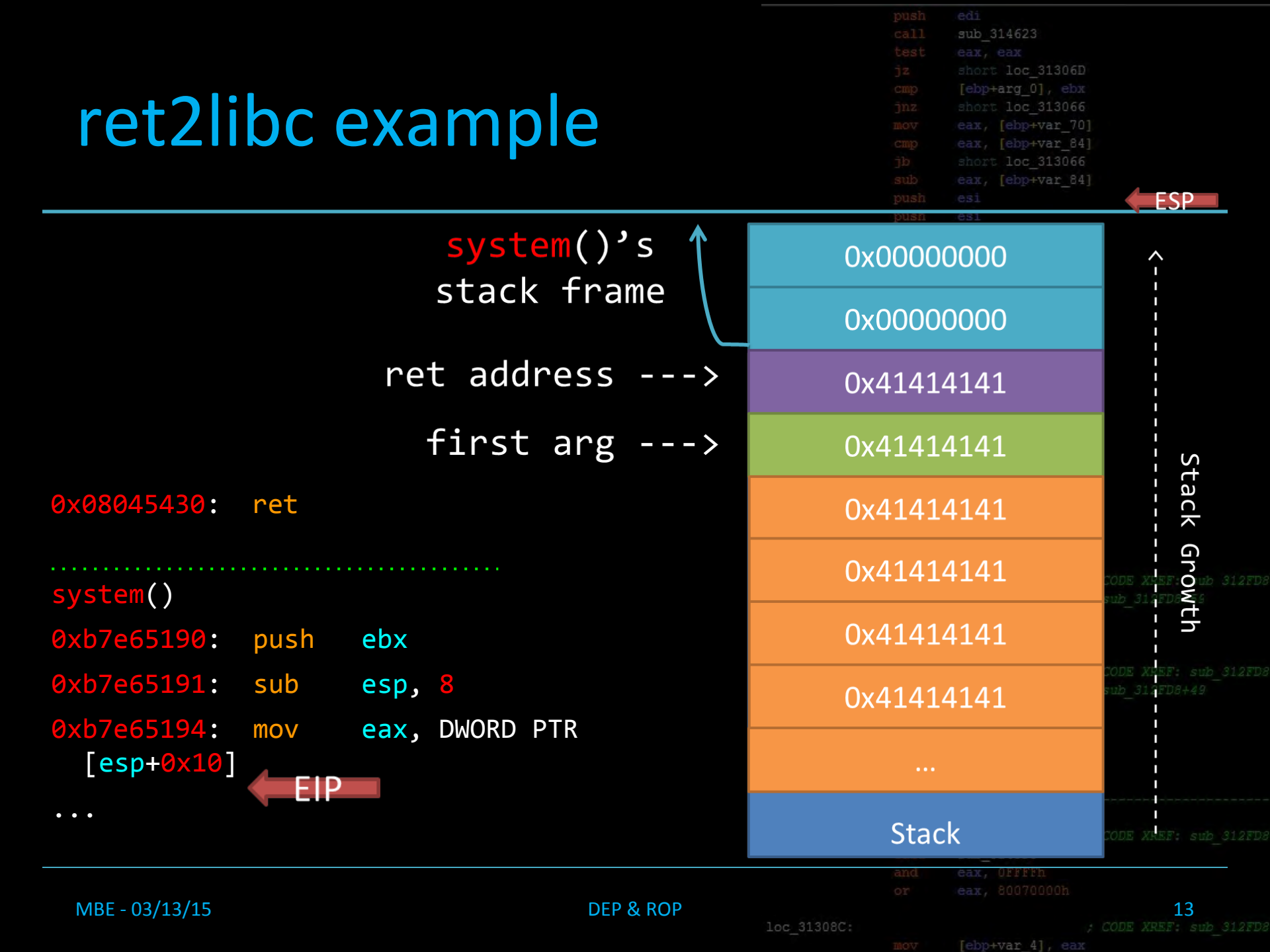
first arg --->



ret2libc example



ret2libc example



REWIND

```
                push    edi
                call    sub_314623
                test    eax, eax
                jz      short loc_31306D
                cmp     [ebp+arg_0], ebx
                jnz     short loc_313066
                mov     eax, [ebp+var_70]
                cmp     eax, [ebp+var_84]
                jb      short loc_313066
                sub     eax, [ebp+var_84]
                push    esi
                push    esi
                push    eax
                push    edi
                mov     [ebp+arg_0], eax
                call    sub_31486A
                test    eax, eax
                jz      short loc_31306D
                push    esi
                lea     eax, [ebp+arg_0]
                push    eax
                mov     esi, 1D0h
                push    esi
                push    [ebp+arg_4]
                push    edi
                call    sub_314623
                test    eax, eax
                jz      short loc_31306D
                cmp     [ebp+arg_0], esi
                jz      short loc_31308F

loc_313066:                             ; CODE XREF: sub_312FD8
                                        ; sub_312FD8+59
                push    0Dh
                call    sub_31411B

loc_31306D:                             ; CODE XREF: sub_312FD8
                                        ; sub_312FD8+49
                call    sub_3140F3
                test    eax, eax
                jg      short loc_31307D
                call    sub_3140F3
                jmp     short loc_31308C
; ---------------------------------------------------------------------------

loc_31307D:                             ; CODE XREF: sub_312FD8
                call    sub_3140F3
                and     eax, 0FFFFh
                or      eax, 80070000h

loc_31308C:                             ; CODE XREF: sub_312FD8
                mov     [ebp+var_4], eax
```

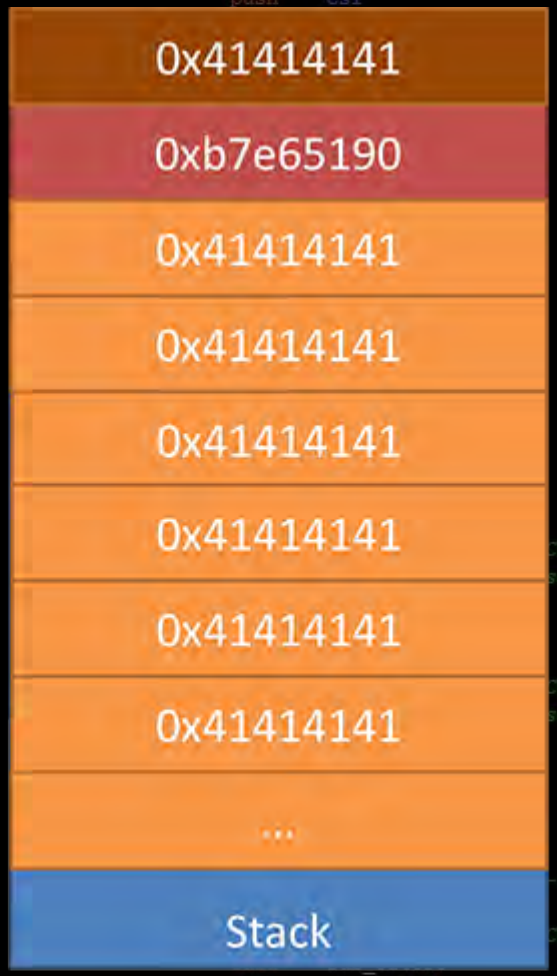
ret2libc example

```
0x08045430:   ret
```

EIP

```
.....................................
system()
0xb7e65190:   push    ebx
0xb7e65191:   sub     esp, 8
0xb7e65194:   mov     eax, DWORD PTR
  [esp+0x10]
...
```

system() --->



ret2libc example

```
0x08045430:  ret
```

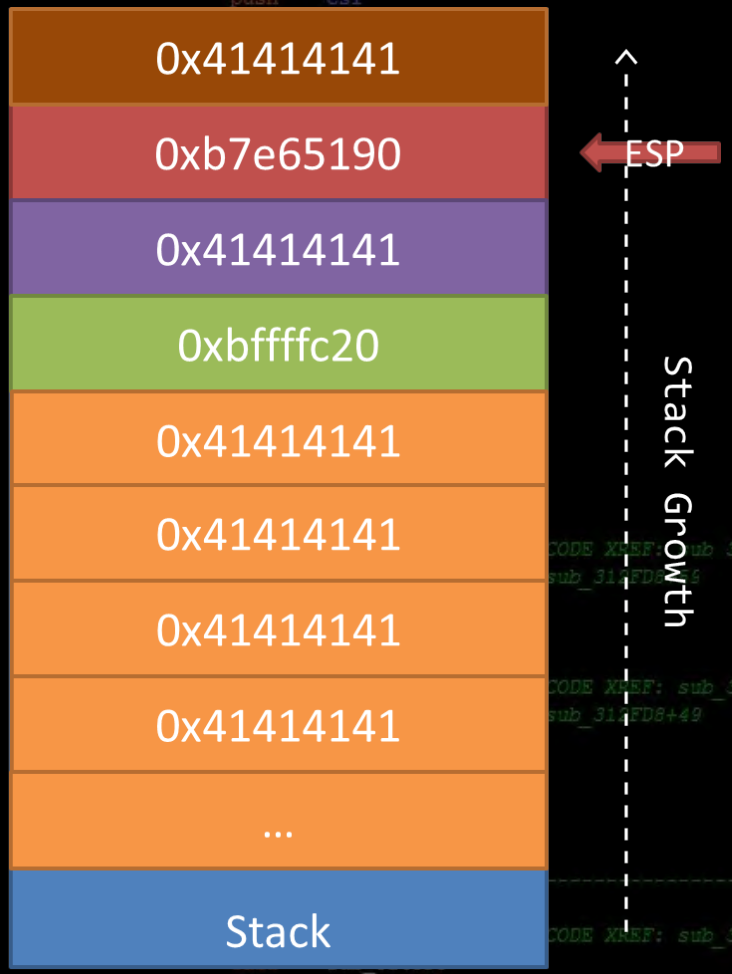
EIP

```
.....................................
system()
0xb7e65190:  push   ebx
0xb7e65191:  sub    esp, 8
0xb7e65194:  mov    eax, DWORD PTR
  [esp+0x10]
...
```

system() --->

ret address --->

first arg ---> "cat flag.txt"



ret2libc example

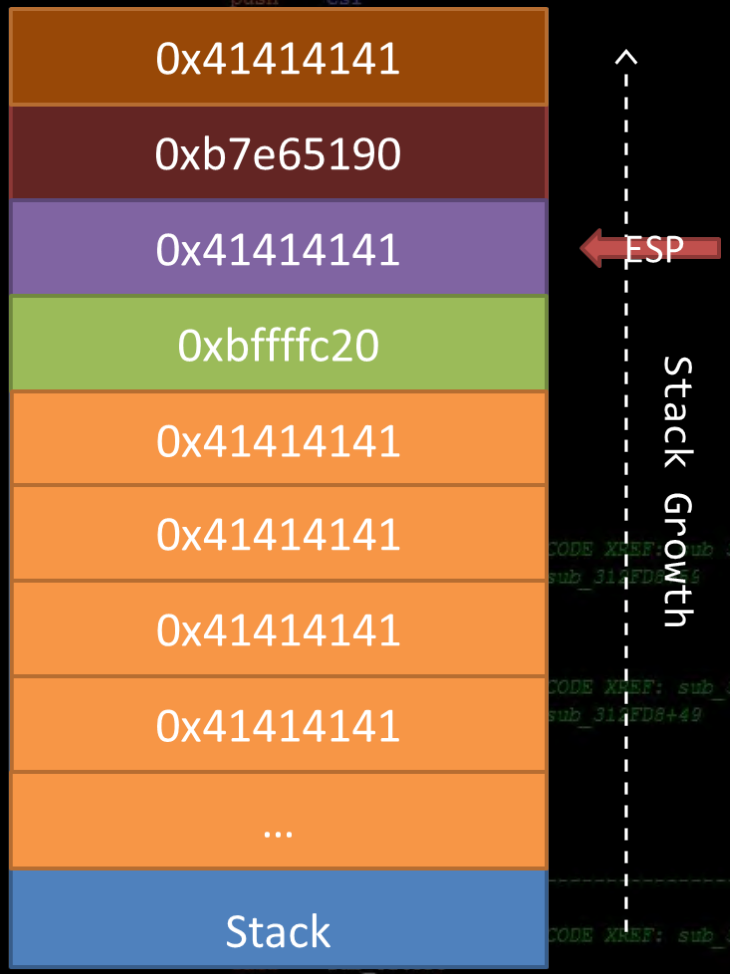
```
0x08045430:   ret

.....................................
system()
0xb7e65190:   push    ebx
0xb7e65191:   sub     esp, 8
0xb7e65194:   mov     eax, DWORD PTR
  [esp+0x10]
...
```

EIP

system() --->

ret address --->

first arg --->
"cat flag.txt"



ret2libc example

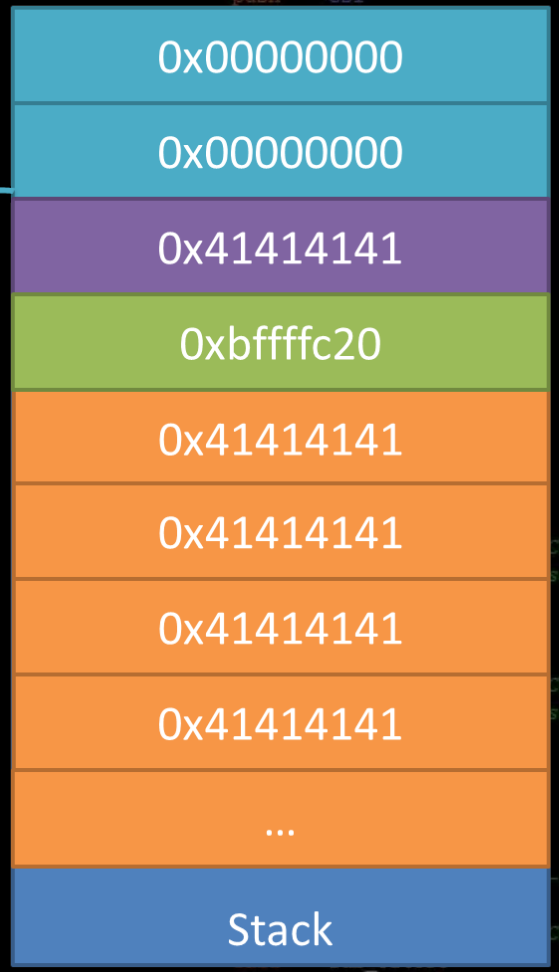
```
0x08045430:   ret

.....................................
system()
0xb7e65190:   push    ebx
0xb7e65191:   sub     esp, 8
0xb7e65194:   mov     eax, DWORD PTR
  [esp+0x10]
...
```

EIP

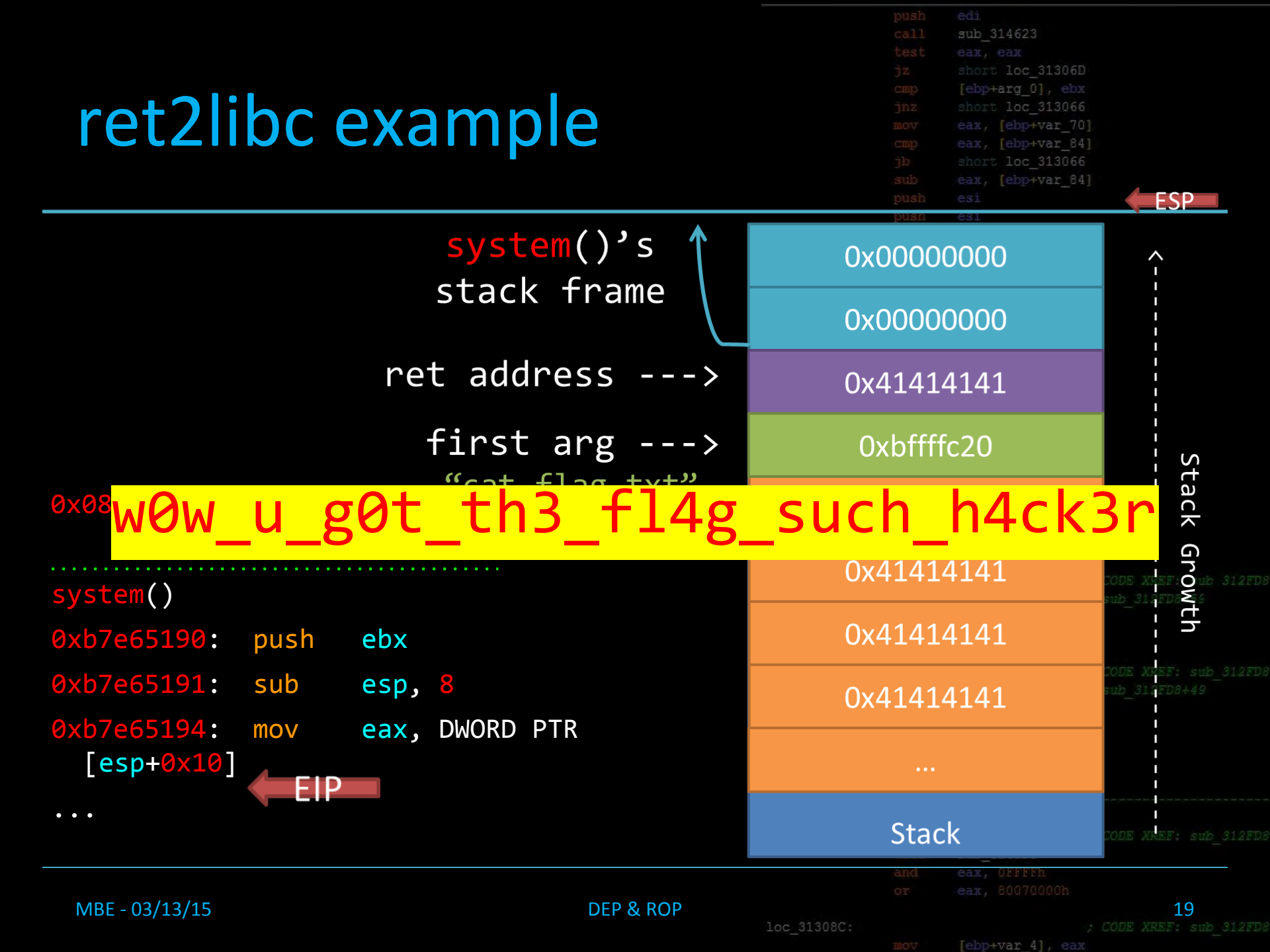
system()'s stack frame

ret address --->

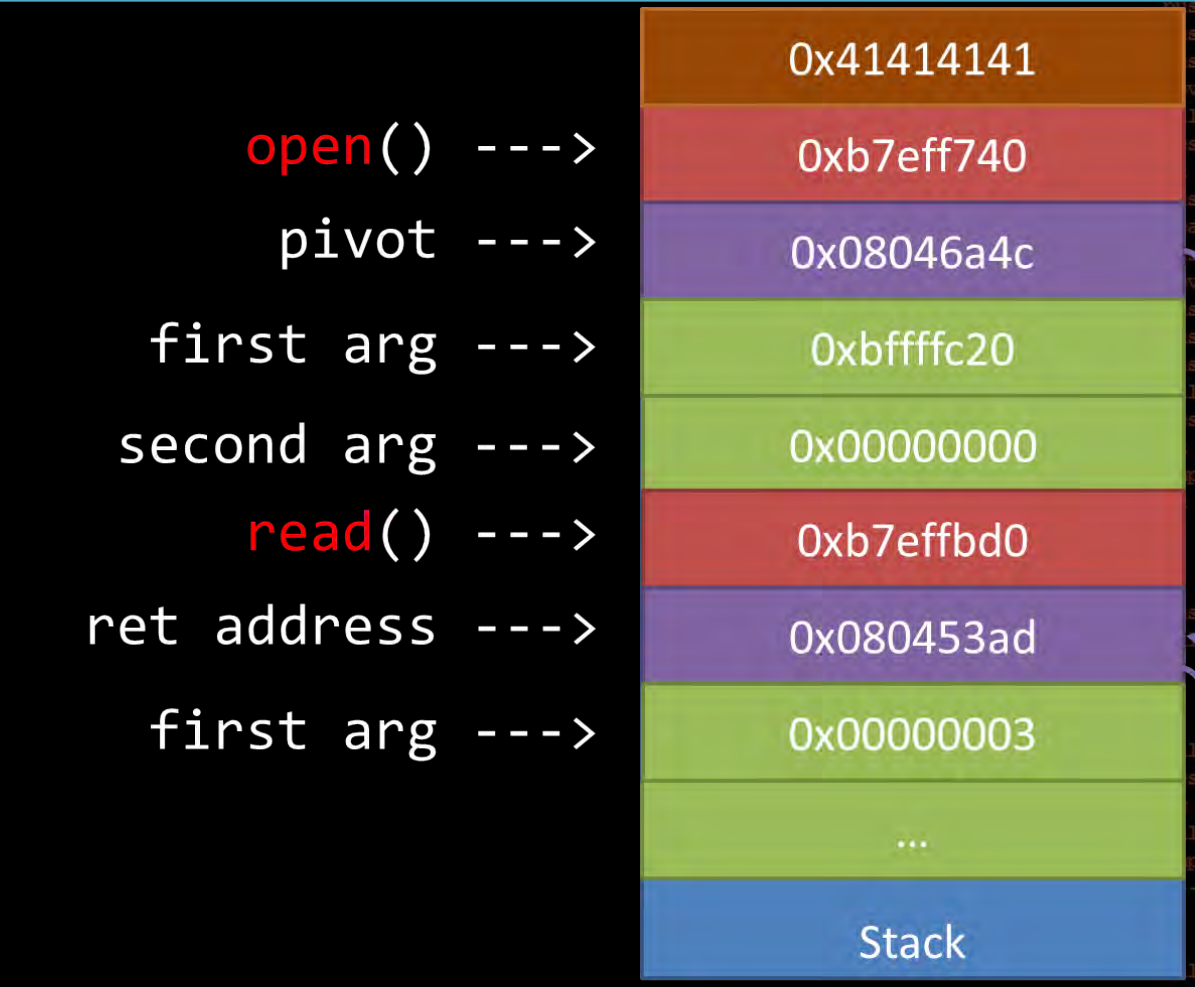
first arg ---> "cat flag.txt"



ret2libc example



Chaining Calls



Lab 5

- lab5C - ret2libc
- lab5B - Basic ROP
- lab5A - More involved ROP