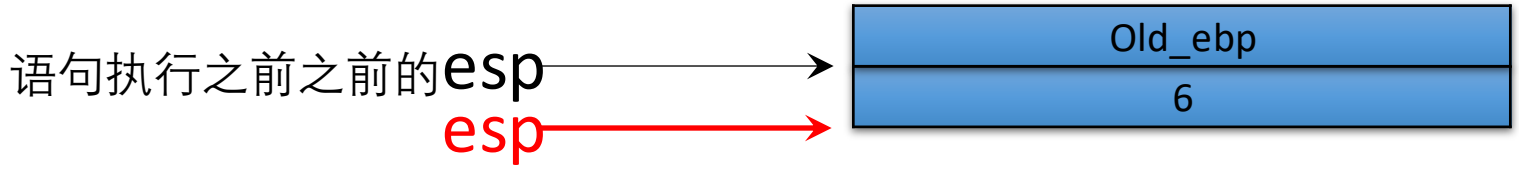
```
push ebp
mov  ebp,esp
push 6
push 5
call AFunc
add esp,8
```

PS: 执行语句之前的EBP在此栈空间的更高处

```
push ebp
mov  ebp,esp
push 6
push 5
call AFunc
add esp,8
```



```
push ebp
mov  ebp,esp
push 6
push 5
call AFunc
add esp,8
```

语句执行之前之前的esp

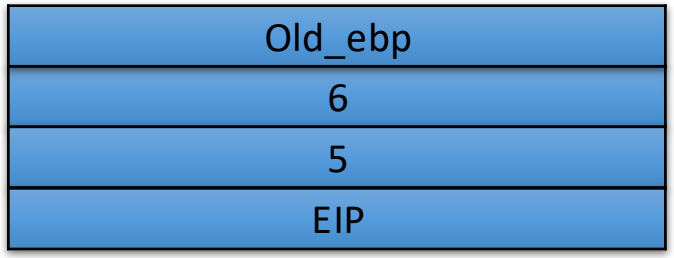
esp

栈底

```
push ebp
mov  ebp,esp
push 6
push 5
call AFunc
add esp,8
```

语句执行之前之前的esp

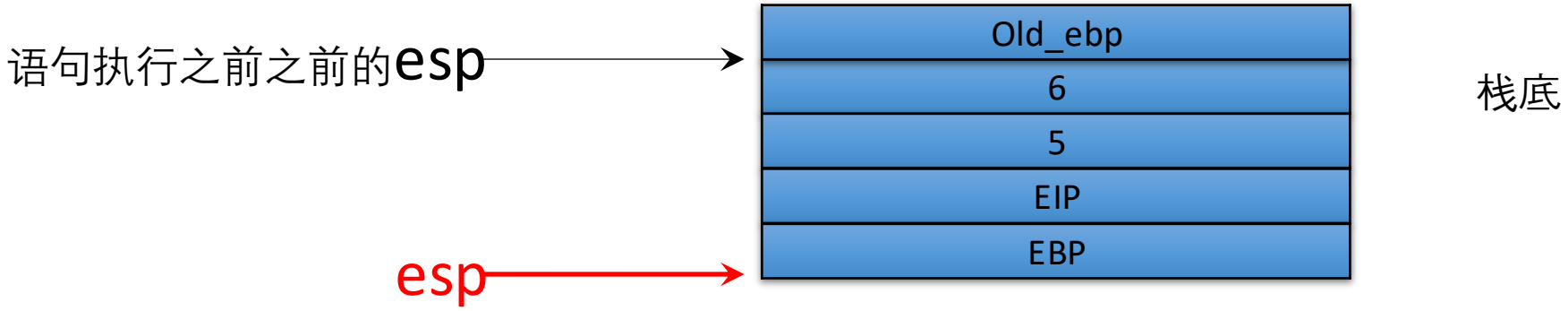
esp



栈底

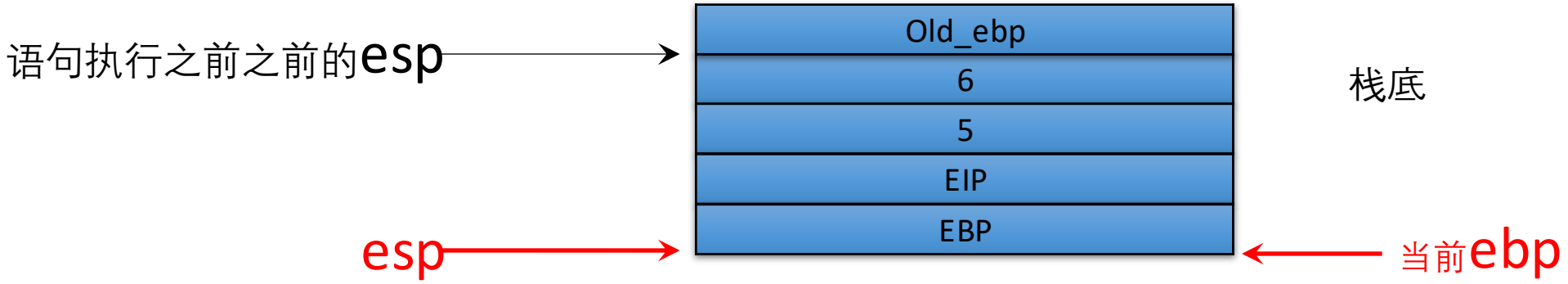
AFunc(5,6);
push 6
push 5
call AFunc
add esp,8

AFunc
push ebp
mov ebp,esp
sub esp,0x10
mov DWORD PTR [ebp-0x4],0x3
mov DWORD PTR [ebp-0x8],0x4
mov eax, DWORD PTR [ebp+0x8]
mov DWORD PTR [ebp-0x4],eax
mov eax,DWORD PTR [ebp+0xc]
mov DWORD PTR [ebp-0x8],eax
push DWORD PTR [ebp-0x8]
push DWORD PTR [ebp-0x4]
call _BFunc
add esp,0x8
mov eax,0x8
leave
ret



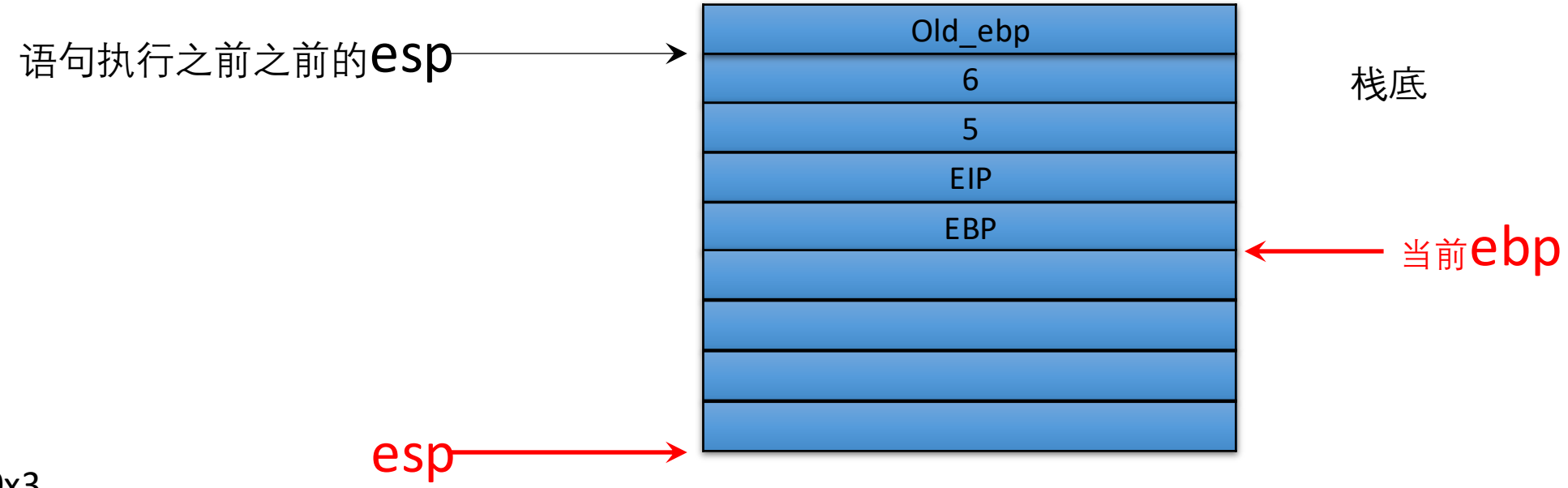
AFunc(5,6);
push 6
push 5
call AFunc
add esp,8

AFunc
push ebp
mov ebp,esp
sub esp,0x10
mov DWORD PTR [ebp-0x4],0x3
mov DWORD PTR [ebp-0x8],0x4
mov eax, DWORD PTR [ebp+0x8]
mov DWORD PTR [ebp-0x4],eax
mov eax,DWORD PTR [ebp+0xc]
mov DWORD PTR [ebp-0x8],eax
push DWORD PTR [ebp-0x8]
push DWORD PTR [ebp-0x4]
call _BFunc
add esp,0x8
mov eax,0x8
leave
ret



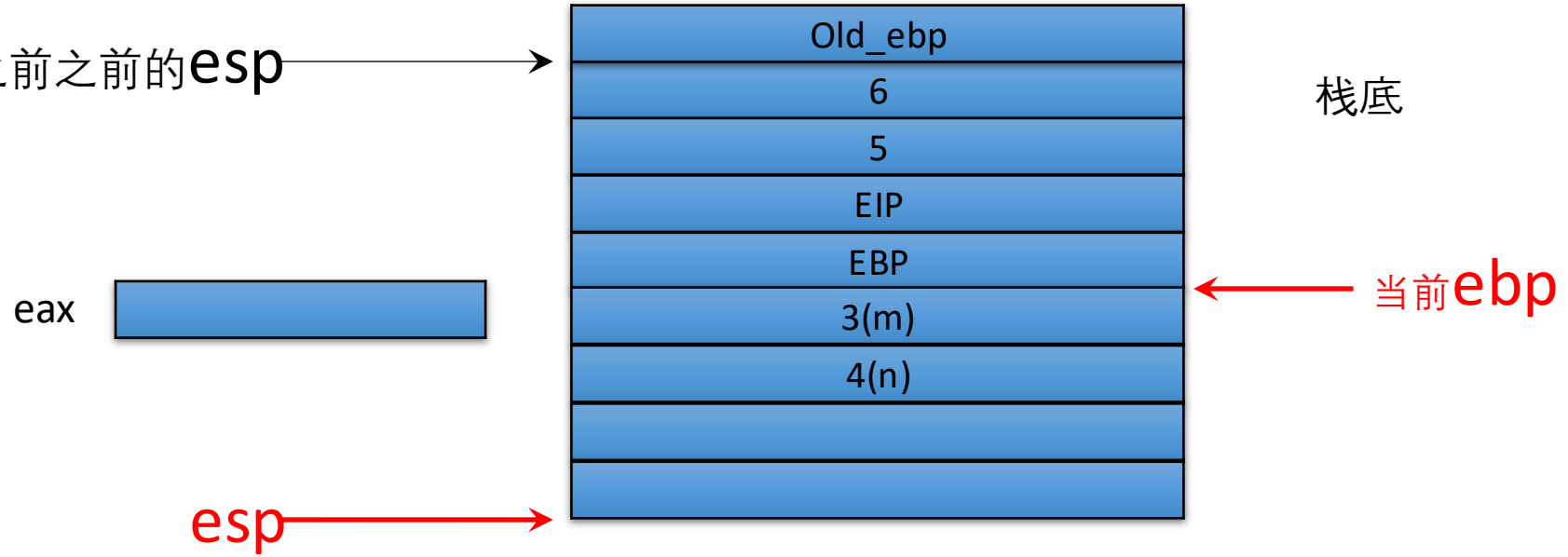
```
AFunc(5,6);
push 6
push 5
call AFunc
add esp,8

AFunc
push ebp
mov ebp,esp
sub esp,0x10
mov DWORD PTR [ebp-0x4],0x3
mov DWORD PTR [ebp-0x8],0x4
mov eax, DWORD PTR [ebp+0x8]
mov DWORD PTR [ebp-0x4],eax
mov eax,DWORD PTR [ebp+0xc]
mov DWORD PTR [ebp-0x8],eax
push DWORD PTR [ebp-0x8]
push DWORD PTR [ebp-0x4]
call _BFunc
add esp,0x8
mov eax,0x8
leave
ret
```



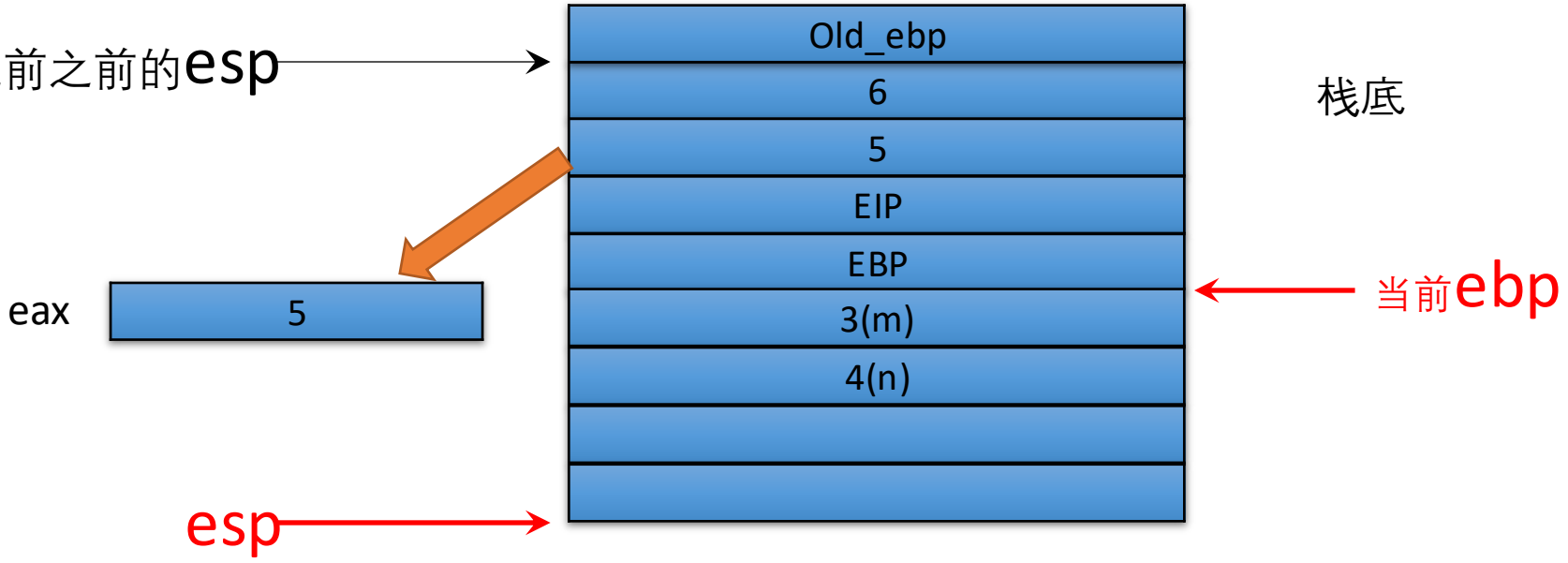
```
AFunc(5,6);
push 6
push 5
call AFunc
add esp,8

AFunc
push ebp
mov ebp,esp
sub esp,0x10
mov DWORD PTR [ebp-0x4],0x3
mov DWORD PTR [ebp-0x8],0x4
mov eax, DWORD PTR [ebp+0x8]
mov DWORD PTR [ebp-0x4],eax
mov eax,DWORD PTR [ebp+0xc]
mov DWORD PTR [ebp-0x8],eax
push DWORD PTR [ebp-0x8]
push DWORD PTR [ebp-0x4]
call _BFunc
add esp,0x8
mov eax,0x8
leave
ret
```



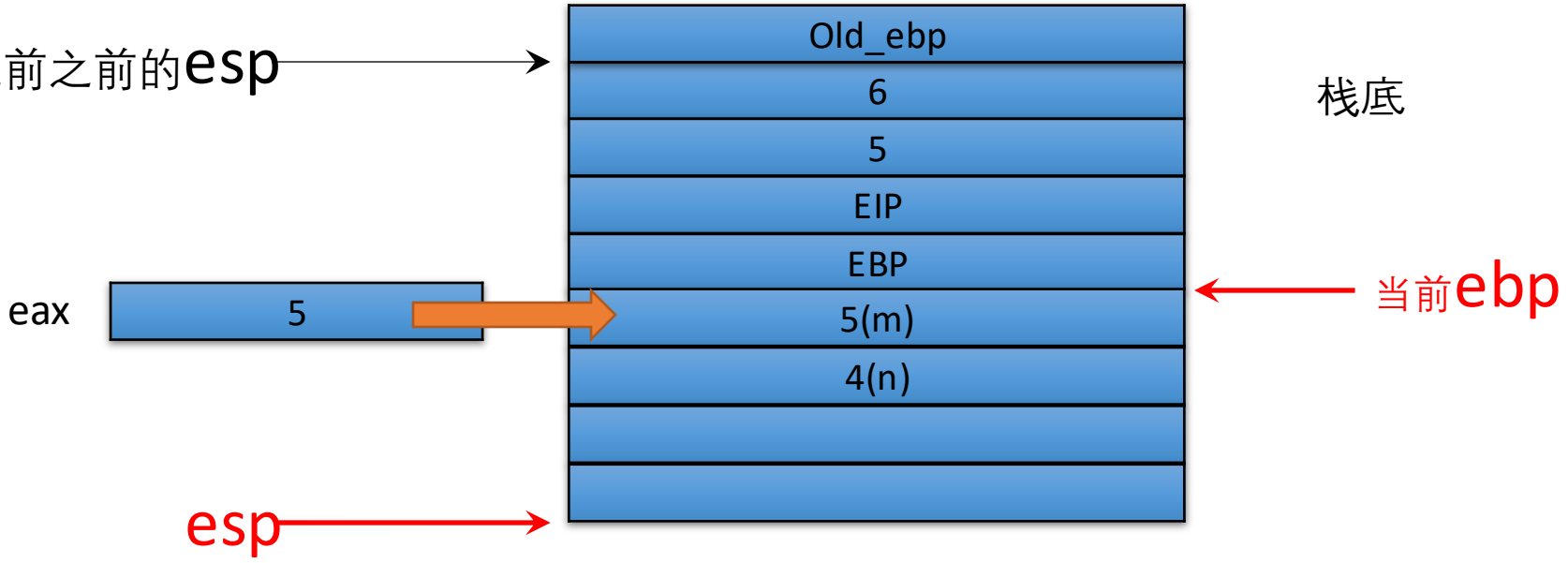
AFunc(5,6);
push 6
push 5
call AFunc
add esp,8

AFunc
push ebp
mov ebp,esp
sub esp,0x10
mov DWORD PTR [ebp-0x4],0x3
mov DWORD PTR [ebp-0x8],0x4
mov eax, DWORD PTR [ebp+0x8]
mov DWORD PTR [ebp-0x4],eax
mov eax,DWORD PTR [ebp+0xc]
mov DWORD PTR [ebp-0x8],eax
push DWORD PTR [ebp-0x8]
push DWORD PTR [ebp-0x4]
call _BFunc
add esp,0x8
mov eax,0x8
leave
ret



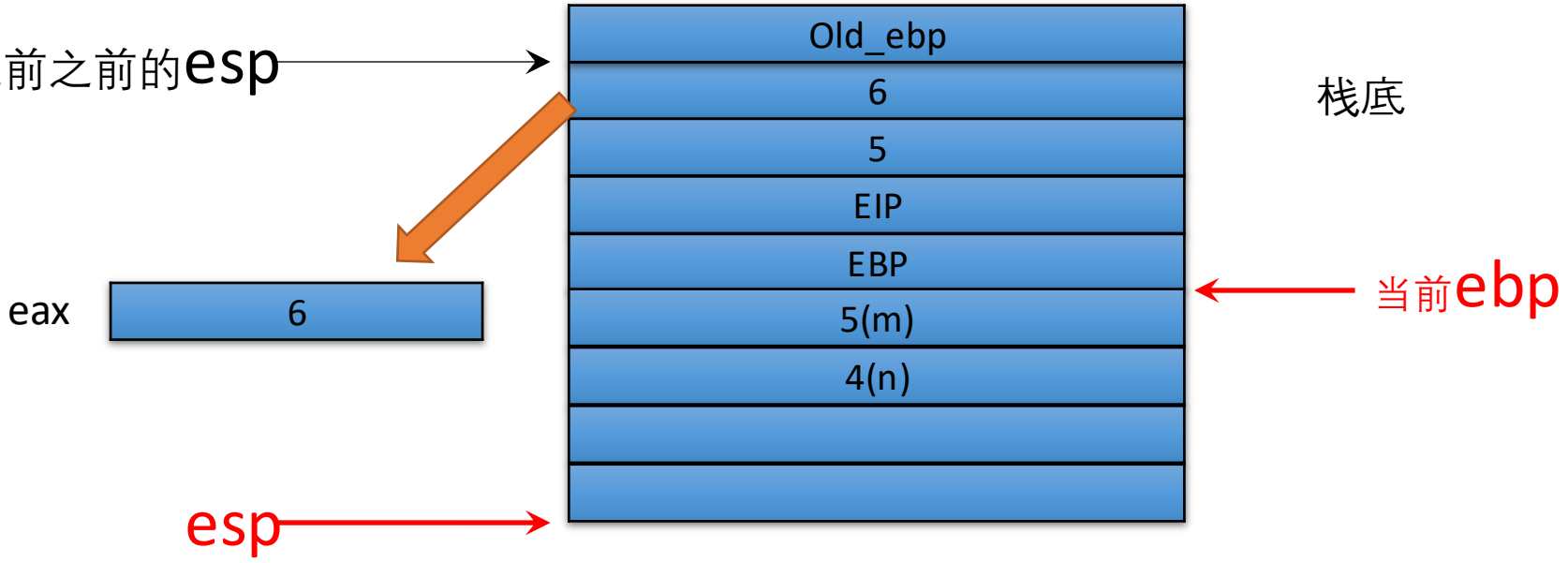
AFunc(5,6);
push 6
push 5
call AFunc
add esp,8

AFunc
push ebp
mov ebp,esp
sub esp,0x10
mov DWORD PTR [ebp-0x4],0x3
mov DWORD PTR [ebp-0x8],0x4
mov eax, DWORD PTR [ebp+0x8]
mov DWORD PTR [ebp-0x4],eax
mov eax,DWORD PTR [ebp+0xc]
mov DWORD PTR [ebp-0x8],eax
push DWORD PTR [ebp-0x8]
push DWORD PTR [ebp-0x4]
call _BFunc
add esp,0x8
mov eax,0x8
leave
ret



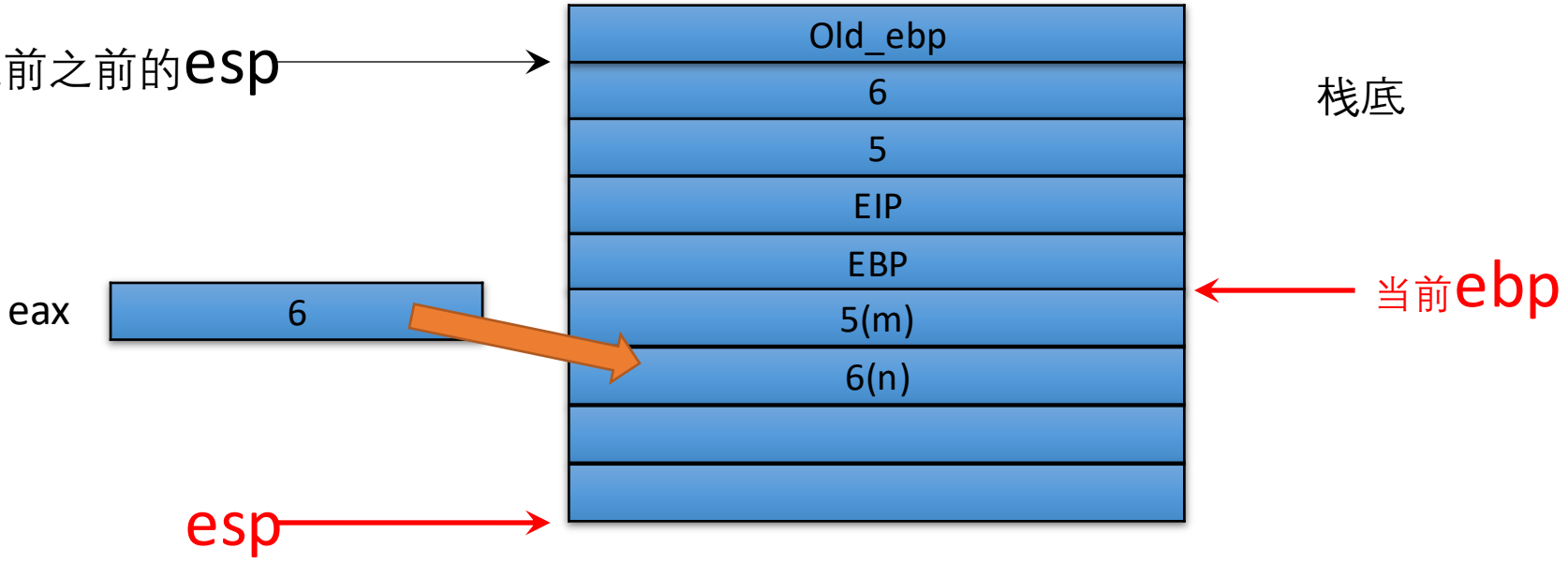
AFunc(5,6)；
push 6
push 5
call AFunc
add esp,8

AFunc
push ebp
mov ebp,esp
sub esp,0x10
mov DWORD PTR [ebp-0x4],0x3
mov DWORD PTR [ebp-0x8],0x4
mov eax, DWORD PTR [ebp+0x8]
mov DWORD PTR [ebp-0x4],eax
mov eax,DWORD PTR [ebp+0xc]
mov DWORD PTR [ebp-0x8],eax
push DWORD PTR [ebp-0x8]
push DWORD PTR [ebp-0x4]
call _BFunc
add esp,0x8
mov eax,0x8
leave
ret



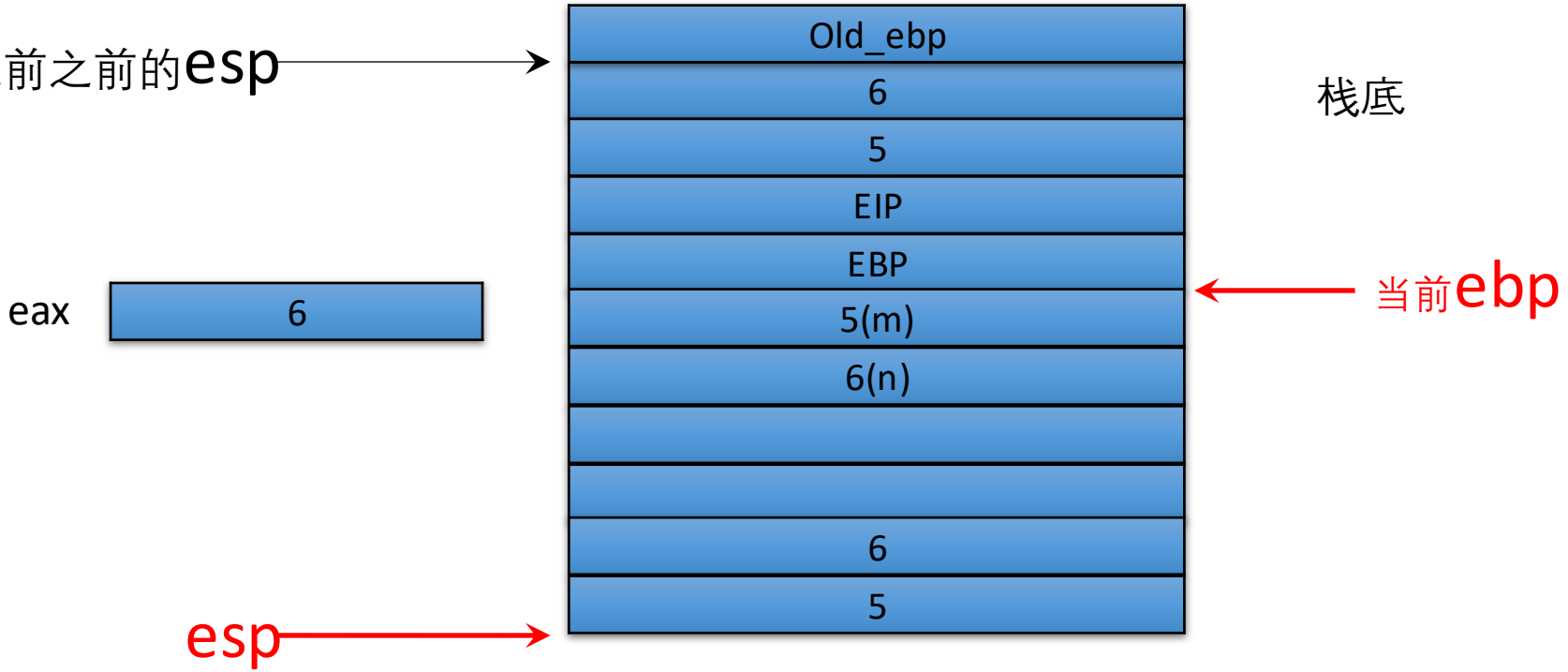
```
AFunc(5,6);
push 6
push 5
call AFunc
add esp,8

AFunc
push ebp
mov ebp,esp
sub esp,0x10
mov DWORD PTR [ebp-0x4],0x3
mov DWORD PTR [ebp-0x8],0x4
mov eax, DWORD PTR [ebp+0x8]
mov DWORD PTR [ebp-0x4],eax
mov eax,DWORD PTR [ebp+0xc]
mov DWORD PTR [ebp-0x8],eax
push DWORD PTR [ebp-0x8]
push DWORD PTR [ebp-0x4]
call _BFunc
add esp,0x8
mov eax,0x8
leave
ret
```



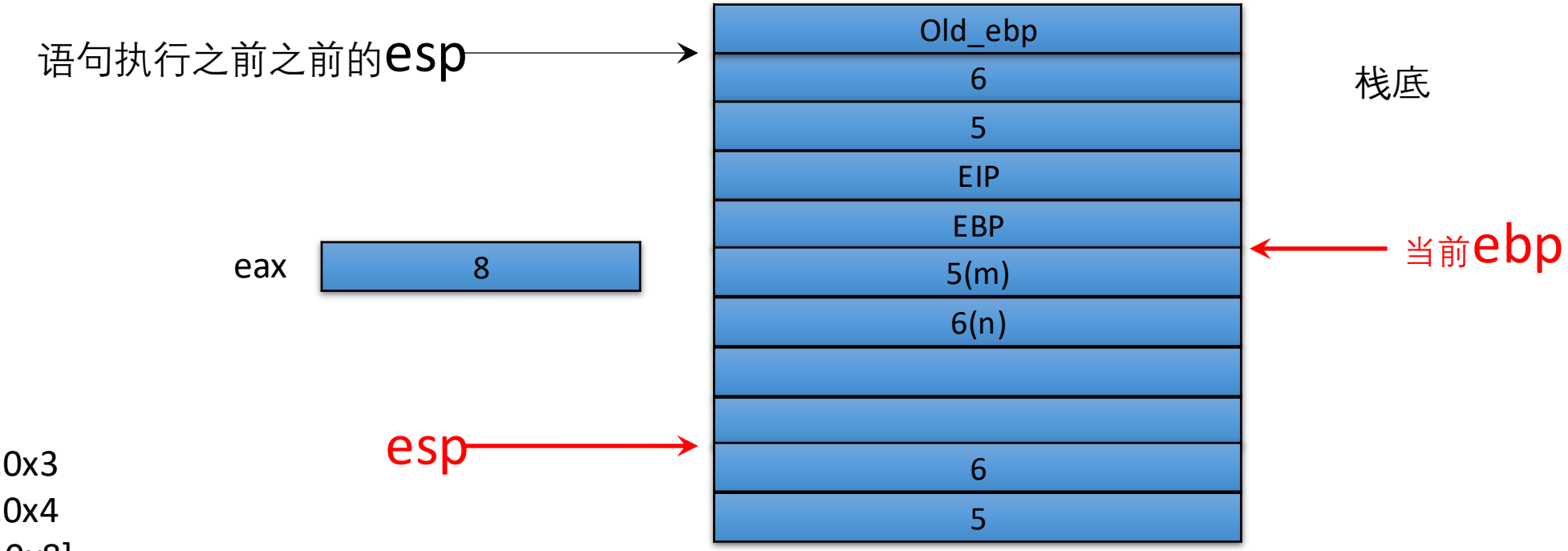
```
AFunc(5,6);
push 6
push 5
call AFunc
add esp,8

AFunc
push ebp
mov ebp,esp
sub esp,0x10
mov DWORD PTR [ebp-0x4],0x3
mov DWORD PTR [ebp-0x8],0x4
mov eax, DWORD PTR [ebp+0x8]
mov DWORD PTR [ebp-0x4],eax
mov eax,DWORD PTR [ebp+0xc]
mov DWORD PTR [ebp-0x8],eax
push DWORD PTR [ebp-0x8]
push DWORD PTR [ebp-0x4]
call _BFunc
add esp,0x8
mov eax,0x8
leave
ret
```



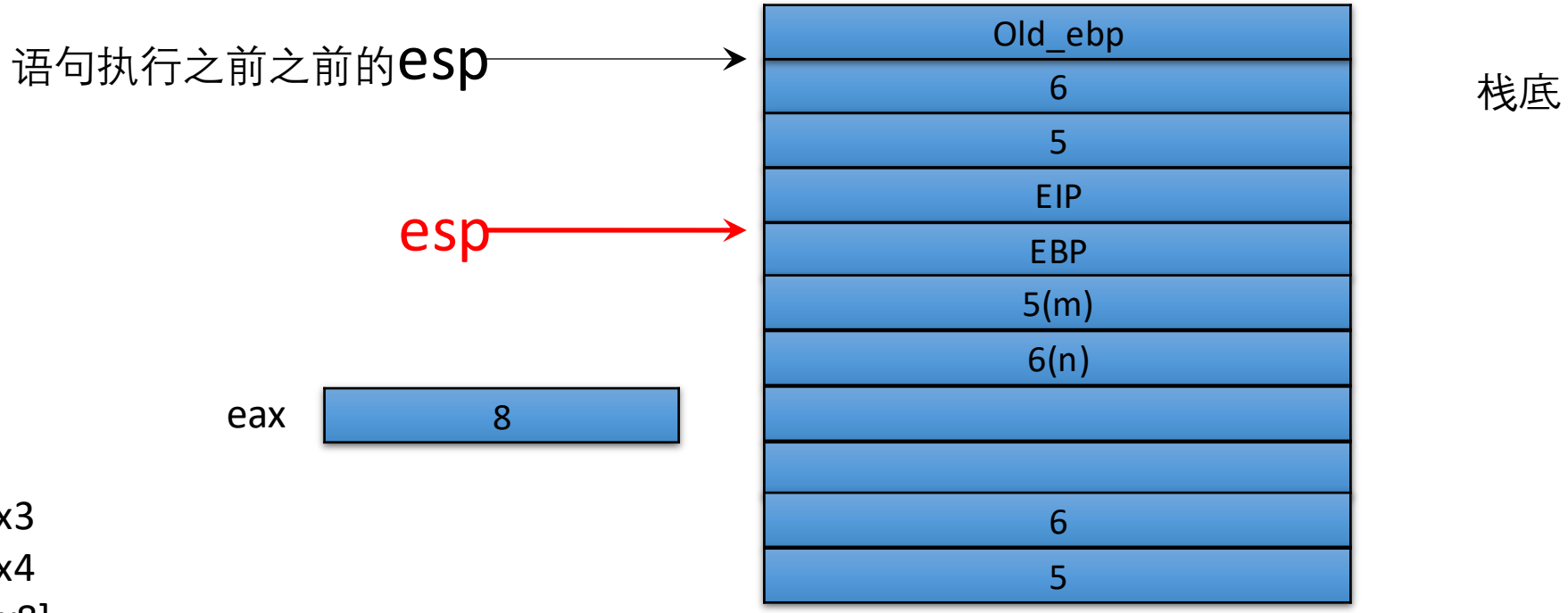
AFunc(5,6);
push 6
push 5
call AFunc
add esp,8

AFunc
push ebp
mov ebp,esp
sub esp,0x10
mov DWORD PTR [ebp-0x4],0x3
mov DWORD PTR [ebp-0x8],0x4
mov eax, DWORD PTR [ebp+0x8]
mov DWORD PTR [ebp-0x4],eax
mov eax,DWORD PTR [ebp+0xc]
mov DWORD PTR [ebp-0x8],eax
push DWORD PTR [ebp-0x8]
push DWORD PTR [ebp-0x4]
call _BFunc
add esp,0x8
mov eax,0x8
leave
ret



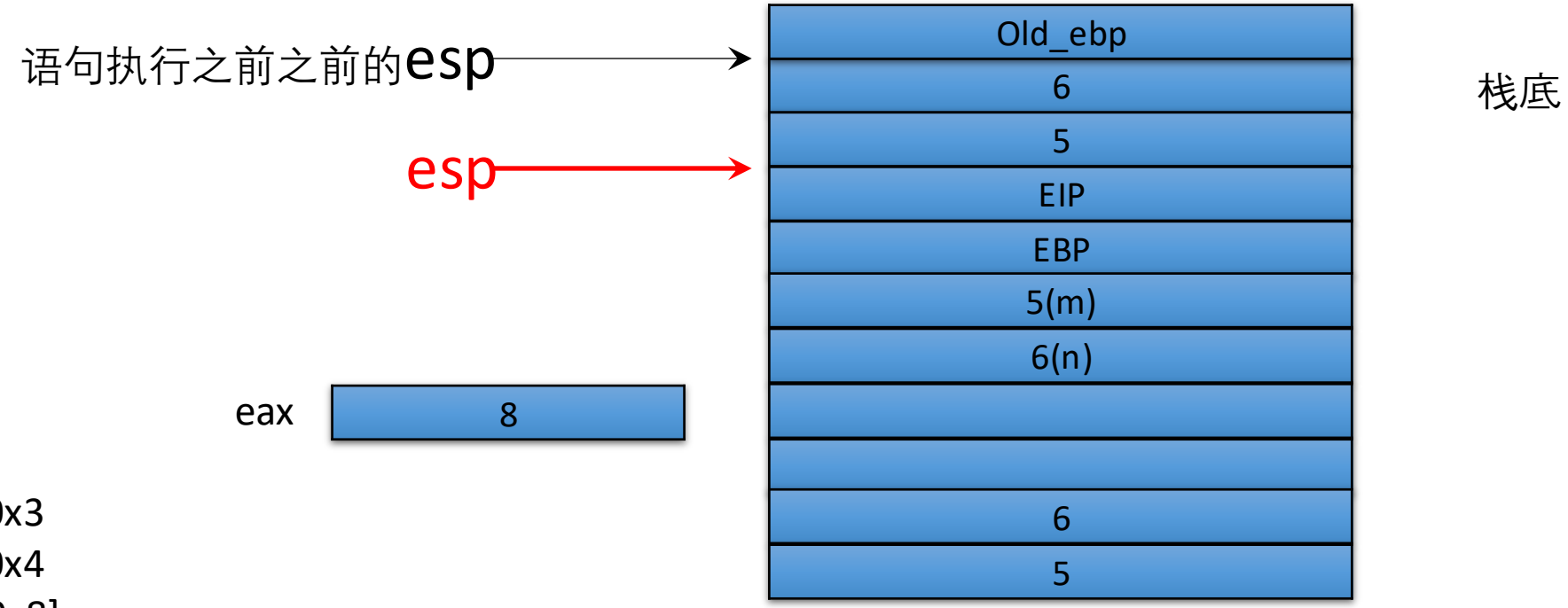
```
AFunc(5,6);
push 6
push 5
call AFunc
add esp,8

AFunc
push ebp
mov ebp,esp
sub esp,0x10
mov DWORD PTR [ebp-0x4],0x3
mov DWORD PTR [ebp-0x8],0x4
mov eax, DWORD PTR [ebp+0x8]
mov DWORD PTR [ebp-0x4],eax
mov eax,DWORD PTR [ebp+0xc]
mov DWORD PTR [ebp-0x8],eax
push DWORD PTR [ebp-0x8]
push DWORD PTR [ebp-0x4]
call _BFunc
add esp,0x8
mov eax,0x8
leave
ret
```



```
AFunc(5,6);
push 6
push 5
call AFunc
add esp,8

AFunc
push ebp
mov ebp,esp
sub esp,0x10
mov DWORD PTR [ebp-0x4],0x3
mov DWORD PTR [ebp-0x8],0x4
mov eax, DWORD PTR [ebp+0x8]
mov DWORD PTR [ebp-0x4],eax
mov eax,DWORD PTR [ebp+0xc]
mov DWORD PTR [ebp-0x8],eax
push DWORD PTR [ebp-0x8]
push DWORD PTR [ebp-0x4]
call _BFunc
add esp,0x8
mov eax,0x8
leave
ret
```



```
AFunc(5,6);
push 6
push 5
call AFunc
add esp,8

AFunc
push ebp
mov ebp,esp
sub esp,0x10
mov DWORD PTR [ebp-0x4],0x3
mov DWORD PTR [ebp-0x8],0x4
mov eax, DWORD PTR [ebp+0x8]
mov DWORD PTR [ebp-0x4],eax
mov eax,DWORD PTR [ebp+0xc]
mov DWORD PTR [ebp-0x8],eax
push DWORD PTR [ebp-0x8]
push DWORD PTR [ebp-0x4]
call _BFunc
add esp,0x8
mov eax,0x8
leave
ret
```