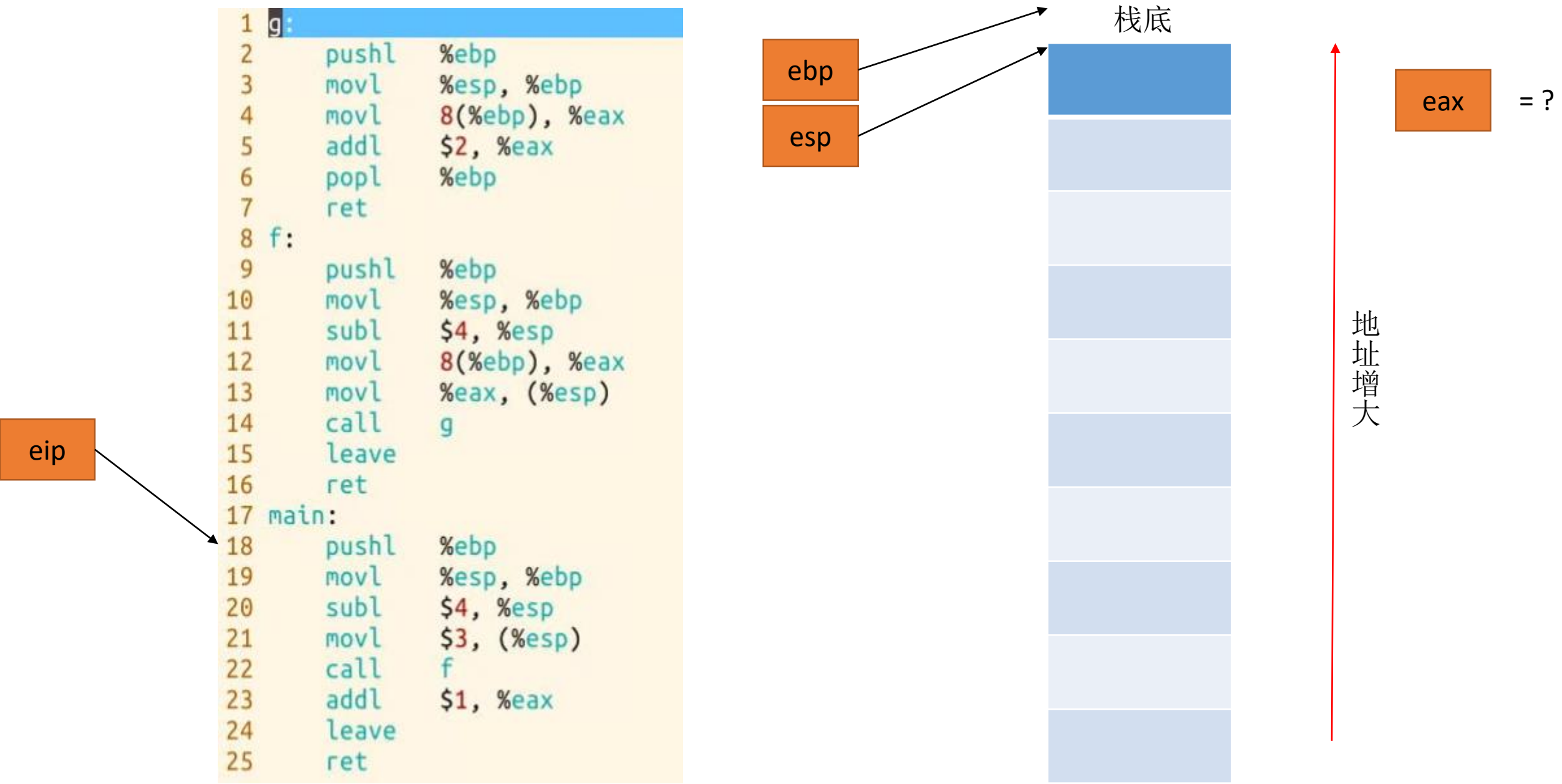
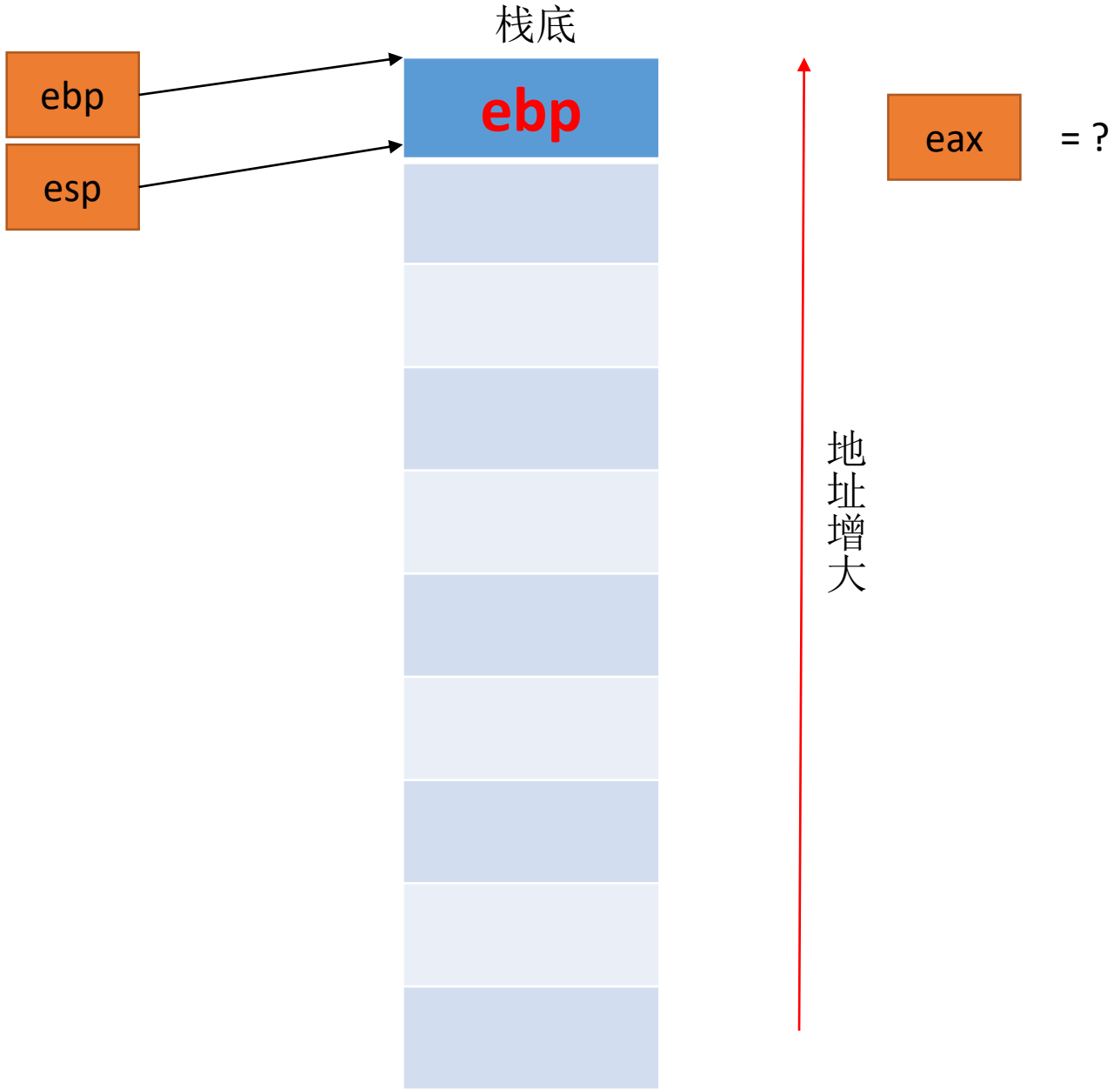
```
1  g:
2      pushl   %ebp
3      movl    %esp, %ebp
4      movl    8(%ebp), %eax
5      addl    $2, %eax
6      popl    %ebp
7      ret
8  f:
9      pushl   %ebp
10     movl    %esp, %ebp
11     subl    $4, %esp
12     movl    8(%ebp), %eax
13     movl    %eax, (%esp)
14     call    g
15     leave
16     ret
17 main:
18     pushl   %ebp
19     movl    %esp, %ebp
20     subl    $4, %esp
21     movl    $3, (%esp)
22     call    f
23     addl    $1, %eax
24     leave
25     ret
```



函数从main函数开始执行

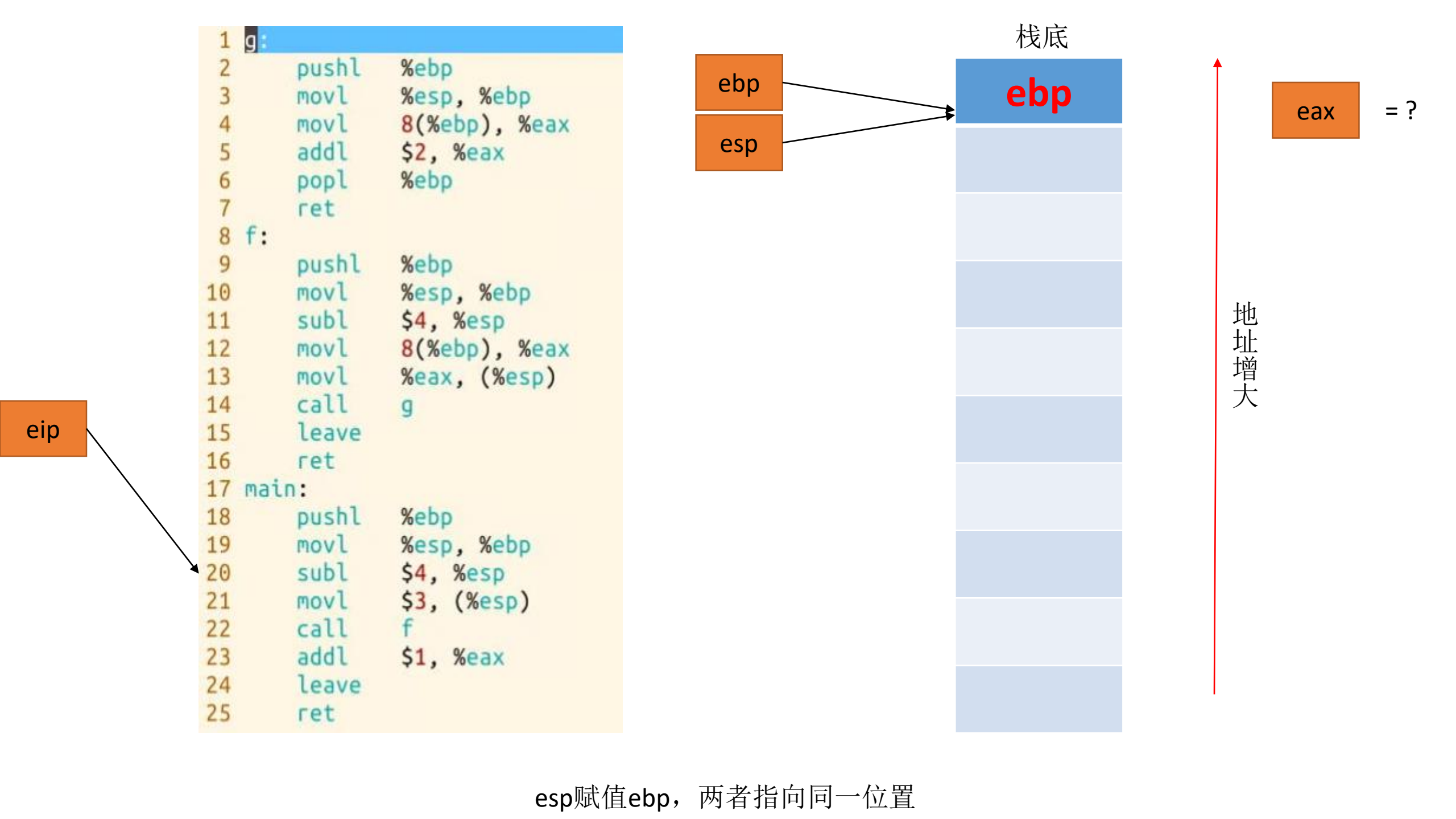
```
1  g:
2      pushl   %ebp
3      movl    %esp, %ebp
4      movl    8(%ebp), %eax
5      addl    $2, %eax
6      popl    %ebp
7      ret
8  f:
9      pushl   %ebp
10     movl    %esp, %ebp
11     subl    $4, %esp
12     movl    8(%ebp), %eax
13     movl    %eax, (%esp)
14     call    g
15     leave
16     ret
17 main:
18     pushl   %ebp
19     movl    %esp, %ebp
20     subl    $4, %esp
21     movl    $3, (%esp)
22     call    f
23     addl    $1, %eax
24     leave
25     ret
```

eip



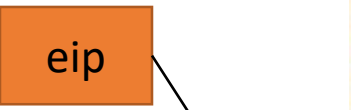
esp-4,ebp入栈

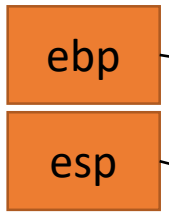
```
1  g:
2      pushl   %ebp
3      movl    %esp, %ebp
4      movl    8(%ebp), %eax
5      addl    $2, %eax
6      popl    %ebp
7      ret
8  f:
9      pushl   %ebp
10     movl    %esp, %ebp
11     subl    $4, %esp
12     movl    8(%ebp), %eax
13     movl    %eax, (%esp)
14     call    g
15     leave
16     ret
17 main:
18     pushl   %ebp
19     movl    %esp, %ebp
20     subl    $4, %esp
21     movl    $3, (%esp)
22     call    f
23     addl    $1, %eax
24     leave
25     ret
```

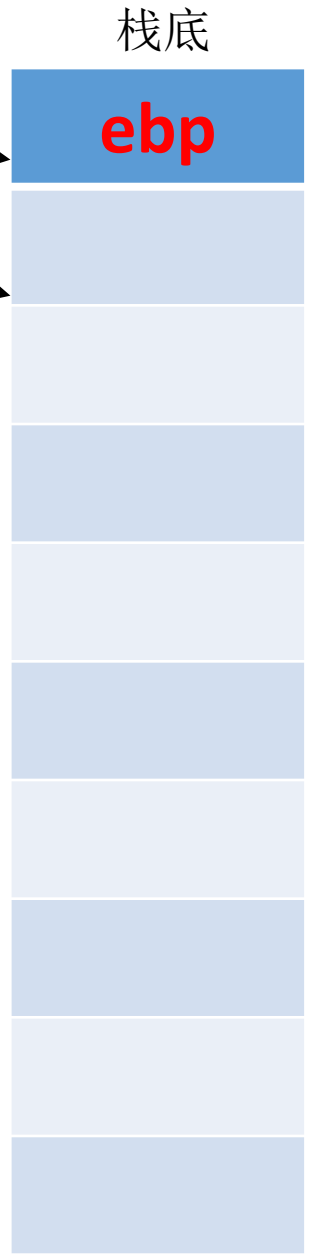


esp赋值ebp，两者指向同一位置

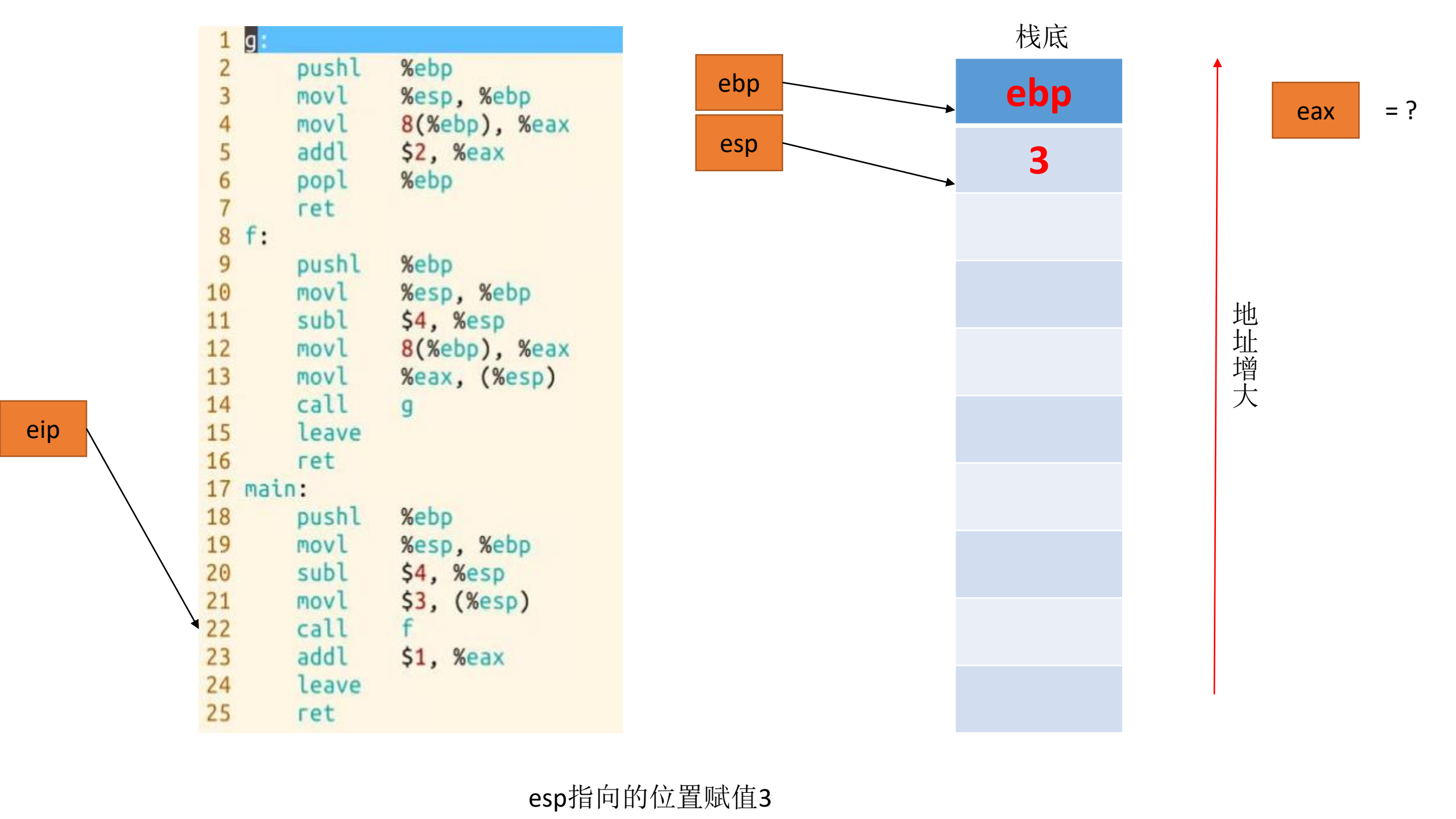
```
1  g:
2      pushl   %ebp
3      movl    %esp, %ebp
4      movl    8(%ebp), %eax
5      addl    $2, %eax
6      popl    %ebp
7      ret
8  f:
9      pushl   %ebp
10     movl    %esp, %ebp
11     subl    $4, %esp
12     movl    8(%ebp), %eax
13     movl    %eax, (%esp)
14     call    g
15     leave
16     ret
17 main:
18     pushl   %ebp
19     movl    %esp, %ebp
20     subl    $4, %esp
21     movl    $3, (%esp)
22     call    f
23     addl    $1, %eax
24     leave
25     ret
```







```
1  g:
2      pushl   %ebp
3      movl    %esp, %ebp
4      movl    8(%ebp), %eax
5      addl    $2, %eax
6      popl    %ebp
7      ret
8  f:
9      pushl   %ebp
10     movl    %esp, %ebp
11     subl    $4, %esp
12     movl    8(%ebp), %eax
13     movl    %eax, (%esp)
14     call    g
15     leave
16     ret
17 main:
18     pushl   %ebp
19     movl    %esp, %ebp
20     subl    $4, %esp
21     movl    $3, (%esp)
22     call    f
23     addl    $1, %eax
24     leave
25     ret
```



esp指向的位置赋值3

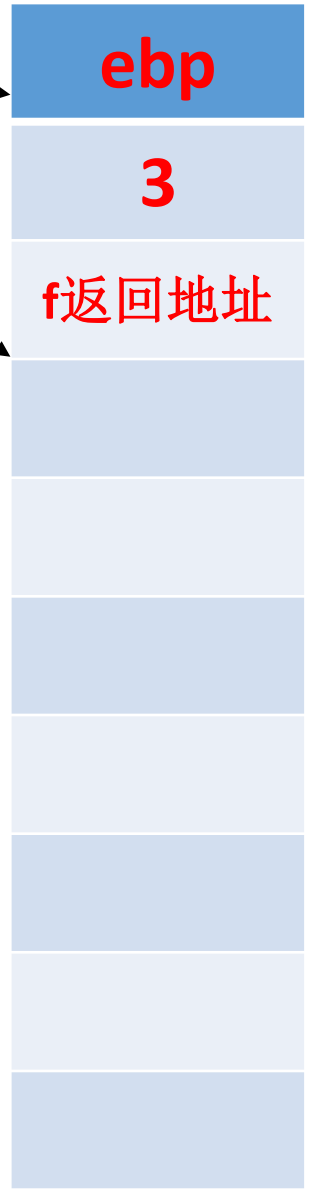
```
1  g:
2      pushl   %ebp
3      movl    %esp, %ebp
4      movl    8(%ebp), %eax
5      addl    $2, %eax
6      popl    %ebp
7      ret
8  f:
9      pushl   %ebp
10     movl    %esp, %ebp
11     subl    $4, %esp
12     movl    8(%ebp), %eax
13     movl    %eax, (%esp)
14     call    g
15     leave
16     ret
17 main:
18     pushl   %ebp
19     movl    %esp, %ebp
20     subl    $4, %esp
21     movl    $3, (%esp)
22     call    f
23     addl    $1, %eax
24     leave
25     ret
```

eip

ebp

esp

栈底



地址增大

eax = ?

esp-4，f函数的返回地址入栈

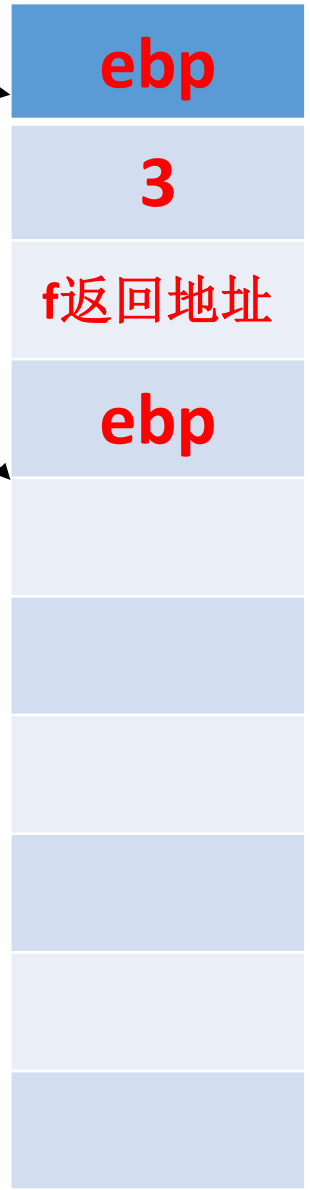
```
1  g:
2      pushl   %ebp
3      movl    %esp, %ebp
4      movl    8(%ebp), %eax
5      addl    $2, %eax
6      popl    %ebp
7      ret
8  f:
9      pushl   %ebp
10     movl    %esp, %ebp
11     subl    $4, %esp
12     movl    8(%ebp), %eax
13     movl    %eax, (%esp)
14     call    g
15     leave
16     ret
17 main:
18     pushl   %ebp
19     movl    %esp, %ebp
20     subl    $4, %esp
21     movl    $3, (%esp)
22     call    f
23     addl    $1, %eax
24     leave
25     ret
```

eip

ebp

esp

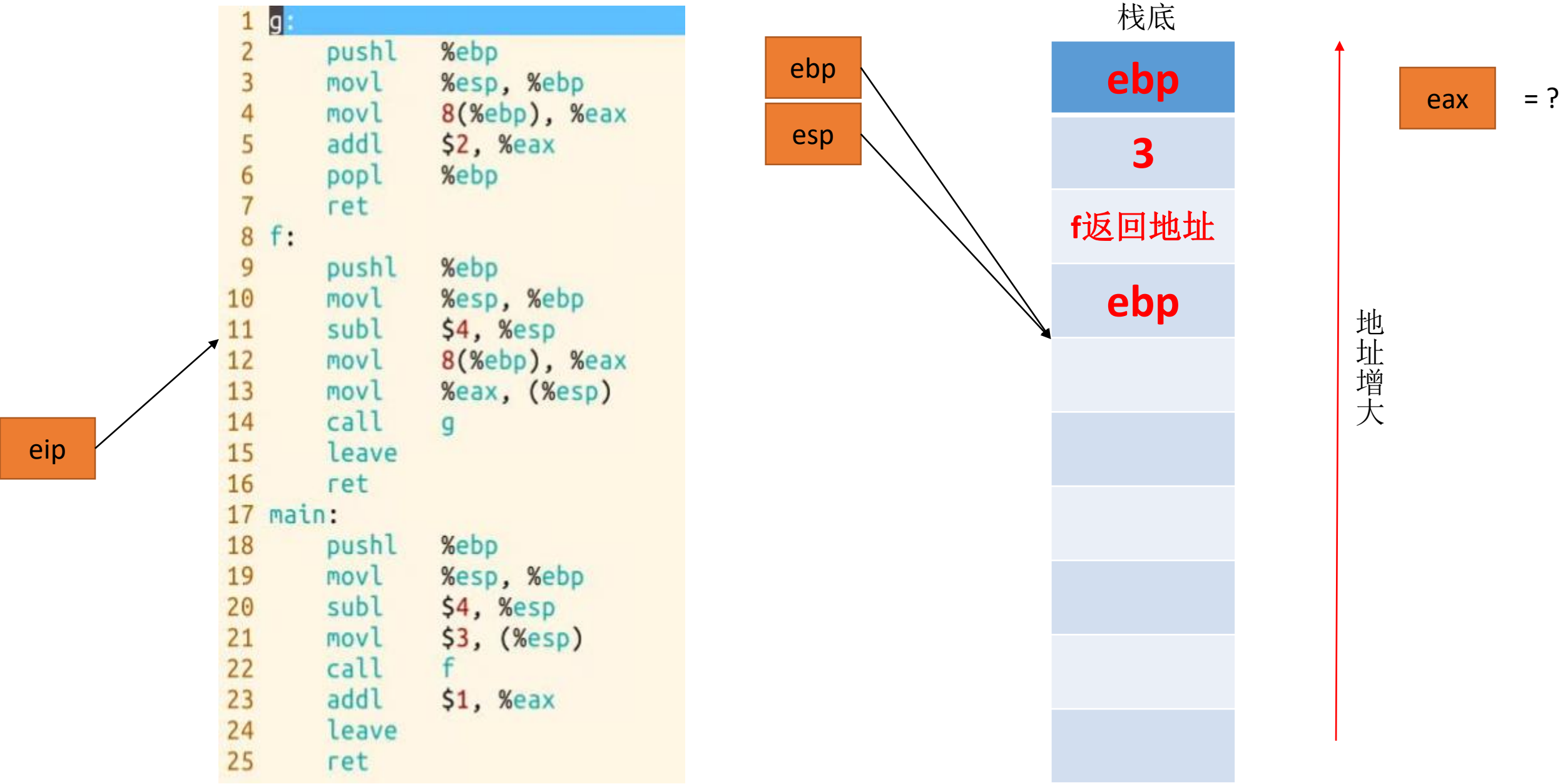
栈底



地址增大

eax = ?

esp-4，ebp入栈



esp赋值ebp，两者指向同一位置

```
1  g:
2      pushl   %ebp
3      movl    %esp, %ebp
4      movl    8(%ebp), %eax
5      addl    $2, %eax
6      popl    %ebp
7      ret
8  f:
9      pushl   %ebp
10     movl    %esp, %ebp
11     subl    $4, %esp
12     movl    8(%ebp), %eax
13     movl    %eax, (%esp)
14     call    g
15     leave
16     ret
17 main:
18     pushl   %ebp
19     movl    %esp, %ebp
20     subl    $4, %esp
21     movl    $3, (%esp)
22     call    f
23     addl    $1, %eax
24     leave
25     ret
```

eip

ebp

esp

栈底

地址增大

eax = ?

```
1  g:
2      pushl   %ebp
3      movl    %esp, %ebp
4      movl    8(%ebp), %eax
5      addl    $2, %eax
6      popl    %ebp
7      ret
8  f:
9      pushl   %ebp
10     movl    %esp, %ebp
11     subl    $4, %esp
12     movl    8(%ebp), %eax
13     movl    %eax, (%esp)
14     call    g
15     leave
16     ret
17 main:
18     pushl   %ebp
19     movl    %esp, %ebp
20     subl    $4, %esp
21     movl    $3, (%esp)
22     call    f
23     addl    $1, %eax
24     leave
25     ret
```

eip

ebp

esp

栈底

地址增大

eax = 3

eax赋值

```
1  g:
2      pushl   %ebp
3      movl    %esp, %ebp
4      movl    8(%ebp), %eax
5      addl    $2, %eax
6      popl    %ebp
7      ret
8  f:
9      pushl   %ebp
10     movl    %esp, %ebp
11     subl    $4, %esp
12     movl    8(%ebp), %eax
13     movl    %eax, (%esp)
14     call    g
15     leave
16     ret
17 main:
18     pushl   %ebp
19     movl    %esp, %ebp
20     subl    $4, %esp
21     movl    $3, (%esp)
22     call    f
23     addl    $1, %eax
24     leave
25     ret
```

eip

ebp

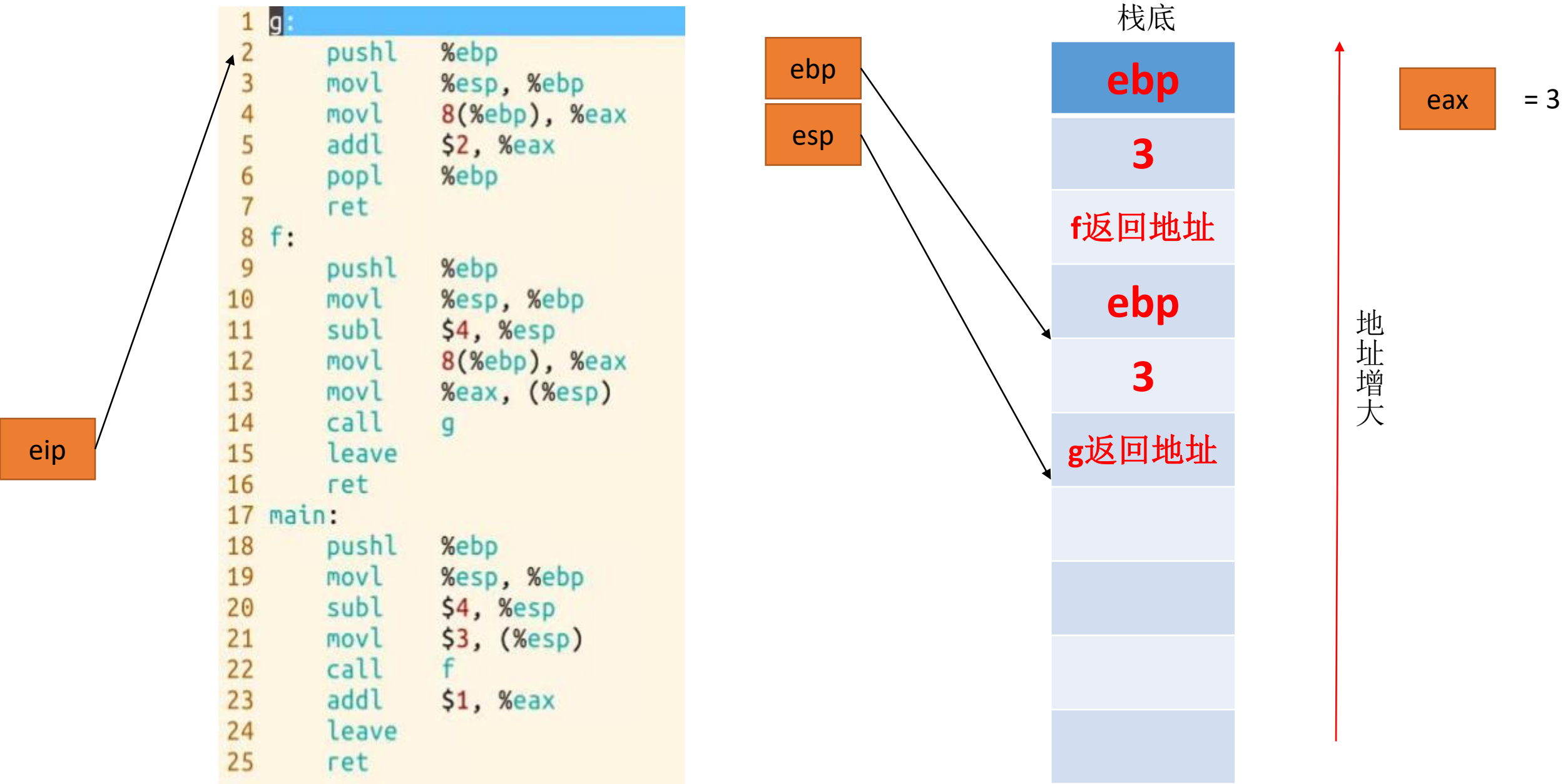
esp

栈底

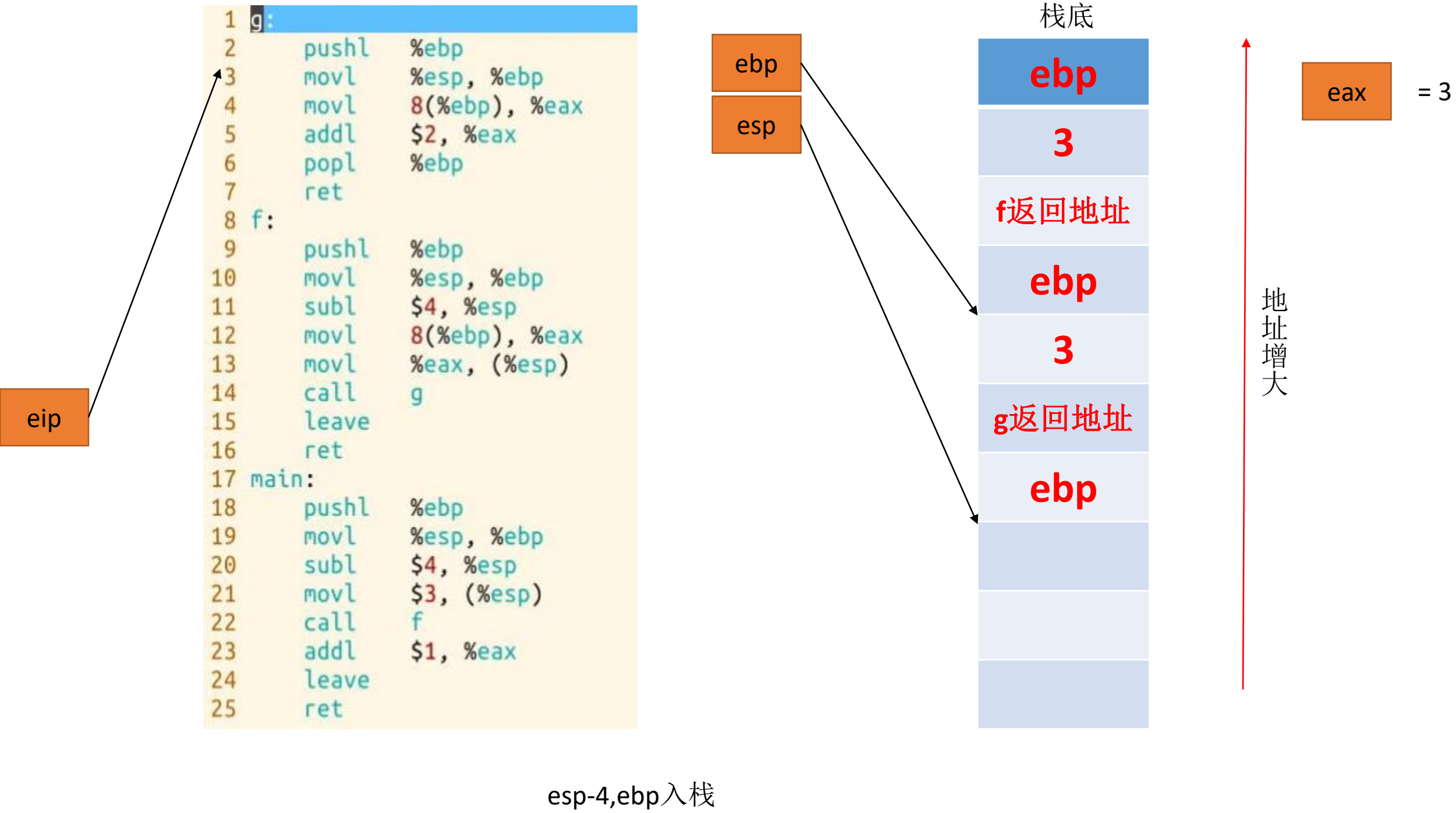
地址增大

eax = 3

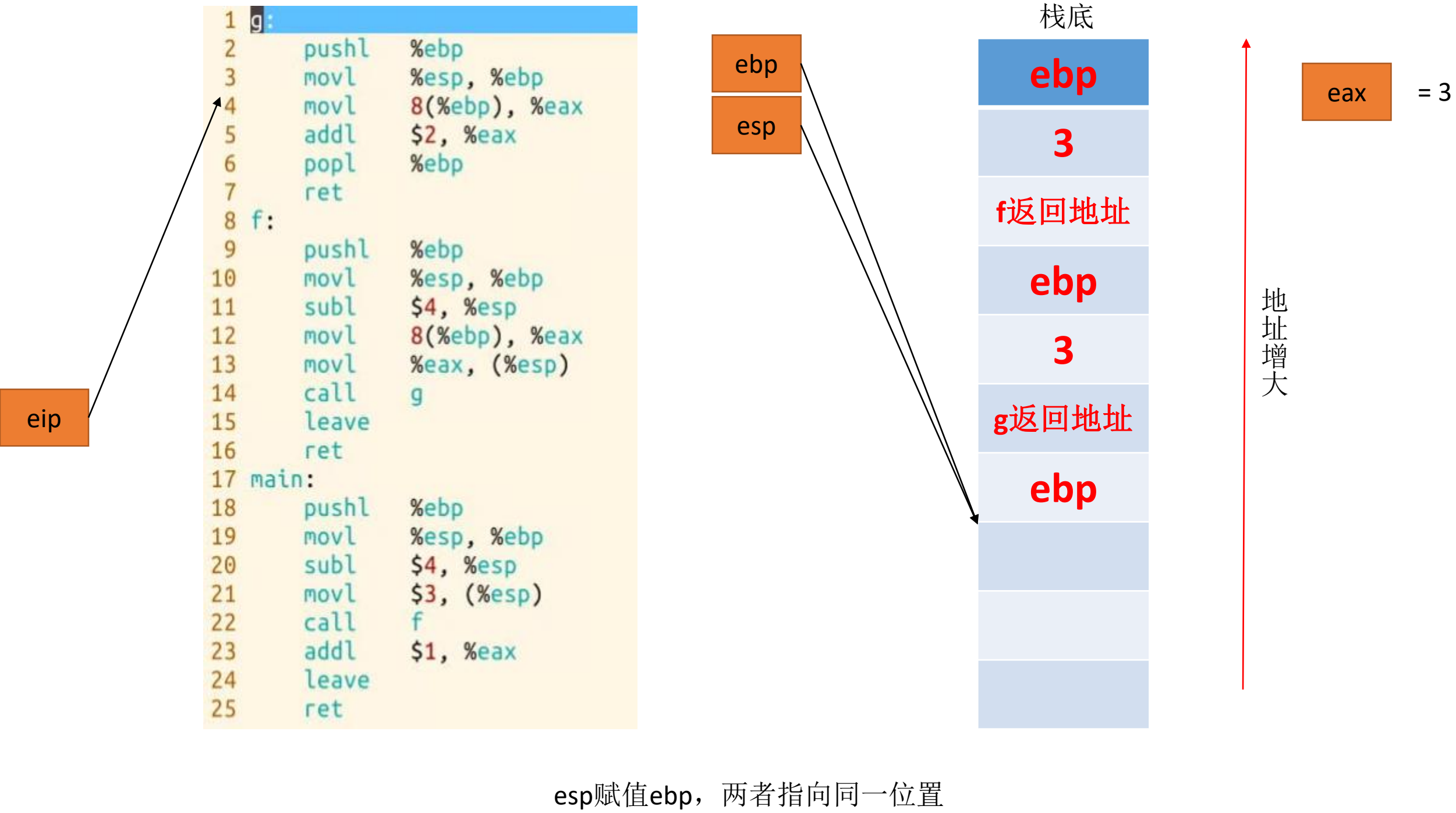
esp指向的位置赋值



esp-4,g函数返回地址入栈


1 g:
2 pushl %ebp
3 movl %esp, %ebp
4 movl 8(%ebp), %eax
5 addl $2, %eax
6 popl %ebp
7 ret
8 f:
9 pushl %ebp
10 movl %esp, %ebp
11 subl $4, %esp
12 movl 8(%ebp), %eax
13 movl %eax, (%esp)
14 call g
15 leave
16 ret
17 main:
18 pushl %ebp
19 movl %esp, %ebp
20 subl $4, %esp
21 movl $3, (%esp)
22 call f
23 addl $1, %eax
24 leave
25 ret
eip
ebp
esp
栈底
ebp
3
f返回地址
ebp
3
g返回地址
ebp
eax
= 3
地址增大

esp-4,ebp入栈



esp赋值ebp，两者指向同一位置

```
1  g:
2      pushl   %ebp
3      movl    %esp, %ebp
4      movl    8(%ebp), %eax
5      addl    $2, %eax
6      popl    %ebp
7      ret
8  f:
9      pushl   %ebp
10     movl    %esp, %ebp
11     subl    $4, %esp
12     movl    8(%ebp), %eax
13     movl    %eax, (%esp)
14     call    g
15     leave
16     ret
17 main:
18     pushl   %ebp
19     movl    %esp, %ebp
20     subl    $4, %esp
21     movl    $3, (%esp)
22     call    f
23     addl    $1, %eax
24     leave
25     ret
```

eip

ebp

esp

栈底

地址增大

eax = 3

eax赋值为3

```
1  g:
2      pushl   %ebp
3      movl    %esp, %ebp
4      movl    8(%ebp), %eax
5      addl    $2, %eax
6      popl    %ebp
7      ret
8  f:
9      pushl   %ebp
10     movl    %esp, %ebp
11     subl    $4, %esp
12     movl    8(%ebp), %eax
13     movl    %eax, (%esp)
14     call    g
15     leave
16     ret
17 main:
18     pushl   %ebp
19     movl    %esp, %ebp
20     subl    $4, %esp
21     movl    $3, (%esp)
22     call    f
23     addl    $1, %eax
24     leave
25     ret
```

eip

ebp

esp

栈底

地址增大

eax = 5

eax+2

```
1  g:
2      pushl   %ebp
3      movl    %esp, %ebp
4      movl    8(%ebp), %eax
5      addl    $2, %eax
6      popl    %ebp
7      ret
8  f:
9      pushl   %ebp
10     movl    %esp, %ebp
11     subl    $4, %esp
12     movl    8(%ebp), %eax
13     movl    %eax, (%esp)
14     call    g
15     leave
16     ret
17 main:
18     pushl   %ebp
19     movl    %esp, %ebp
20     subl    $4, %esp
21     movl    $3, (%esp)
22     call    f
23     addl    $1, %eax
24     leave
25     ret
```

eip

ebp

esp

栈底

ebp
3
f返回地址
ebp
3
g返回地址

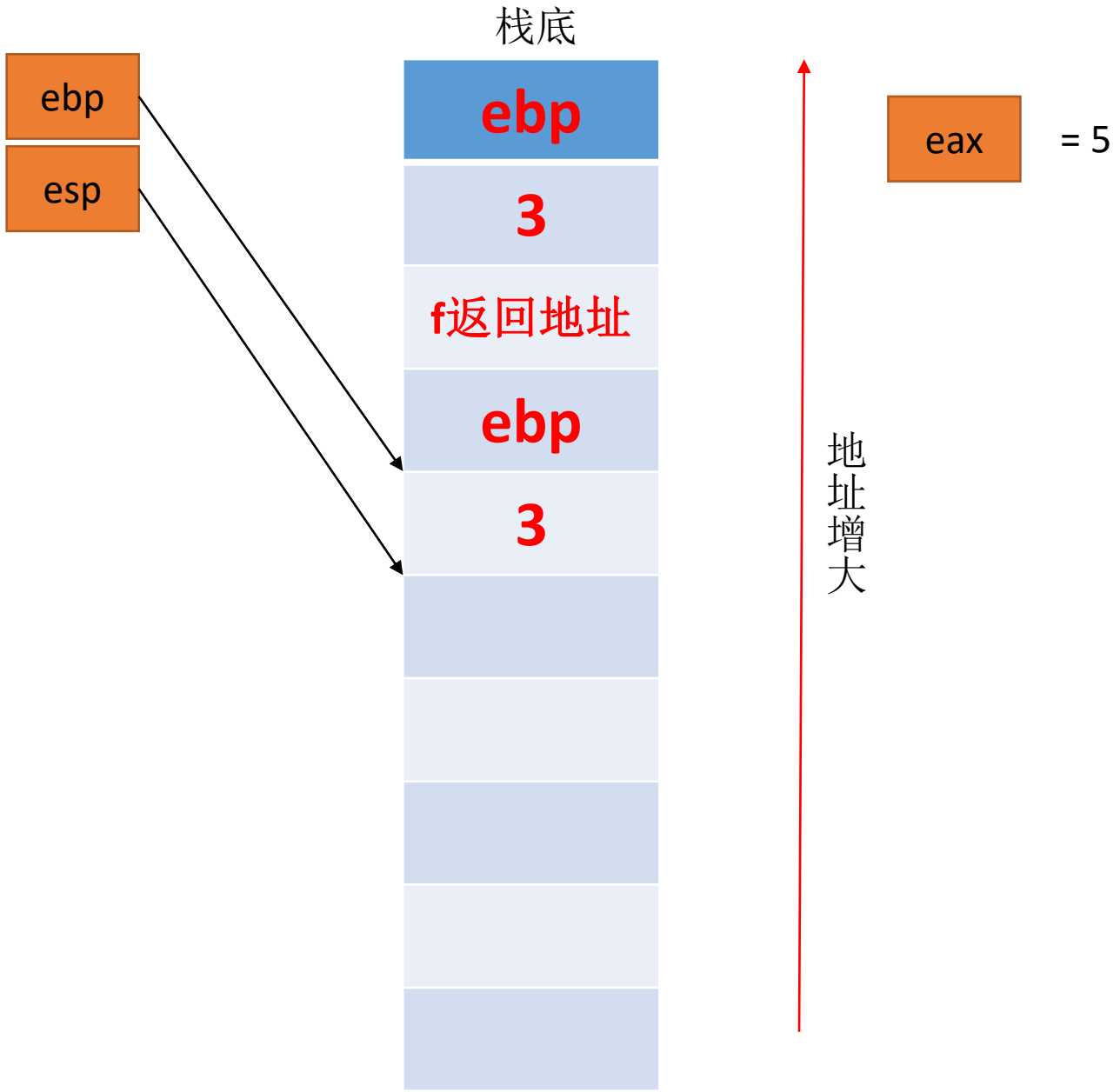
地址增大

eax = 5

ebp出栈,赋值给ebp,esp+4

```
1  g:
2      pushl   %ebp
3      movl    %esp, %ebp
4      movl    8(%ebp), %eax
5      addl    $2, %eax
6      popl    %ebp
7      ret
8  f:
9      pushl   %ebp
10     movl    %esp, %ebp
11     subl    $4, %esp
12     movl    8(%ebp), %eax
13     movl    %eax, (%esp)
14     call    g
15     leave
16     ret
17 main:
18     pushl   %ebp
19     movl    %esp, %ebp
20     subl    $4, %esp
21     movl    $3, (%esp)
22     call    f
23     addl    $1, %eax
24     leave
25     ret
```

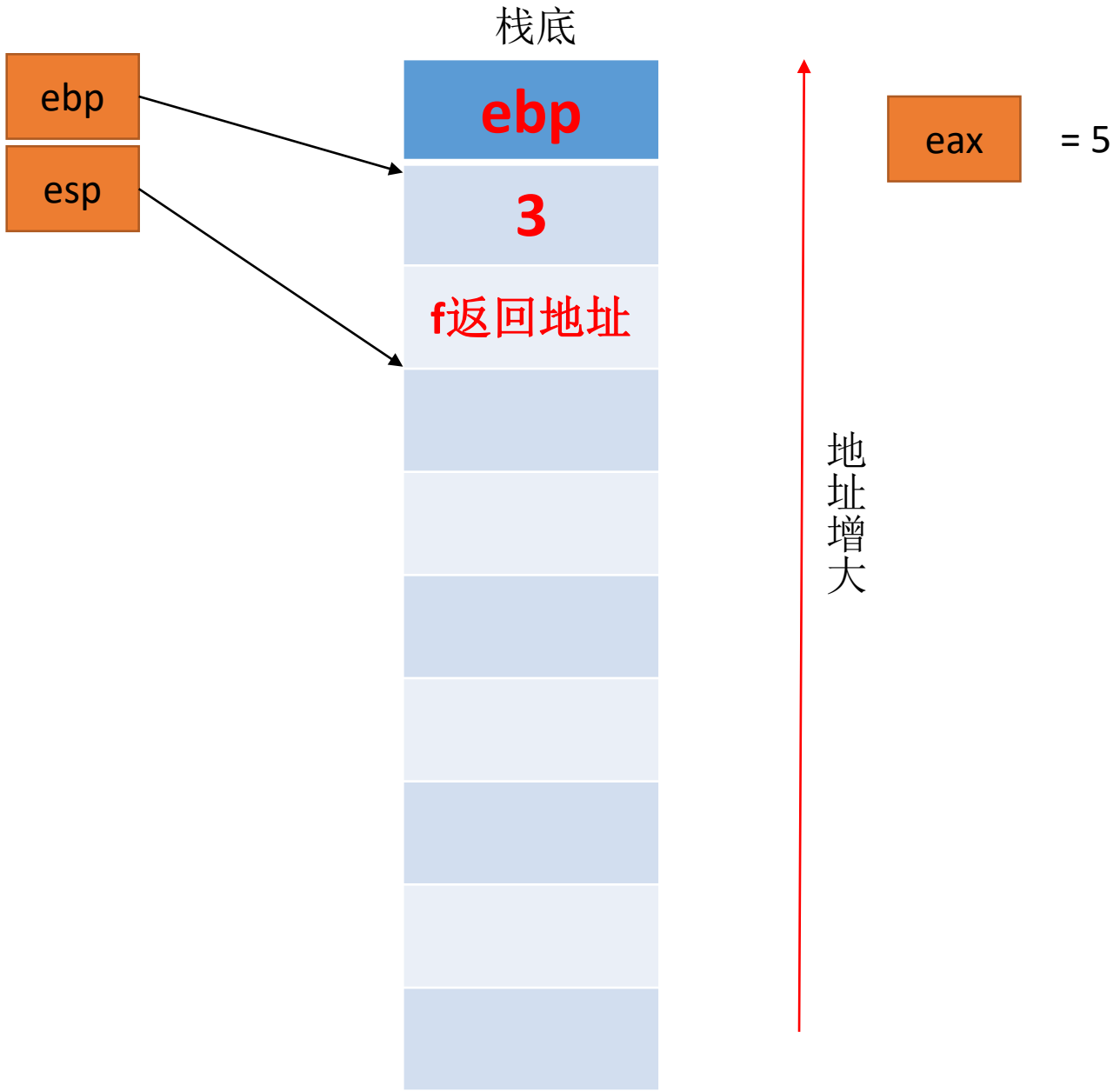
eip



g返回地址出栈，赋值给eip， esp+4

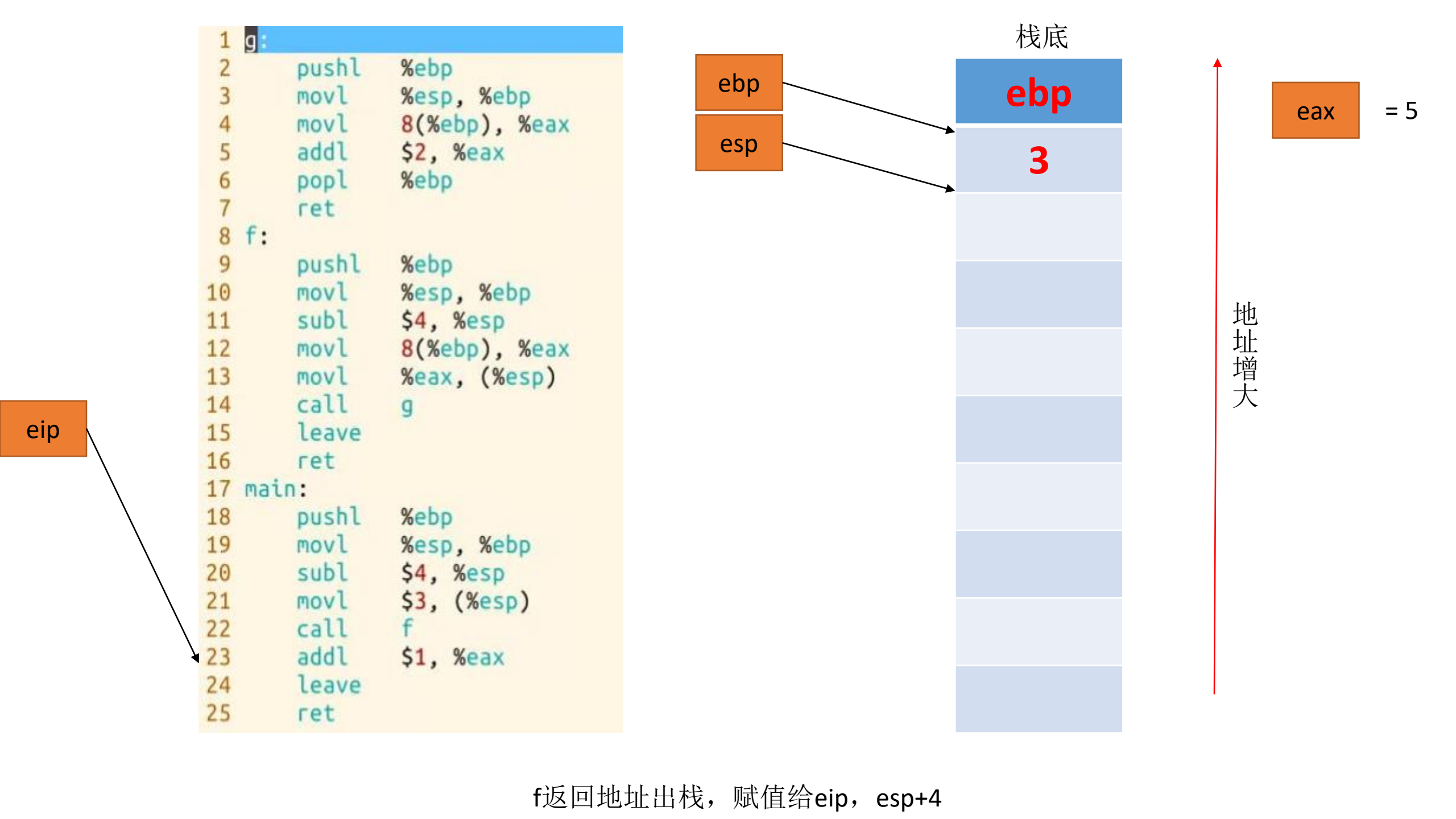
```
1  g:
2      pushl   %ebp
3      movl    %esp, %ebp
4      movl    8(%ebp), %eax
5      addl    $2, %eax
6      popl    %ebp
7      ret
8  f:
9      pushl   %ebp
10     movl    %esp, %ebp
11     subl    $4, %esp
12     movl    8(%ebp), %eax
13     movl    %eax, (%esp)
14     call    g
15     leave
16     ret
17 main:
18     pushl   %ebp
19     movl    %esp, %ebp
20     subl    $4, %esp
21     movl    $3, (%esp)
22     call    f
23     addl    $1, %eax
24     leave
25     ret
```

eip



ebp赋值给esp,ebp出栈赋值给ebp， esp+4,esp在leave指令中变化了两次

```
1  g:
2      pushl   %ebp
3      movl    %esp, %ebp
4      movl    8(%ebp), %eax
5      addl    $2, %eax
6      popl    %ebp
7      ret
8  f:
9      pushl   %ebp
10     movl    %esp, %ebp
11     subl    $4, %esp
12     movl    8(%ebp), %eax
13     movl    %eax, (%esp)
14     call    g
15     leave
16     ret
17 main:
18     pushl   %ebp
19     movl    %esp, %ebp
20     subl    $4, %esp
21     movl    $3, (%esp)
22     call    f
23     addl    $1, %eax
24     leave
25     ret
```



f返回地址出栈，赋值给eip，esp+4

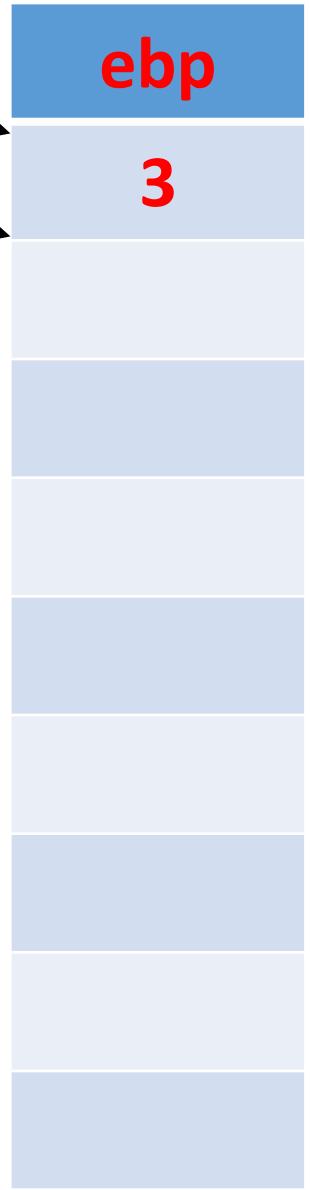
```
1  g:
2      pushl   %ebp
3      movl    %esp, %ebp
4      movl    8(%ebp), %eax
5      addl    $2, %eax
6      popl    %ebp
7      ret
8  f:
9      pushl   %ebp
10     movl    %esp, %ebp
11     subl    $4, %esp
12     movl    8(%ebp), %eax
13     movl    %eax, (%esp)
14     call    g
15     leave
16     ret
17 main:
18     pushl   %ebp
19     movl    %esp, %ebp
20     subl    $4, %esp
21     movl    $3, (%esp)
22     call    f
23     addl    $1, %eax
24     leave
25     ret
```

eip

ebp

esp

栈底

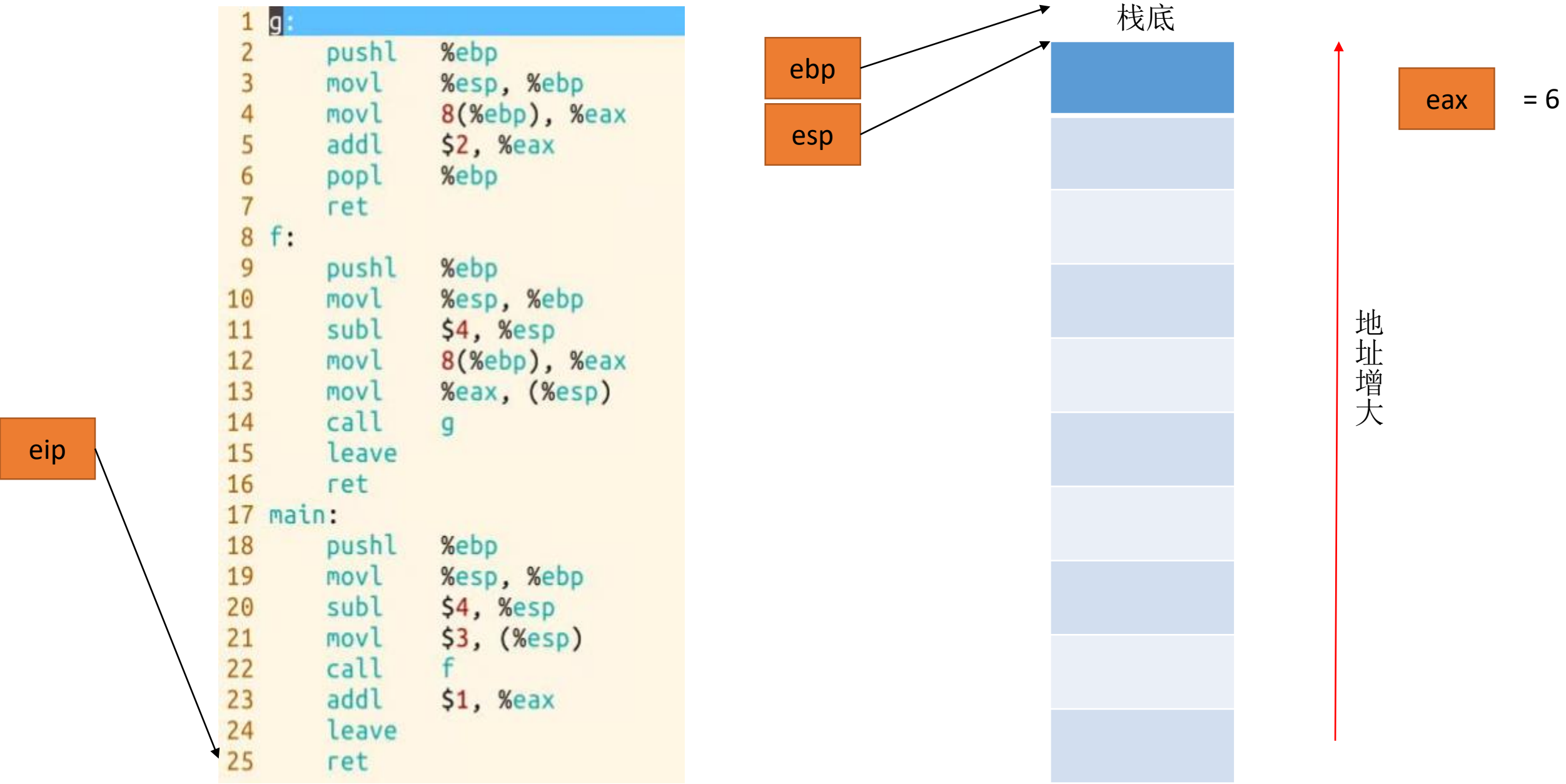


地址增大

eax = 6

eax+1

```
1  g:
2      pushl   %ebp
3      movl    %esp, %ebp
4      movl    8(%ebp), %eax
5      addl    $2, %eax
6      popl    %ebp
7      ret
8  f:
9      pushl   %ebp
10     movl    %esp, %ebp
11     subl    $4, %esp
12     movl    8(%ebp), %eax
13     movl    %eax, (%esp)
14     call    g
15     leave
16     ret
17 main:
18     pushl   %ebp
19     movl    %esp, %ebp
20     subl    $4, %esp
21     movl    $3, (%esp)
22     call    f
23     addl    $1, %eax
24     leave
25     ret
```



ebp赋值给esp,ebp出栈赋值给ebp， esp+4,esp在leave指令中变化了两次

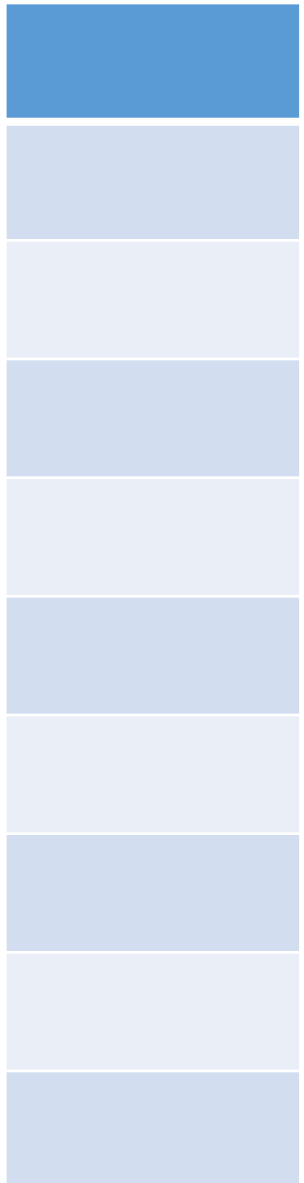
```
1  g:
2      pushl   %ebp
3      movl    %esp, %ebp
4      movl    8(%ebp), %eax
5      addl    $2, %eax
6      popl    %ebp
7      ret
8  f:
9      pushl   %ebp
10     movl    %esp, %ebp
11     subl    $4, %esp
12     movl    8(%ebp), %eax
13     movl    %eax, (%esp)
14     call    g
15     leave
16     ret
17 main:
18     pushl   %ebp
19     movl    %esp, %ebp
20     subl    $4, %esp
21     movl    $3, (%esp)
22     call    f
23     addl    $1, %eax
24     leave
25     ret
```

ebp

esp

eip

栈底



地址增大

eax = 6

运行结束