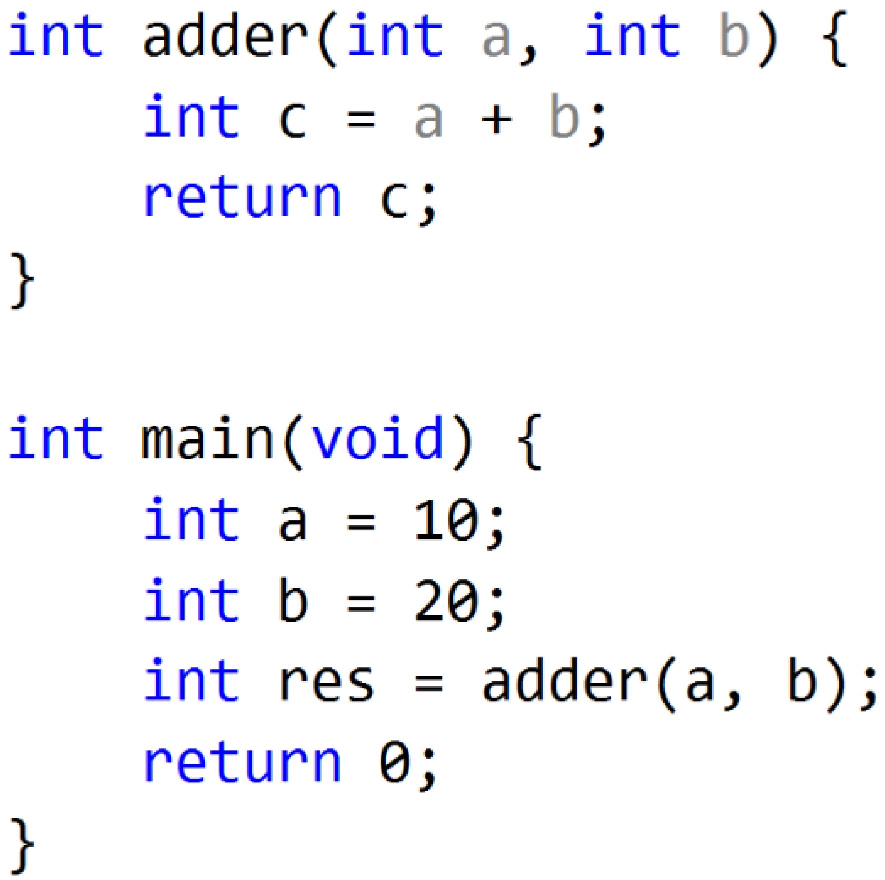
Stack frame

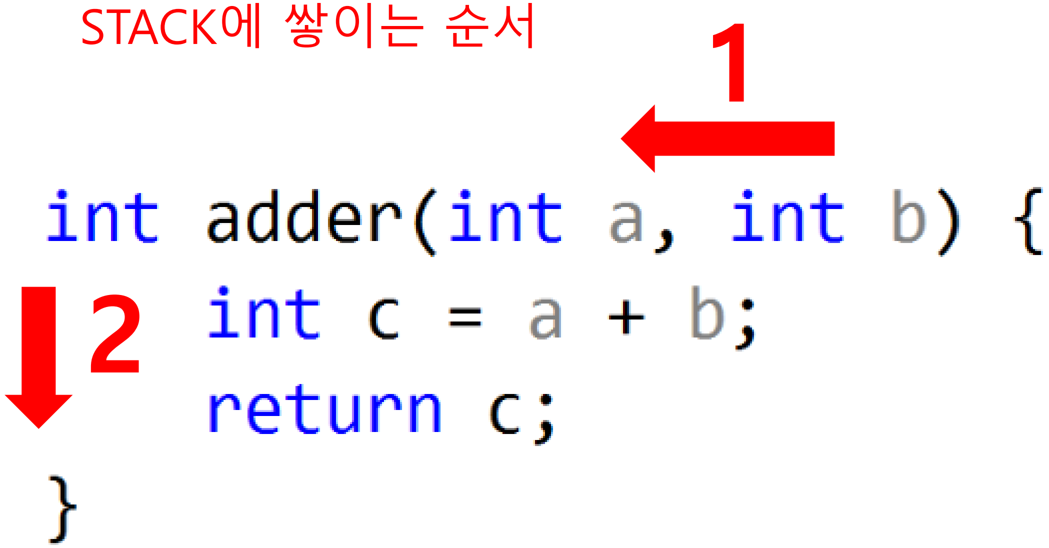


```c
int adder(int a, int b) {
    int c = a + b;
    return c;
}

int main(void) {
    int a = 10;
    int b = 20;
    int res = adder(a, b);
    return 0;
}
```

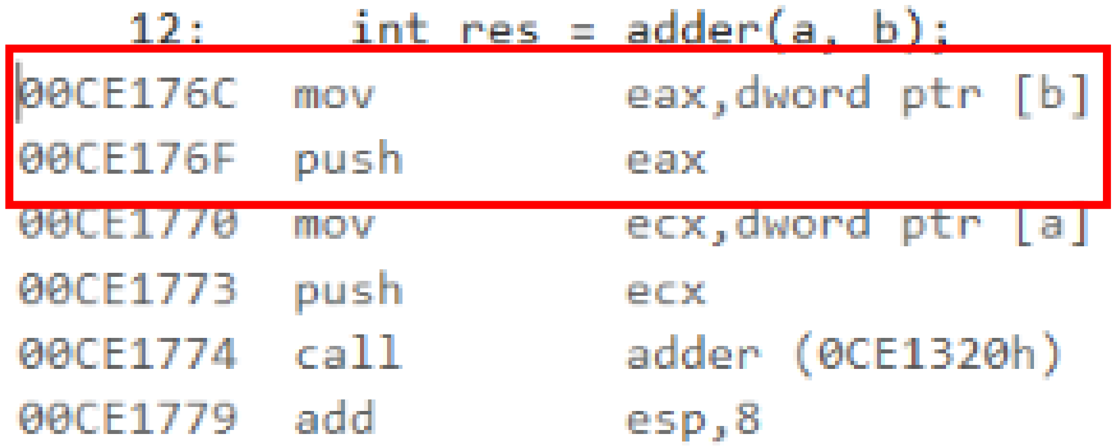
Stack frame

STACK에 쌓이는 순서



```c
int adder(int a, int b) {
    int c = a + b;
    return c;
}
```

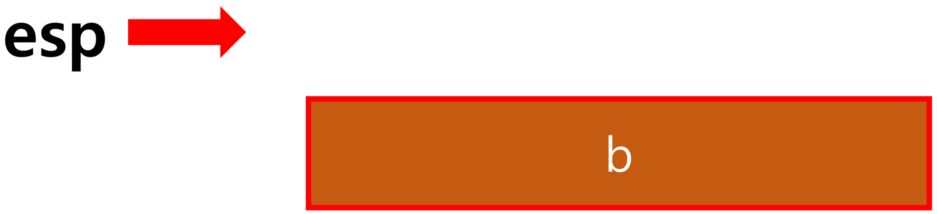
Stack frame



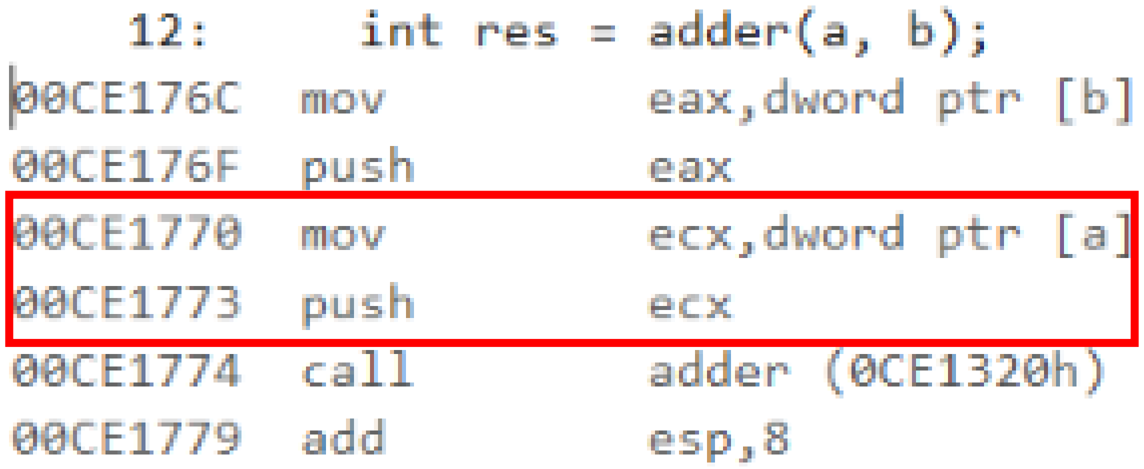
```
    12:         int res = adder(a, b);
00CE176C   mov          eax,dword ptr [b]
00CE176F   push         eax
00CE1770   mov          ecx,dword ptr [a]
00CE1773   push         ecx
00CE1774   call         adder (0CE1320h)
00CE1779   add          esp,8
```

stack pointer가 가리키는 곳에 push

esp : extended stack pointer

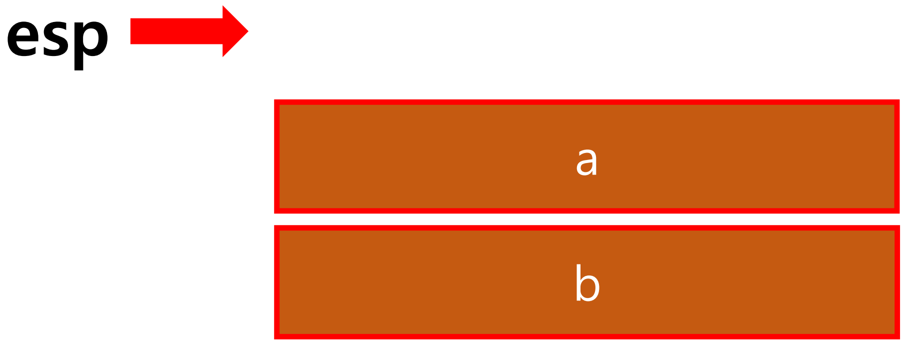


Stack frame

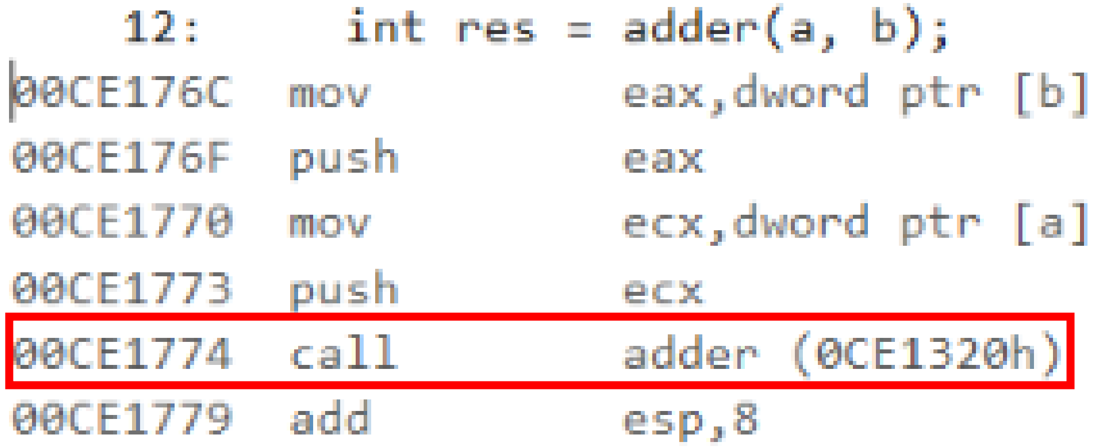


```
    12:      int res = adder(a, b);
00CE176C  mov          eax,dword ptr [b]
00CE176F  push         eax
00CE1770  mov          ecx,dword ptr [a]
00CE1773  push         ecx
00CE1774  call         adder (0CE1320h)
00CE1779  add          esp,8
```

stack pointer가 가리키는 곳에 push

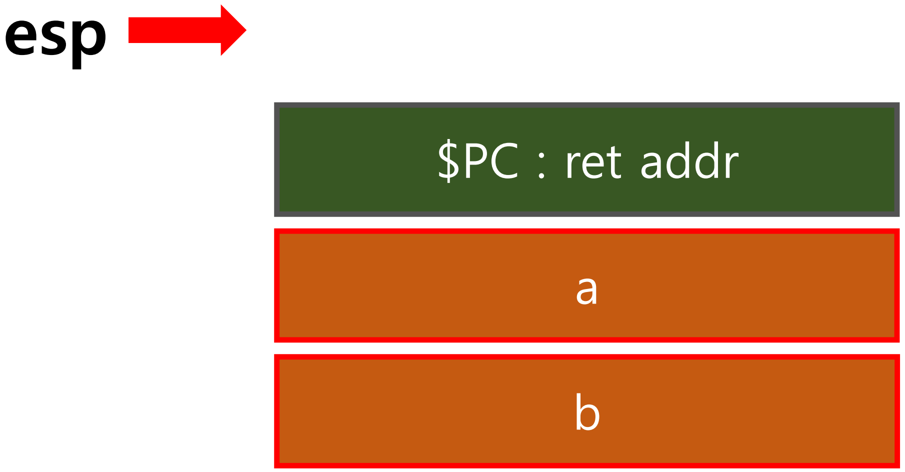


Stack frame



```
    12:      int res = adder(a, b);
00CE176C  mov          eax,dword ptr [b]
00CE176F  push         eax
00CE1770  mov          ecx,dword ptr [a]
00CE1773  push         ecx
00CE1774  call         adder (0CE1320h)
00CE1779  add          esp,8
```

함수 호출



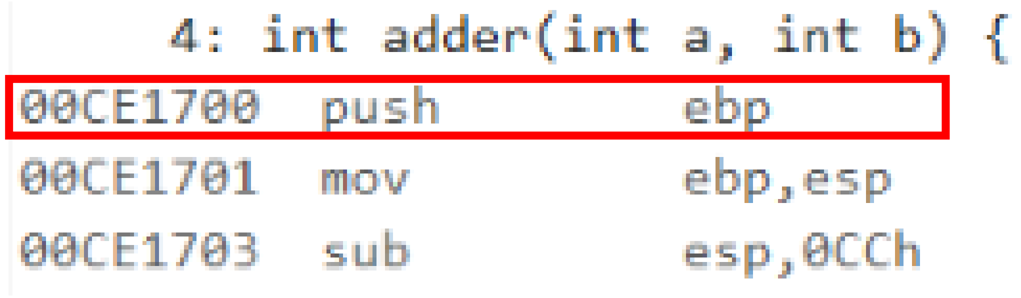
esp

$PC : ret addr

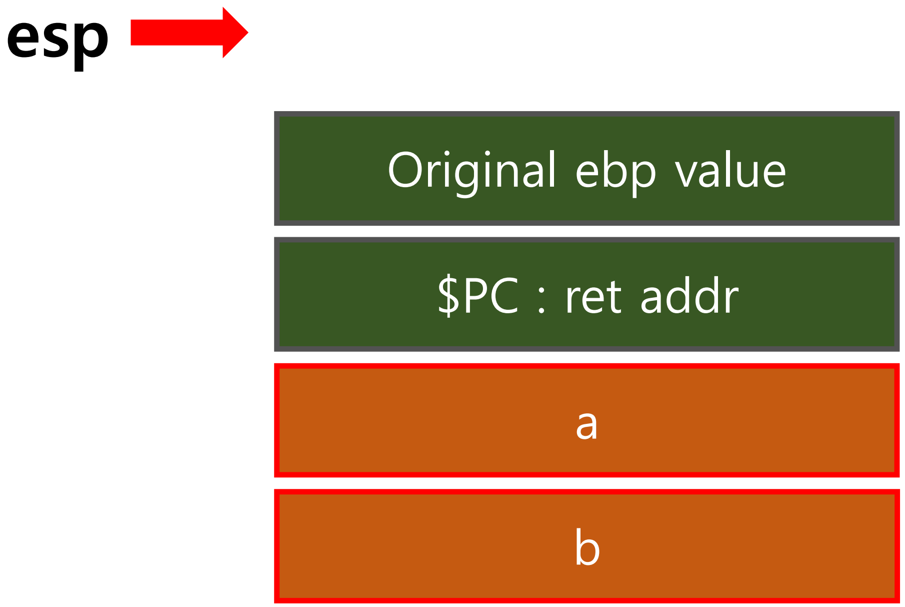
a

b

Stack frame



```
     4: int adder(int a, int b) {
00CE1700  push         ebp
00CE1701  mov          ebp,esp
00CE1703  sub          esp,0CCh
```



esp

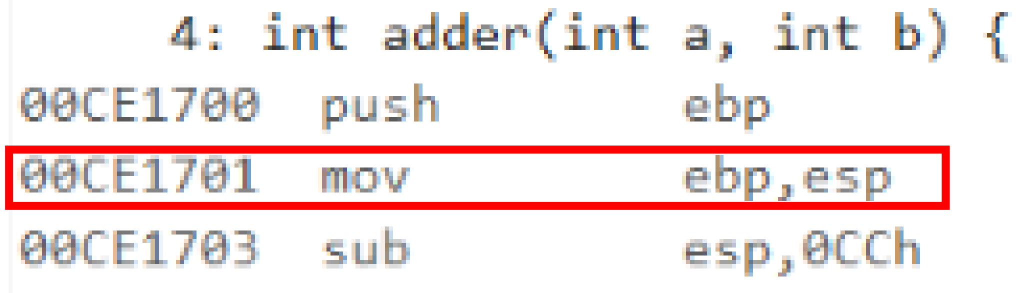
Original ebp value

$PC : ret addr

a

b

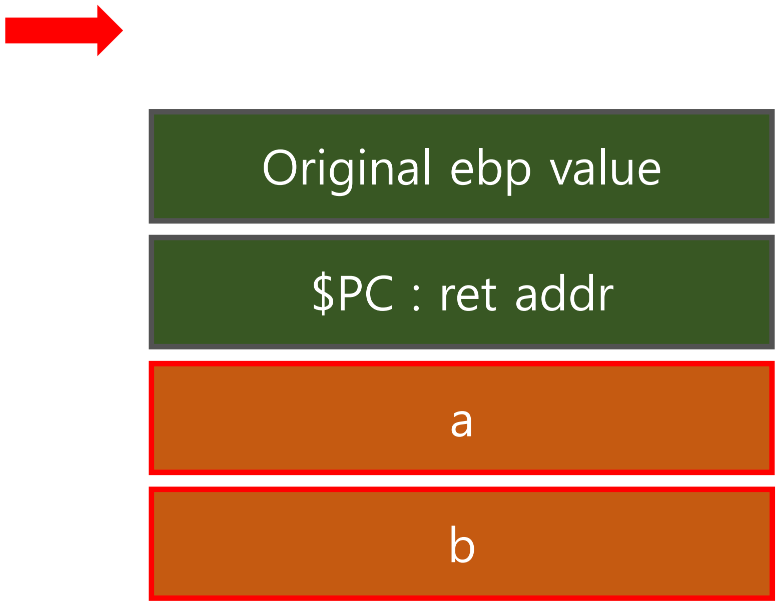
Stack frame



```
      4: int adder(int a, int b) {
00CE1700  push        ebp
00CE1701  mov         ebp,esp
00CE1703  sub         esp,0CCh
```

ebp : extended base pointer

ebp = esp



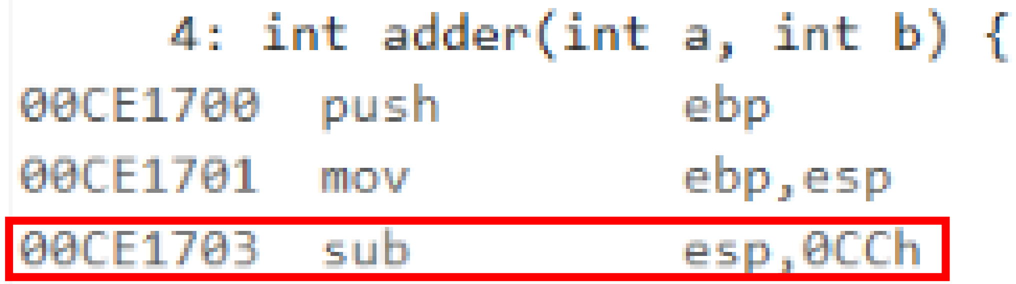
Original ebp value

$PC : ret addr

a

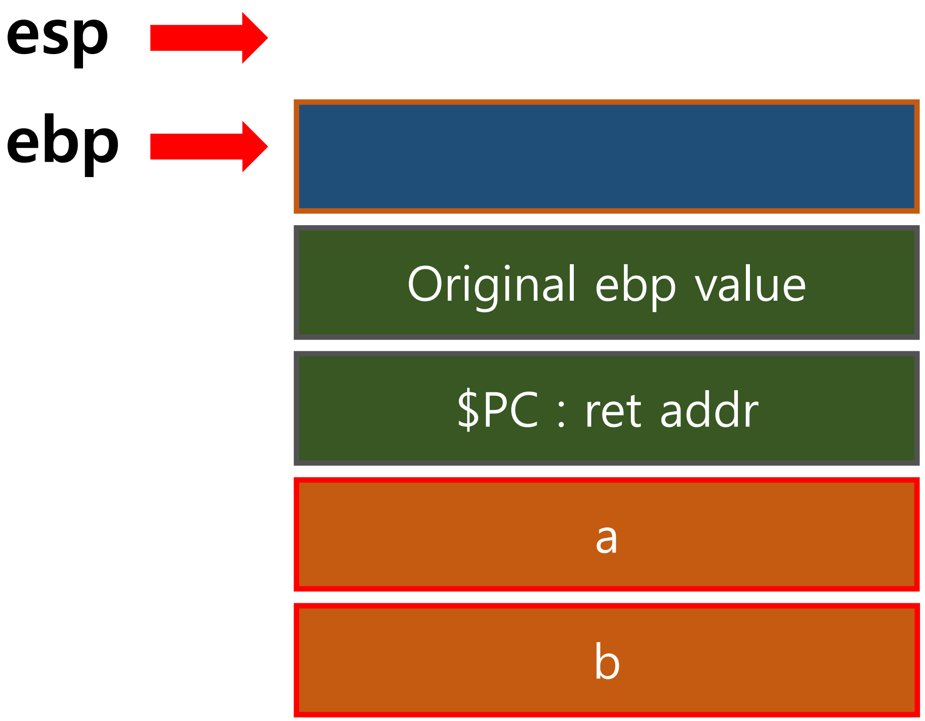
b

Stack frame

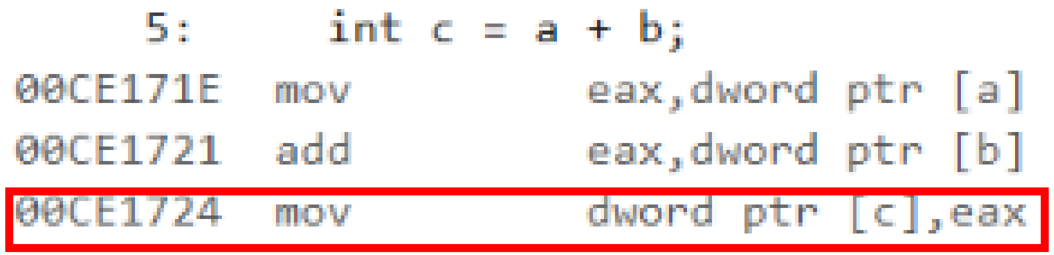


```
     4: int adder(int a, int b) {
00CE1700  push         ebp
00CE1701  mov          ebp,esp
00CE1703  sub          esp,0CCh
```

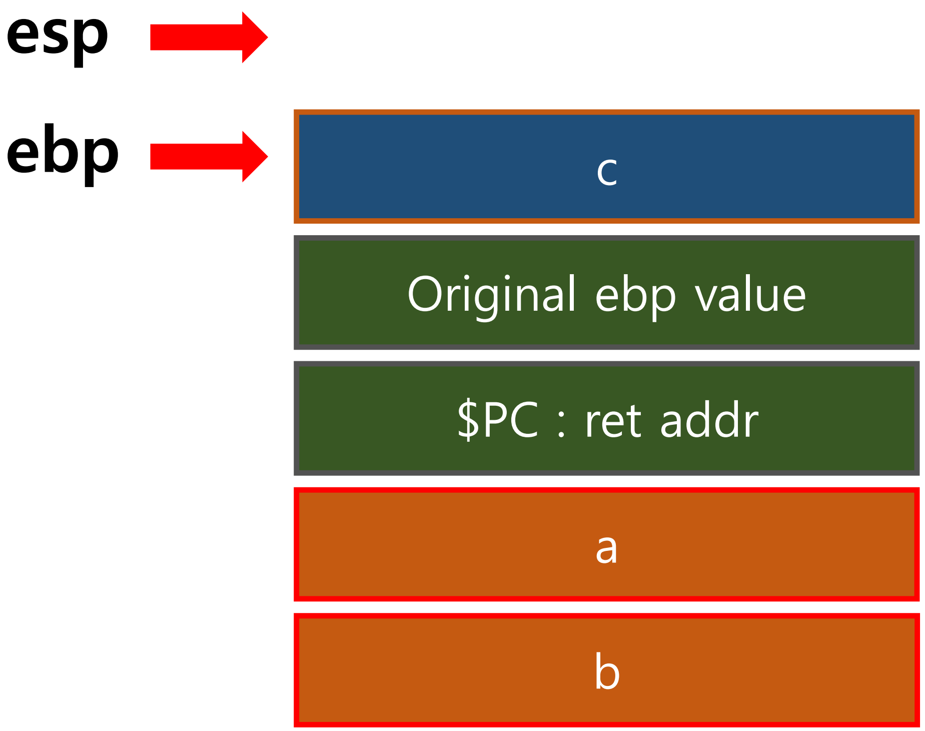
지역 변수 공간 미리 확보



Stack frame

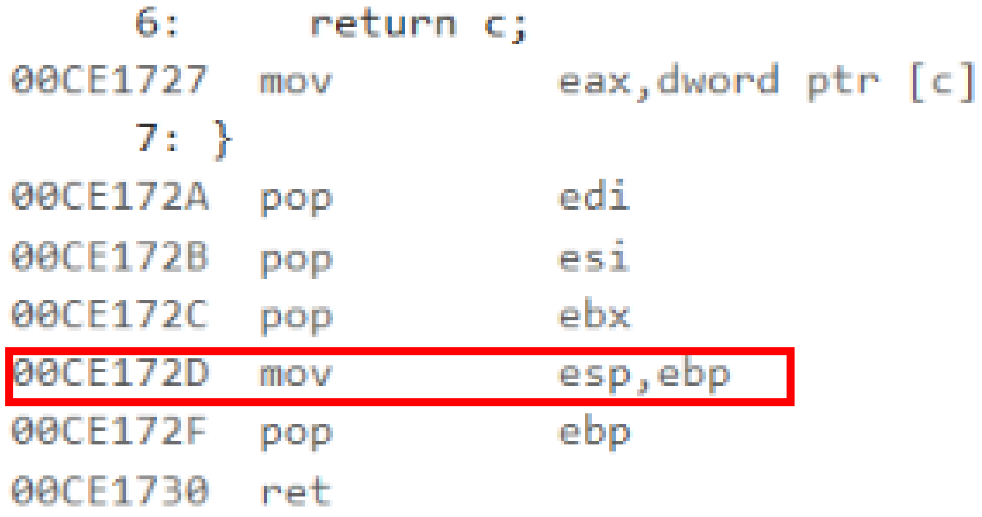


```
     5:       int c = a + b;
00CE171E  mov          eax,dword ptr [a]
00CE1721  add          eax,dword ptr [b]
00CE1724  mov          dword ptr [c],eax
```



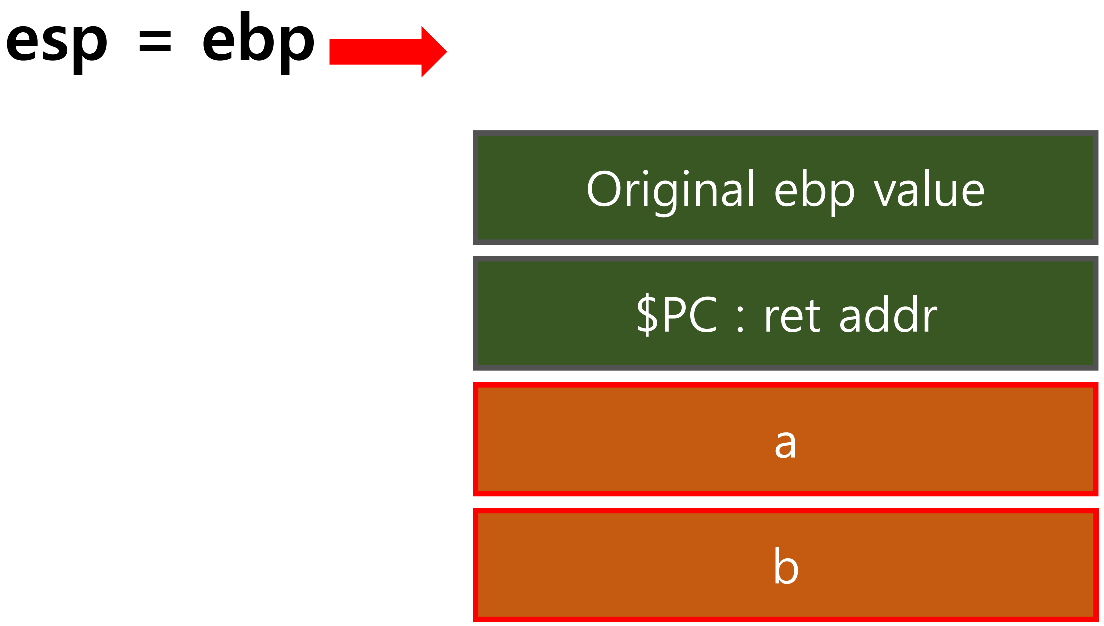
Stack frame

함수 호출이 끝나고 STACK에서 해지되는 순서



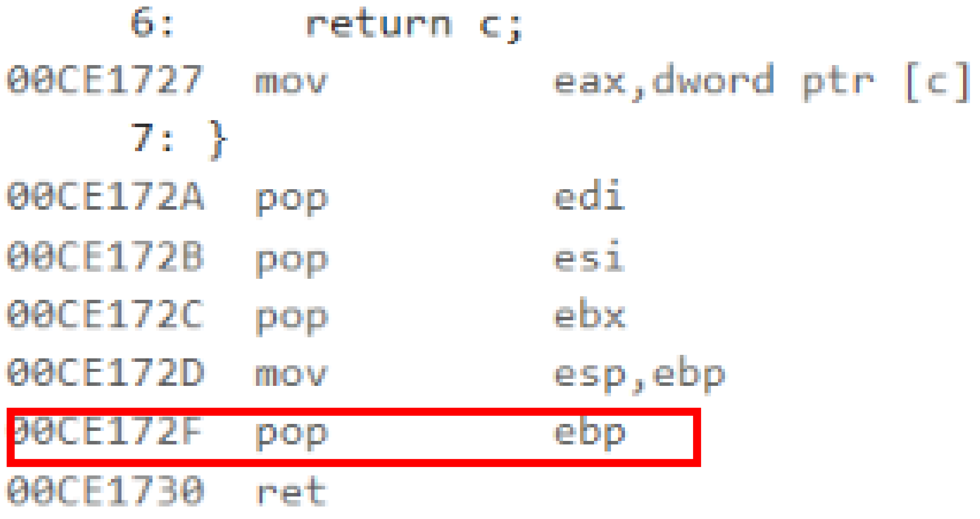
```
     6:      return c;
00CE1727   mov          eax,dword ptr [c]
     7: }
00CE172A   pop          edi
00CE172B   pop          esi
00CE172C   pop          ebx
00CE172D   mov          esp,ebp
00CE172F   pop          ebp
00CE1730   ret
```

실제 지우지는 않지만 stack pointer가 움직인 것이
결국엔 해지

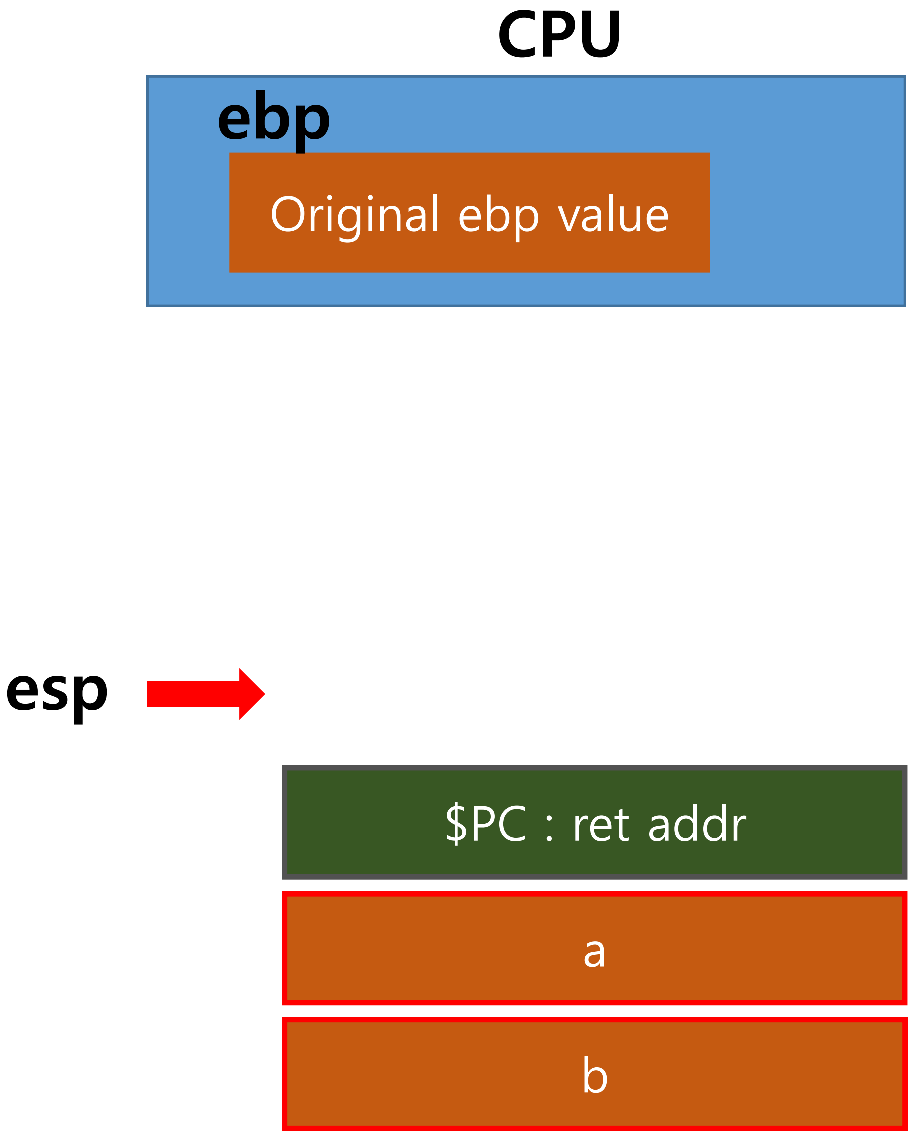


Stack frame

함수 호출이 끝나고 STACK에서 해지되는 순서

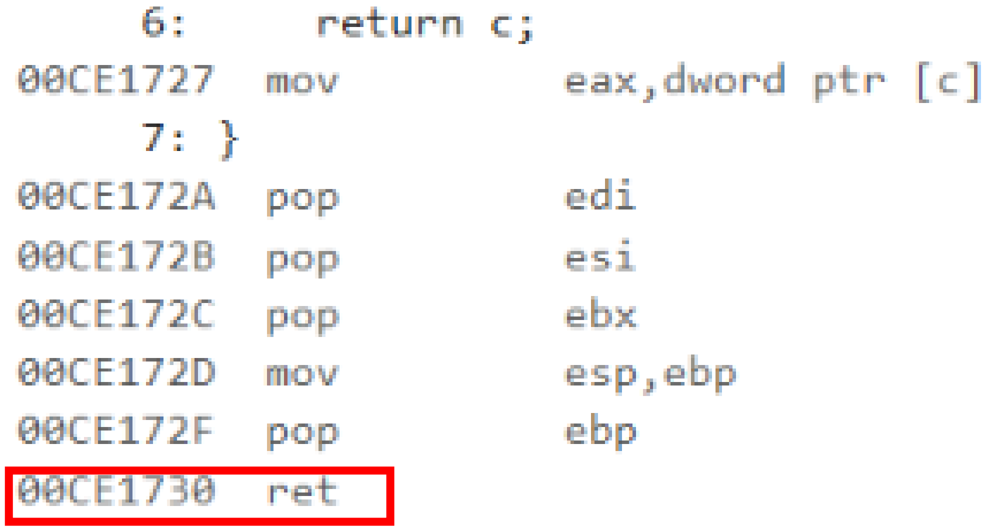


```
     6:     return c;
00CE1727  mov          eax,dword ptr [c]
     7: }
00CE172A  pop          edi
00CE172B  pop          esi
00CE172C  pop          ebx
00CE172D  mov          esp,ebp
00CE172F  pop          ebp
00CE1730  ret
```

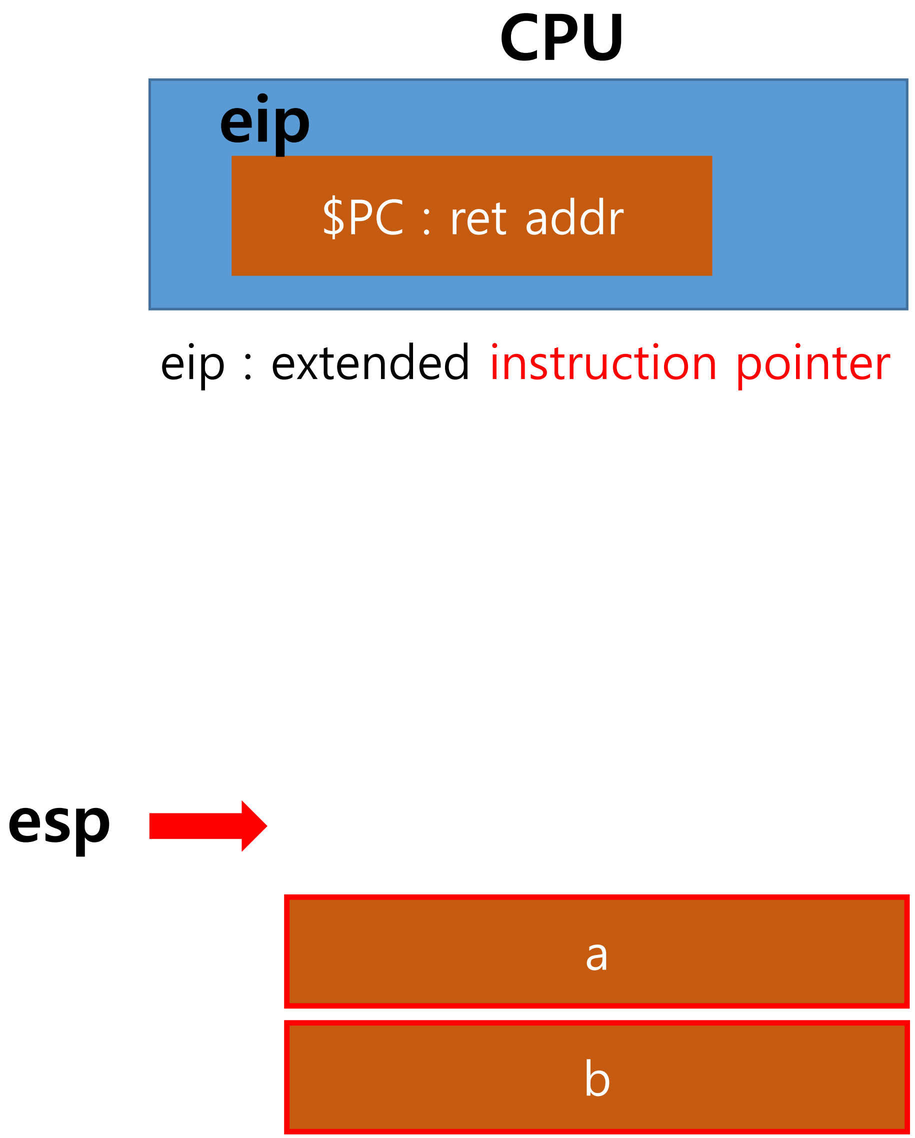


Stack frame

함수 호출이 끝나고 STACK에서 해지되는 순서



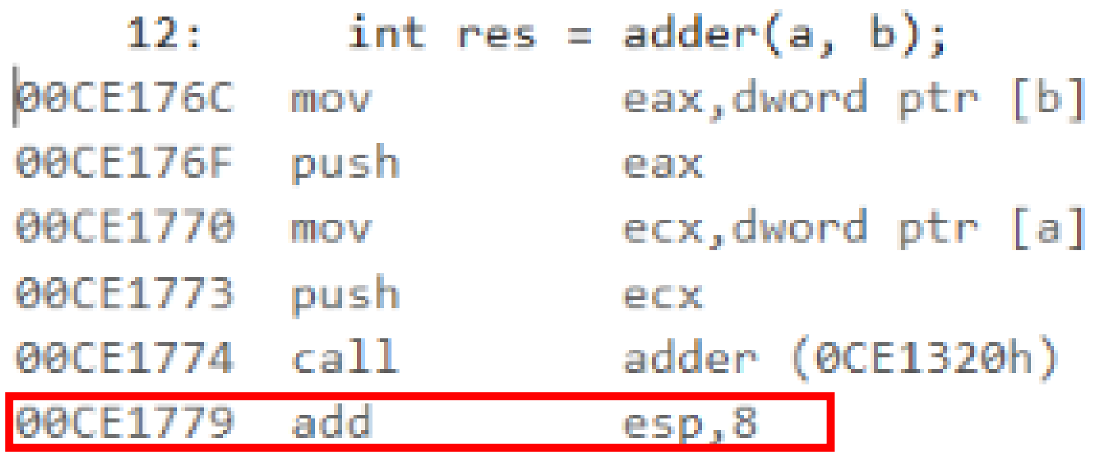
```
      6:      return c;
00CE1727  mov         eax,dword ptr [c]
      7: }
00CE172A  pop         edi
00CE172B  pop         esi
00CE172C  pop         ebx
00CE172D  mov         esp,ebp
00CE172F  pop         ebp
00CE1730  ret
```



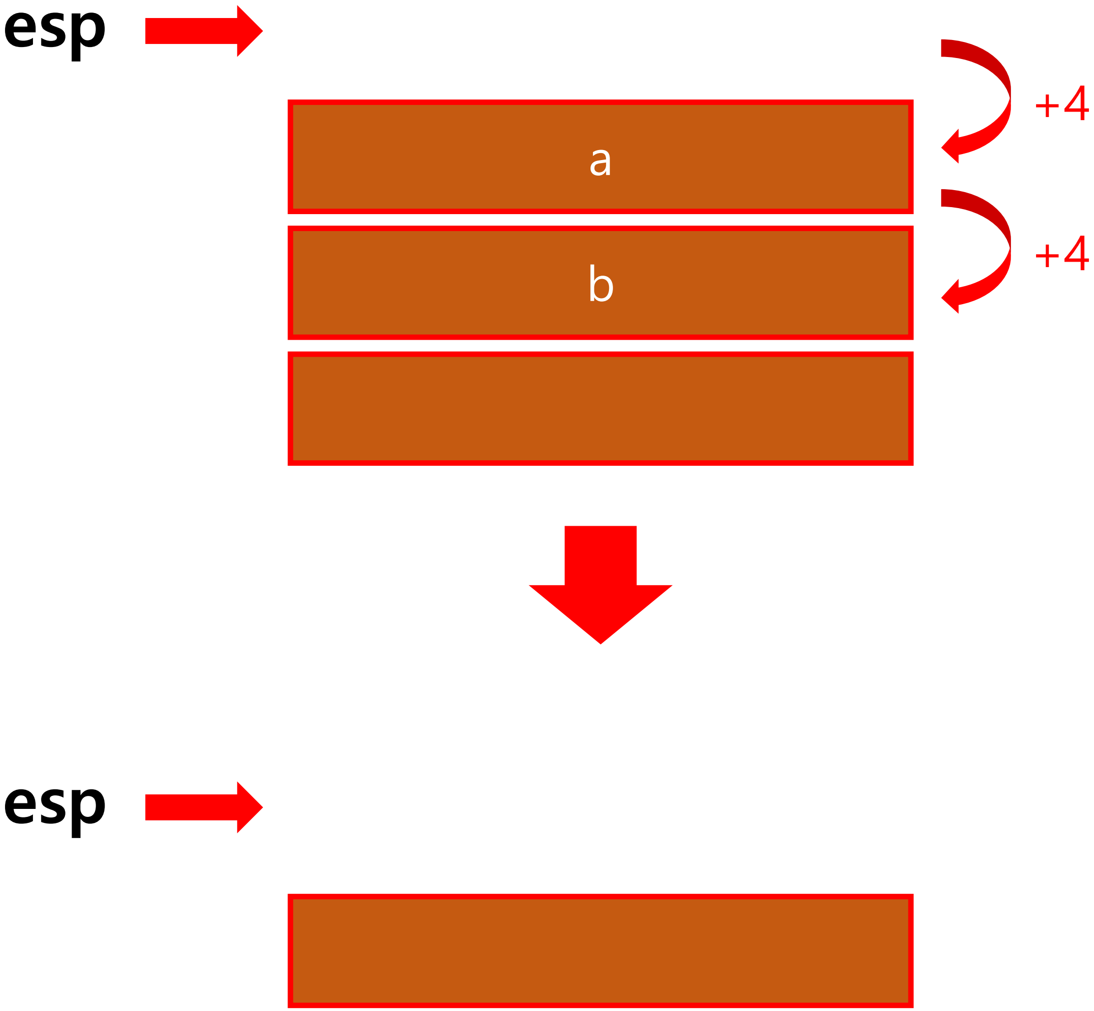
eip : extended instruction pointer

Stack frame

함수 호출이 끝나고 STACK에서 해지되는 순서



```
    12:        int res = adder(a, b);
00CE176C   mov          eax,dword ptr [b]
00CE176F   push         eax
00CE1770   mov          ecx,dword ptr [a]
00CE1773   push         ecx
00CE1774   call         adder (0CE1320h)
00CE1779   add          esp,8
```


esp
+4
a
+4
b
esp