

# Stack & Function call

# Simple source code

```
int f1(int a, int b) {  
    volatile int c;  
        c=a+b;  
        return c;  
}
```

```
int main() {  
    volatile int x;  
    volatile int a, b, c, d;  
        a = 0x1111; b = 0x2222; c = 0x3333; d = 0x4444;  
        x=f1(a,b);  
        ...  
}
```

[illegible]

← ESP

0x80482e0 &lt;main&gt;:

push %ebp

```
0x80482e1 <main+1>:      mov    %esp,%ebp
```

```
0x80482e3 <main+3>:      sub    $0x20,%esp
```

```
0x80482e6 <main+6>:    movl  $0x1111,-0x10(%ebp)
```

```
0x80482ed <main+13>:    movl  $0x2222,-0xc(%ebp)
```

```
0x80482f4 <main+20>:    movl  $0x3333,-0x8(%ebp)
```

```
0x80482fb <main+27>:    movl  $0x4444,-0x4(%ebp)
```

```
0x8048302 <main+34>:  mov  -0xc(%ebp),%edx
```

```
0x8048305 <main+37>:  mov  -0x10(%ebp),%eax
```

```
0x8048308 <main+40>:    push    %edx
```

```
0x8048309 <main+41>:    push  %eax
```

```
0x804830a <main+42>:    call 0x804843b <f1>
```

```
0x804830f <main+47>:  mov    %eax,-0x14(%ebp)
```

```
0x8048312 <main+50>:  mov  -0x14(%ebp),%eax
```

```
0x8048315 <main+53>:    pop    %ecx
```

```
0x8048316 <main+54>:    pop    %edx
```

```
0x8048317 <main+55>:    cmp    $0x7,%eax
```

esp	ff	ff	d6	e8
ebp	??	??	??	??

[illegible]

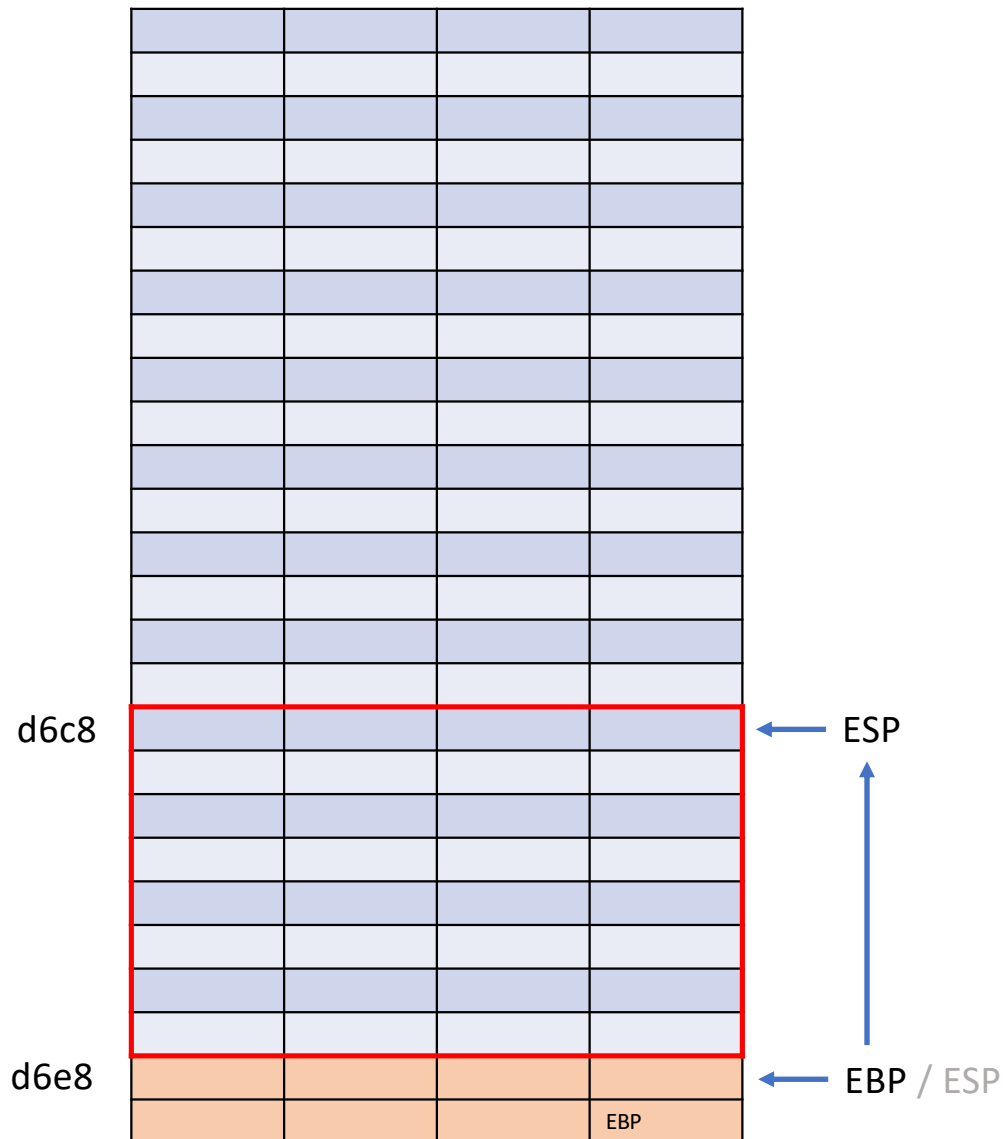
← ESP / EBP

```

0x80482e0 <main>:      push    %ebp
0x80482e1 <main+1>:    mov     %esp,%ebp
0x80482e3 <main+3>:    sub     $0x20,%esp
0x80482e6 <main+6>:    movl    $0x1111,-0x10(%ebp)
0x80482ed <main+13>:   movl    $0x2222,-0xc(%ebp)
0x80482f4 <main+20>:   movl    $0x3333,-0x8(%ebp)
0x80482fb <main+27>:   movl    $0x4444,-0x4(%ebp)
0x8048302 <main+34>:   mov     -0xc(%ebp),%edx
0x8048305 <main+37>:   mov     -0x10(%ebp),%eax
0x8048308 <main+40>:   push    %edx
0x8048309 <main+41>:   push    %eax
0x804830a <main+42>:   call   0x804843b <f1>
0x804830f <main+47>:   mov     %eax,-0x14(%ebp)
0x8048312 <main+50>:   mov     -0x14(%ebp),%eax
0x8048315 <main+53>:   pop     %ecx
0x8048316 <main+54>:   pop     %edx
0x8048317 <main+55>:   cmp     $0x7,%eax

```

<b>esp</b>	ff	ff	d6	e8
<b>ebp</b>	ff	ff	d6	e8



```

0x80482e0 <main>:      push  %ebp
0x80482e1 <main+1>:    mov   %esp,%ebp
0x80482e3 <main+3>:    sub   $0x20,%esp
0x80482e6 <main+6>:    movl  $0x1111,-0x10(%ebp)
0x80482ed <main+13>:   movl  $0x2222,-0xc(%ebp)
0x80482f4 <main+20>:   movl  $0x3333,-0x8(%ebp)
0x80482fb <main+27>:   movl  $0x4444,-0x4(%ebp)
0x8048302 <main+34>:   mov   -0xc(%ebp),%edx
0x8048305 <main+37>:   mov   -0x10(%ebp),%eax
0x8048308 <main+40>:   push  %edx
0x8048309 <main+41>:   push  %eax
0x804830a <main+42>:   call 0x804843b <f1>
0x804830f <main+47>:   mov   %eax,-0x14(%ebp)
0x8048312 <main+50>:   mov   -0x14(%ebp),%eax
0x8048315 <main+53>:   pop   %ecx
0x8048316 <main+54>:   pop   %edx
0x8048317 <main+55>:   cmp   $0x7,%eax

```

esp	ff	ff	d6	c8
ebp	ff	ff	d6	e8

d6c8				← ESP
-0x10	00	00	11	11
-0x0c	00	00	22	22
-0x08	00	00	33	33
-0x04	00	00	44	44
d6e8				← EBP

ESP

-0x10

a

-0x0c

**b**

-0x08

C

-0x04

**a**

d6e8

EBP

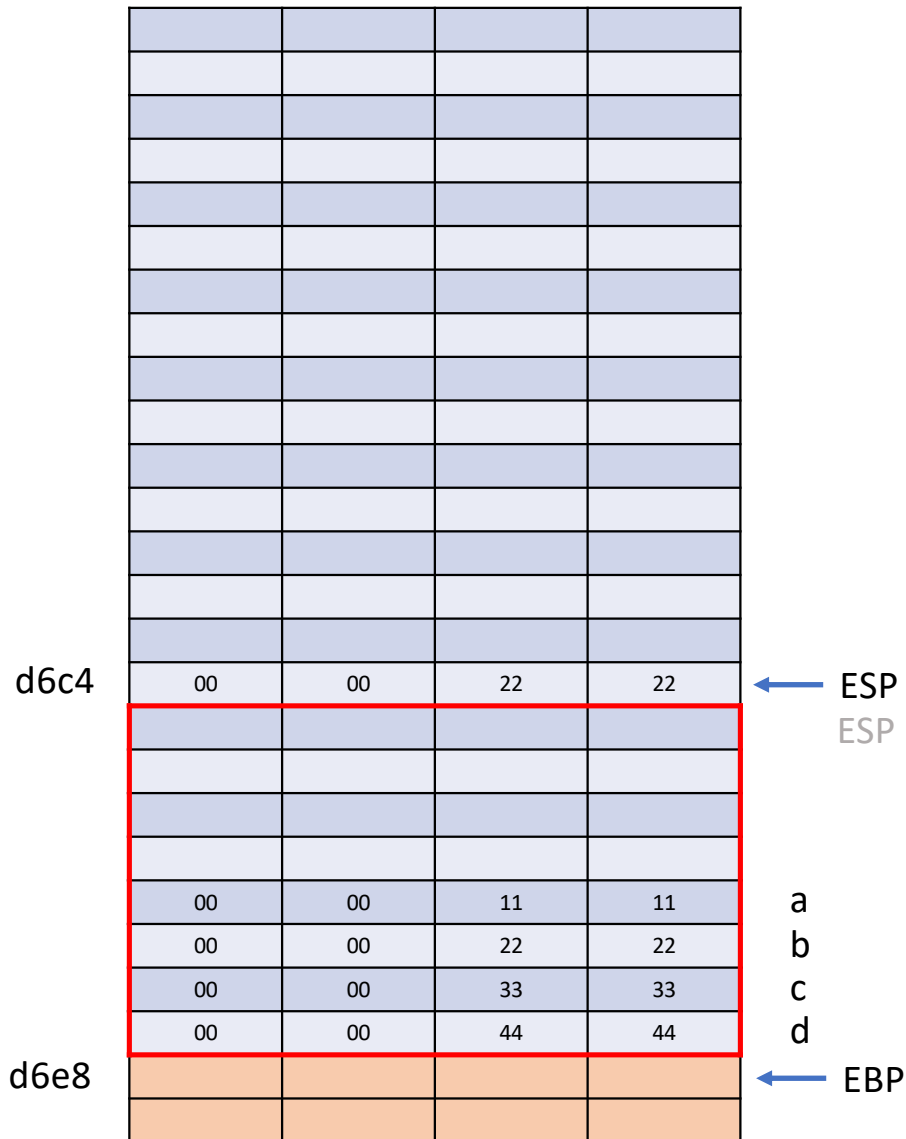
```

0x80482e0 <main>:      push  %ebp
0x80482e1 <main+1>:    mov   %esp,%ebp
0x80482e3 <main+3>:    sub   $0x20,%esp
0x80482e6 <main+6>:    movl  $0x1111,-0x10(%ebp)
0x80482ed <main+13>:   movl  $0x2222,-0xc(%ebp)
0x80482f4 <main+20>:   movl  $0x3333,-0x8(%ebp)
0x80482fb <main+27>:   movl  $0x4444,-0x4(%ebp)
0x8048302 <main+34>:   mov   -0xc(%ebp),%edx
0x8048305 <main+37>:   mov   -0x10(%ebp),%eax
0x8048308 <main+40>:   push  %edx
0x8048309 <main+41>:   push  %eax
0x804830a <main+42>:   call 0x804843b <f1>
0x804830f <main+47>:   mov   %eax,-0x14(%ebp)
0x8048312 <main+50>:   mov   -0x14(%ebp),%eax
0x8048315 <main+53>:   pop   %ecx
0x8048316 <main+54>:   pop   %edx
0x8048317 <main+55>:   cmp   $0x7,%eax

```

esp	ff	ff	d6	c8
ebp	ff	ff	d6	E8
edx				
eax				





```
0x80482e0 <main>:      push  %ebp
0x80482e1 <main+1>:    mov   %esp,%ebp
0x80482e3 <main+3>:    sub   $0x20,%esp
0x80482e6 <main+6>:    movl  $0x1111,-0x10(%ebp)
0x80482ed <main+13>:   movl  $0x2222,-0xc(%ebp)
0x80482f4 <main+20>:   movl  $0x3333,-0x8(%ebp)
0x80482fb <main+27>:   movl  $0x4444,-0x4(%ebp)
0x8048302 <main+34>:   mov   -0xc(%ebp),%edx
0x8048305 <main+37>:   mov   -0x10(%ebp),%eax
0x8048308 <main+40>:   push  %edx
0x8048309 <main+41>:   push  %eax
0x804830a <main+42>:   call 0x804843b <f1>
0x804830f <main+47>:   mov   %eax,-0x14(%ebp)
0x8048312 <main+50>:   mov   -0x14(%ebp),%eax
0x8048315 <main+53>:   pop   %ecx
0x8048316 <main+54>:   pop   %edx
0x8048317 <main+55>:   cmp   $0x7,%eax
```

esp	ff	ff	d6	c4
ebp	ff	ff	d6	E8
eax	00	00	11	11
edx	00	00	22	22





d6bc	08	04	83	0f
	00	00	11	11
	00	00	22	22
	00	00	11	11
	00	00	22	22
	00	00	33	33
	00	00	44	44
d6e8				

← ESP

ESP

a

b

c

d

← EBP

```

0x80482e0 <main>:      push  %ebp
0x80482e1 <main+1>:    mov   %esp,%ebp
0x80482e3 <main+3>:    sub   $0x20,%esp
0x80482e6 <main+6>:    movl  $0x1111,-0x10(%ebp)
0x80482ed <main+13>:   movl  $0x2222,-0xc(%ebp)
0x80482f4 <main+20>:   movl  $0x3333,-0x8(%ebp)
0x80482fb <main+27>:   movl  $0x4444,-0x4(%ebp)
0x8048302 <main+34>:   mov   -0xc(%ebp),%edx
0x8048305 <main+37>:   mov   -0x10(%ebp),%eax
0x8048308 <main+40>:   push  %edx
0x8048309 <main+41>:   push  %eax
0x804830a <main+42>:   call 0x804843b <f1>
0x804830f <main+47>:   mov   %eax,-0x14(%ebp)
0x8048312 <main+50>:   mov   -0x14(%ebp),%eax
0x8048315 <main+53>:   pop   %ecx
0x8048316 <main+54>:   pop   %edx
0x8048317 <main+55>:   cmp   $0x7,%eax

```

esp	ff	ff	d6	bc
ebp	ff	ff	d6	e8
eax	00	00	11	11
edx	00	00	22	22

d6b8	ff	ff	d6	e8
	08	04	83	0f
	00	00	11	11
	00	00	22	22
	00	00	11	11
	00	00	22	22
	00	00	33	33
	00	00	44	44
d6e8				

← ESP

ESP

a  
b  
c  
d

← EBP

0x804843b <f1>:

0x804843c <f1+1>:

0x804843e <f1+3>:

0x8048441 <f1+6>:

0x8048444 <f1+9>:

0x8048447 <f1+12>:

0x804844a <f1+15>:

0x804844d <f1+18>:

0x804844e <f1+19>:

push %ebp

mov %esp,%ebp

sub \$0x10,%esp

mov 0xc(%ebp),%eax

add 0x8(%ebp),%eax

mov %eax,-0x4(%ebp)

mov -0x4(%ebp),%eax

leave

ret

esp	ff	ff	d6	b8
ebp	ff	ff	d6	e8
eax	00	00	11	11
edx	00	00	22	22

d6b8	ff	ff	d6	e8
	08	04	83	0f
	00	00	11	11
	00	00	22	22
	00	00	11	11
	00	00	22	22
	00	00	33	33
	00	00	44	44
d6e8				

← ESP / EBP

a  
b  
c  
d

EBP

d6e8

← ESP / EBP

a  
b  
c  
d

EBP

```
0x804843b <f1>:      push    %ebp
0x804843c <f1+1>:     mov     %esp,%ebp
0x804843e <f1+3>:     sub     $0x10,%esp
0x8048441 <f1+6>:     mov     0xc(%ebp),%eax
0x8048444 <f1+9>:     add     0x8(%ebp),%eax
0x8048447 <f1+12>:    mov     %eax,-0x4(%ebp)
0x804844a <f1+15>:    mov     -0x4(%ebp),%eax
0x804844d <f1+18>:    leave
0x804844e <f1+19>:    ret
```

0x804843c <f1+1>:

0x804843e &lt;f1+3&gt;:

0x8048441 &lt;f1+6&gt;:

0x8048444 &lt;f1+9&gt;:

0x8048447 &lt;f1+12&gt;:

0x804844a &lt;f1+15&gt;:

0x804844d &lt;f1+18&gt;:

0x804844e &lt;f1+19&gt;:

push %ebp

```
mov    %esp,%ebp
```

```
sub    $0x10,%esp
```

```
mov 0xc(%ebp),%eax
```

```
add    0x8(%ebp),%eax
```

```
mov    %eax,-0x4(%ebp)
```

```
mov    -0x4(%ebp),%eax
```

leave

ret

esp	ff	ff	d6	b8
ebp	ff	ff	d6	b8
eax	00	00	11	11
edx	00	00	22	22

d6a8				← ESP
				↑
d6b8	ff	ff	d6	e8 ← EBP
	08	04	83	0f
	00	00	11	11
	00	00	22	22
	00	00	11	11
	00	00	22	22
	00	00	33	33
	00	00	44	44
d6e8				

a  
b  
c  
d

```

0x804843b <f1>:      push  %ebp
0x804843c <f1+1>:    mov   %esp,%ebp
0x804843e <f1+3>:    sub   $0x10,%esp
0x8048441 <f1+6>:    mov   0xc(%ebp),%eax
0x8048444 <f1+9>:    add   0x8(%ebp),%eax
0x8048447 <f1+12>:   mov   %eax,-0x4(%ebp)
0x804844a <f1+15>:   mov   -0x4(%ebp),%eax
0x804844d <f1+18>:   leave
0x804844e <f1+19>:   ret

```

esp	ff	ff	d6	a8
ebp	ff	ff	d6	b8
eax	00	00	11	11
edx	00	00	22	22

d6a8				← ESP
d6b8	ff	ff	d6	e8 ← EBP
	08	04	83	0f
	00	00	11	11
	00	00	22	22
	00	00	11	11
	00	00	22	22
	00	00	33	33
	00	00	44	44
d6e8				

```

0x804843b <f1>:      push  %ebp
0x804843c <f1+1>:    mov   %esp,%ebp
0x804843e <f1+3>:    sub   $0x10,%esp
0x8048441 <f1+6>:    mov   0xc(%ebp),%eax
0x8048444 <f1+9>:    add   0x8(%ebp),%eax
0x8048447 <f1+12>:   mov   %eax,-0x4(%ebp)
0x804844a <f1+15>:   mov   -0x4(%ebp),%eax
0x804844d <f1+18>:   leave
0x804844e <f1+19>:   ret

```

a  
b  
c  
d

esp	ff	ff	d6	a8
ebp	ff	ff	d6	b8
eax	00	00	22	22
edx	00	00	22	22

d6a8				← ESP
d6b8	ff	ff	d6	e8 ← EBP
	08	04	83	0f
	00	00	11	11
	00	00	22	22
	00	00	11	11
	00	00	22	22
	00	00	33	33
	00	00	44	44
d6e8				

```

0x804843b <f1>:      push %ebp
0x804843c <f1+1>:    mov  %esp,%ebp
0x804843e <f1+3>:    sub  $0x10,%esp
0x8048441 <f1+6>:    mov  0xc(%ebp),%eax
0x8048444 <f1+9>:    add  0x8(%ebp),%eax
0x8048447 <f1+12>:   mov  %eax,-0x4(%ebp)
0x804844a <f1+15>:   mov  -0x4(%ebp),%eax
0x804844d <f1+18>:   leave
0x804844e <f1+19>:   ret

```

a  
b  
c  
d

esp	ff	ff	d6	a8
ebp	ff	ff	d6	b8
eax	00	00	33	33
edx	00	00	22	22

d6a8				← ESP
d6b8	00	00	33	33
	ff	ff	d6	e8
	08	04	83	0f
	00	00	11	11
	00	00	22	22
	00	00	11	11
	00	00	22	22
	00	00	33	33
	00	00	44	44
d6e8				

```

0x804843b <f1>:
0x804843c <f1+1>:
0x804843e <f1+3>:
0x8048441 <f1+6>:
0x8048444 <f1+9>:
0x8048447 <f1+12>:
0x804844a <f1+15>:
0x804844d <f1+18>:
0x804844e <f1+19>:

```

```

push %ebp
mov  %esp,%ebp
sub  $0x10,%esp
mov  0xc(%ebp),%eax
add  0x8(%ebp),%eax
mov  %eax,-0x4(%ebp)
mov  -0x4(%ebp),%eax
leave
ret

```

esp	ff	ff	d6	a8
ebp	ff	ff	d6	b8
eax	00	00	33	33
edx	00	00	22	22



d6a8				← ESP
	00	00	33	33
d6b8	ff	ff	d6	e8
	08	04	83	0f
	00	00	11	11
	00	00	22	22
	00	00	11	11
	00	00	22	22
	00	00	33	33
	00	00	44	44
d6e8				

a  
b  
c  
d

```

0x804843b <f1>:      push %ebp
0x804843c <f1+1>:    mov  %esp,%ebp
0x804843e <f1+3>:    sub  $0x10,%esp
0x8048441 <f1+6>:    mov  0xc(%ebp),%eax
0x8048444 <f1+9>:    add  0x8(%ebp),%eax
0x8048447 <f1+12>:   mov  %eax,-0x4(%ebp)
0x804844a <f1+15>:   mov  -0x4(%ebp),%eax
0x804844d <f1+18>:   leave
0x804844e <f1+19>:   ret

```

esp	ff	ff	d6	a8
ebp	ff	ff	d6	b8
eax	00	00	33	33
edx	00	00	22	22

d6a8				← ESP
	00	00	33	33
d6b8	ff	ff	d6	e8
d6bc	08	04	83	0f
	00	00	11	11
	00	00	22	22
	00	00	11	11
	00	00	22	22
	00	00	33	33
	00	00	44	44
d6e8				

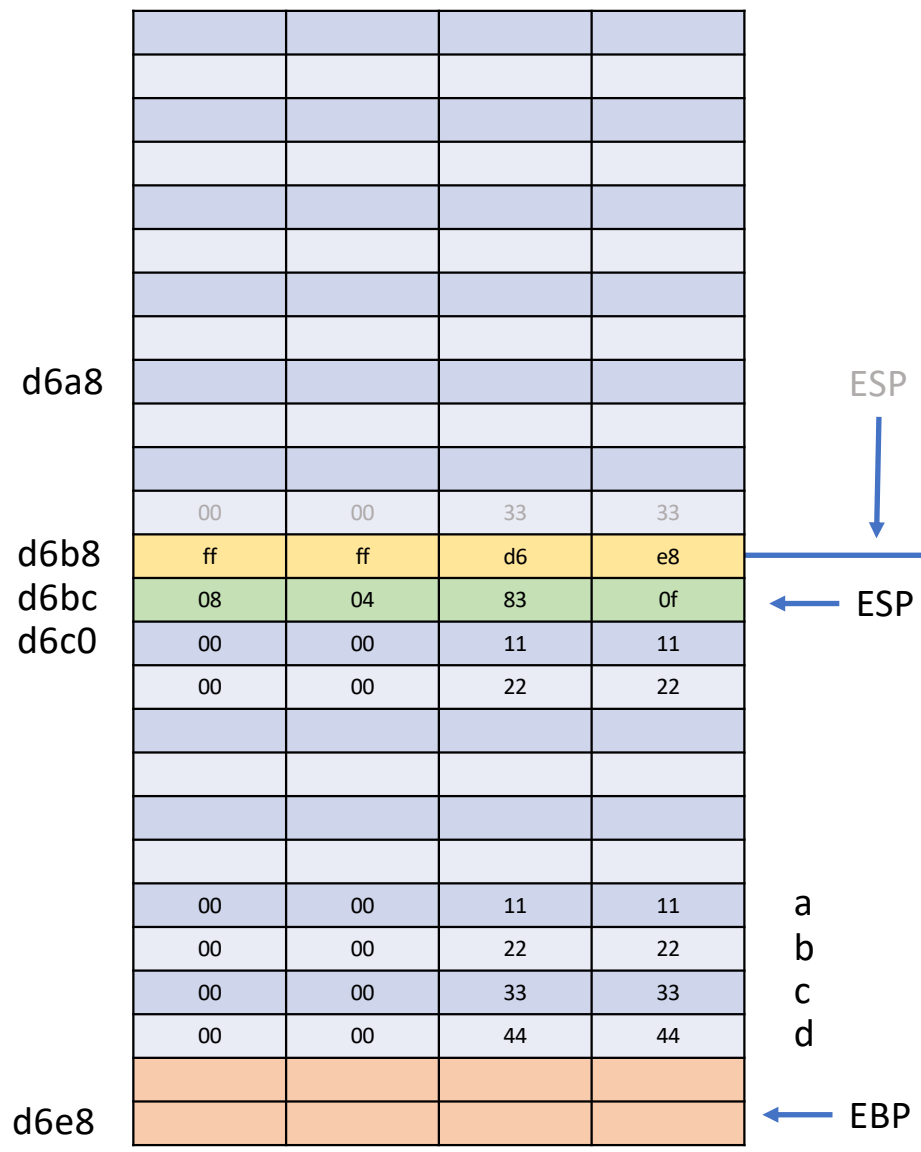
```

0x804843b <f1>:      push  %ebp
0x804843c <f1+1>:    mov   %esp,%ebp
0x804843e <f1+3>:    sub   $0x10,%esp
0x8048441 <f1+6>:    mov   0xc(%ebp),%eax
0x8048444 <f1+9>:    add   0x8(%ebp),%eax
0x8048447 <f1+12>:   mov   %eax,-0x4(%ebp)
0x804844a <f1+15>:   mov   -0x4(%ebp),%eax
0x804844d <f1+18>:   leave
0x804844e <f1+19>:   ret

```

esp	ff	ff	d6	a8
ebp	ff	ff	d6	b8
eax	00	00	33	33
edx	00	00	22	22
eip	08	04	84	4d

0x804843b <f1>:	push %ebp
0x804843c <f1+1>:	mov %esp,%ebp
0x804843e <f1+3>:	sub \$0x10,%esp
0x8048441 <f1+6>:	mov 0xc(%ebp),%eax
0x8048444 <f1+9>:	add 0x8(%ebp),%eax
0x8048447 <f1+12>:	mov %eax,-0x4(%ebp)
0x804844a <f1+15>:	mov -0x4(%ebp),%eax
0x804844d <f1+18>:	leave
0x804844e <f1+19>:	ret



esp	ff	ff	d6	bc
ebp	ff	ff	d6	e8
eax	00	00	33	33
edx	00	00	22	22
eip	08	04	83	4e

	00	00	33 33
d6b8	ff	ff	d6 e8
	08	04	83 0f
d6c0	00	00	11 11
	00	00	22 22
d6c8			
	00	00	33 33
	00	00	11 11
	00	00	22 22
	00	00	33 33
	00	00	44 44
d6e8			

← ESP

x  
a  
b  
c  
d

← EBP

```

0x80482e0 <main>:      push  %ebp
0x80482e1 <main+1>:    mov   %esp,%ebp
0x80482e3 <main+3>:    sub   $0x20,%esp
0x80482e6 <main+6>:    movl  $0x1111,-0x10(%ebp)
0x80482ed <main+13>:   movl  $0x2222,-0xc(%ebp)
0x80482f4 <main+20>:   movl  $0x3333,-0x8(%ebp)
0x80482fb <main+27>:   movl  $0x4444,-0x4(%ebp)
0x8048302 <main+34>:   mov   -0xc(%ebp),%edx
0x8048305 <main+37>:   mov   -0x10(%ebp),%eax
0x8048308 <main+40>:   push  %edx
0x8048309 <main+41>:   push  %eax
0x804830a <main+42>:   call 0x804843b <f1>
0x804830f <main+47>:   mov   %eax,-0x14(%ebp)
0x8048312 <main+50>:   mov   -0x14(%ebp),%eax
0x8048315 <main+53>:   pop   %ecx
0x8048316 <main+54>:   pop   %edx
0x8048317 <main+55>:   cmp   $0x7,%eax

```

esp	ff	ff	d6	c0
ebp	ff	ff	d6	e8
eax	00	00	33	33
edx	00	00	22	22
eip	08	04	83	0f

	00	00	33	33
d6b8	ff	ff	d6	e8
	08	04	83	0f
d6c0	00	00	11	11
	00	00	22	22
d6c8				
	00	00	33	33
	00	00	11	11
	00	00	22	22
	00	00	33	33
	00	00	44	44
d6e8				

ESP  
 ↓  
 ESP  
 ←  
 ESP  
  
 a  
 b  
 c  
 d  
 ← EBP

```

0x80482e0 <main>:      push  %ebp
0x80482e1 <main+1>:    mov   %esp,%ebp
0x80482e3 <main+3>:    sub   $0x20,%esp
0x80482e6 <main+6>:    movl  $0x1111,-0x10(%ebp)
0x80482ed <main+13>:   movl  $0x2222,-0xc(%ebp)
0x80482f4 <main+20>:   movl  $0x3333,-0x8(%ebp)
0x80482fb <main+27>:   movl  $0x4444,-0x4(%ebp)
0x8048302 <main+34>:   mov   -0xc(%ebp),%edx
0x8048305 <main+37>:   mov   -0x10(%ebp),%eax
0x8048308 <main+40>:   push  %edx
0x8048309 <main+41>:   push  %eax
0x804830a <main+42>:   call 0x804843b <f1>
0x804830f <main+47>:   mov   %eax,-0x14(%ebp)
0x8048312 <main+50>:   mov   -0x14(%ebp),%eax
0x8048315 <main+53>:   pop   %ecx
0x8048316 <main+54>:   pop   %edx
0x8048317 <main+55>:   cmp   $0x7,%eax
  
```

esp	ff	ff	d6	c8
ebp	ff	ff	d6	e8
eax	00	00	33	33
edx	00	00	22	22
eip	08	04	83	0f