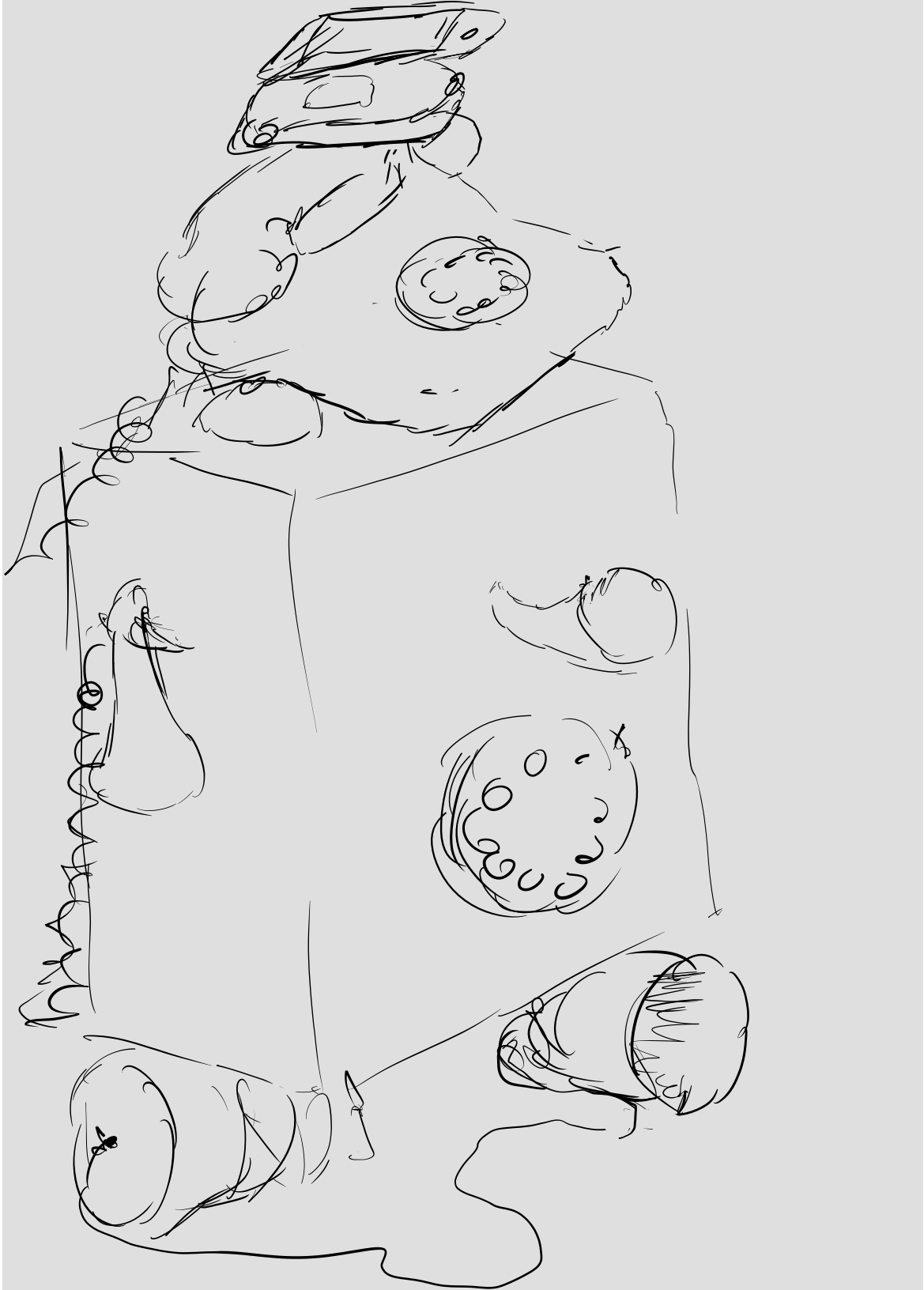
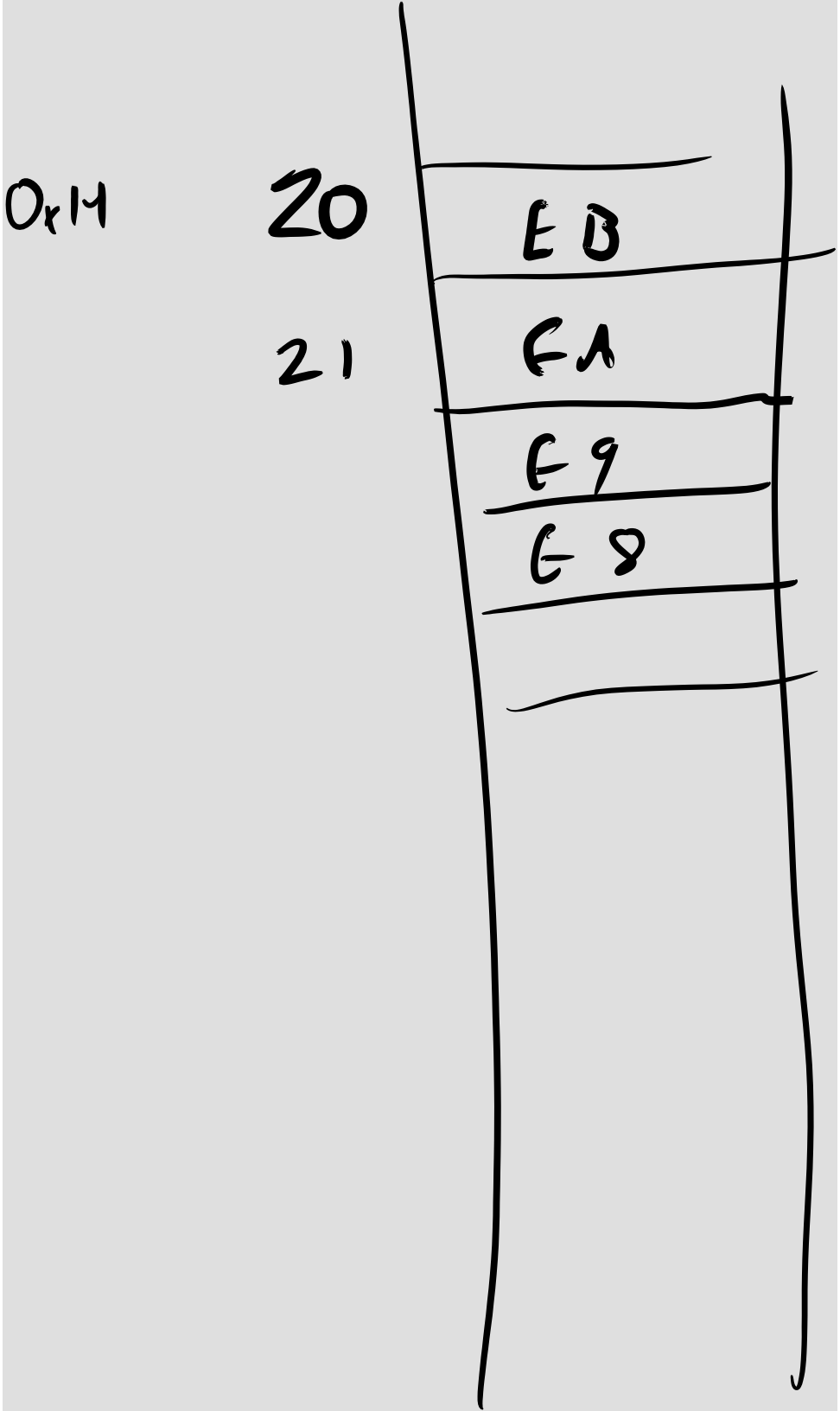



0x14
20
21
ED
FA
F9
E8

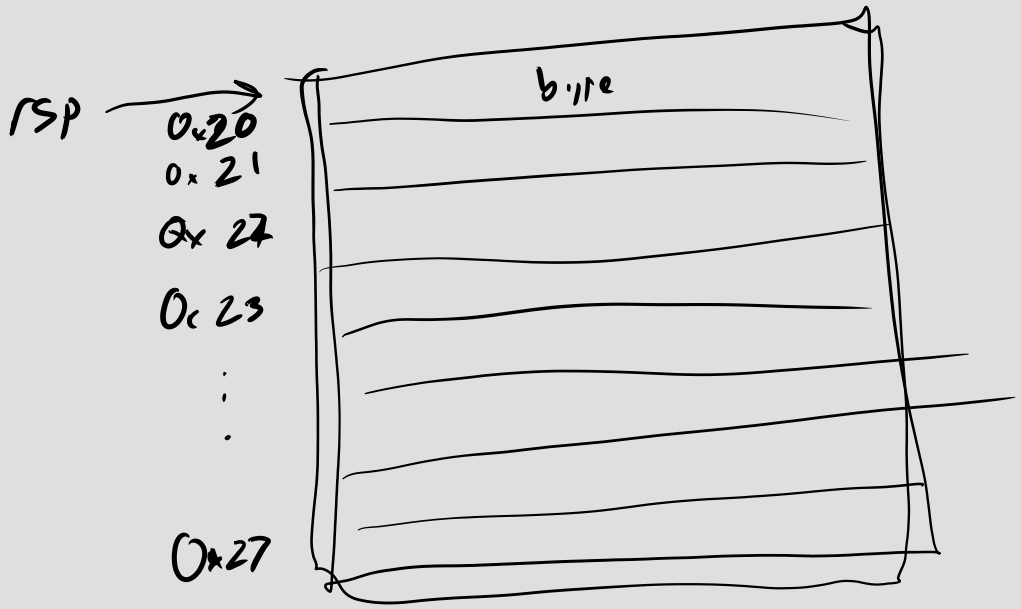
rsp

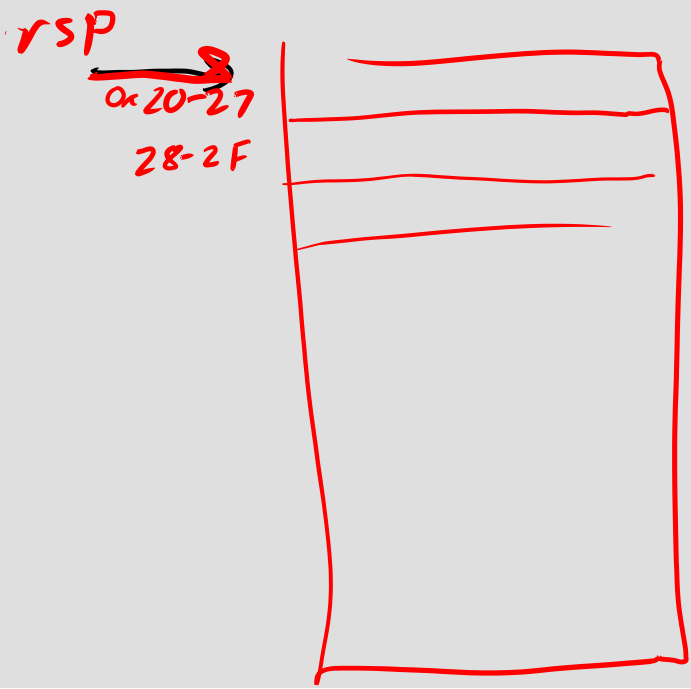
Pushq

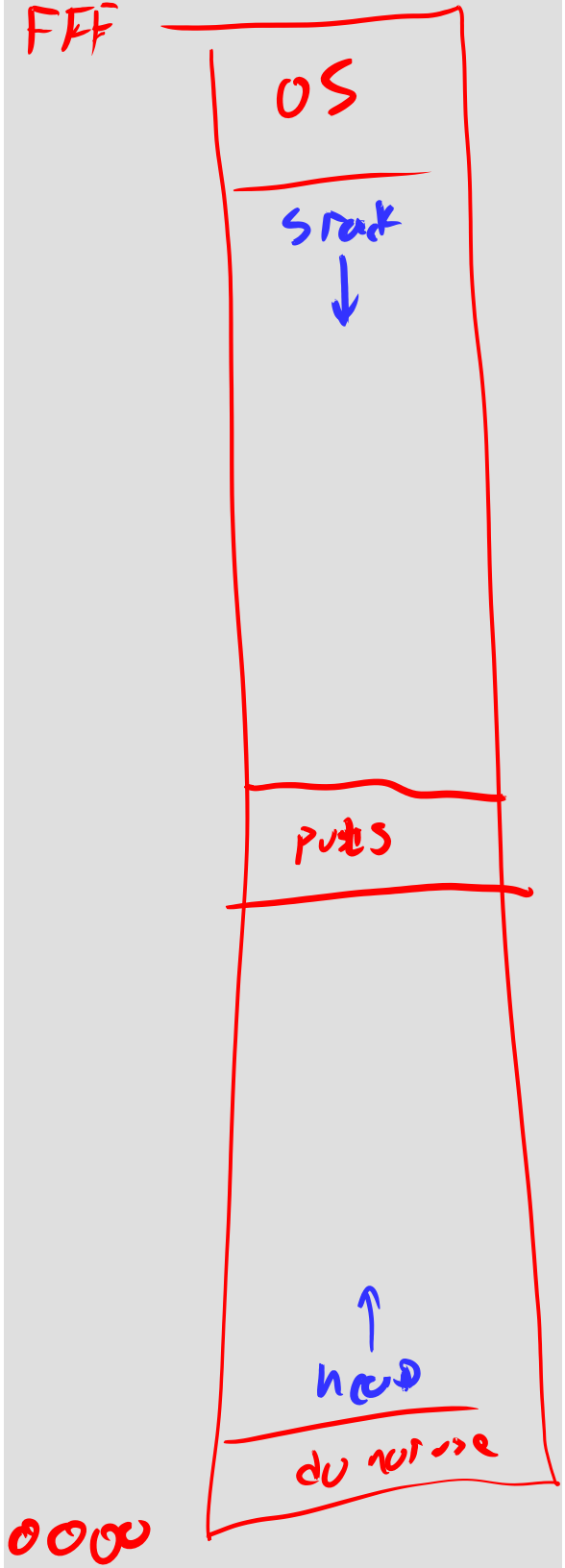
Popq

callq

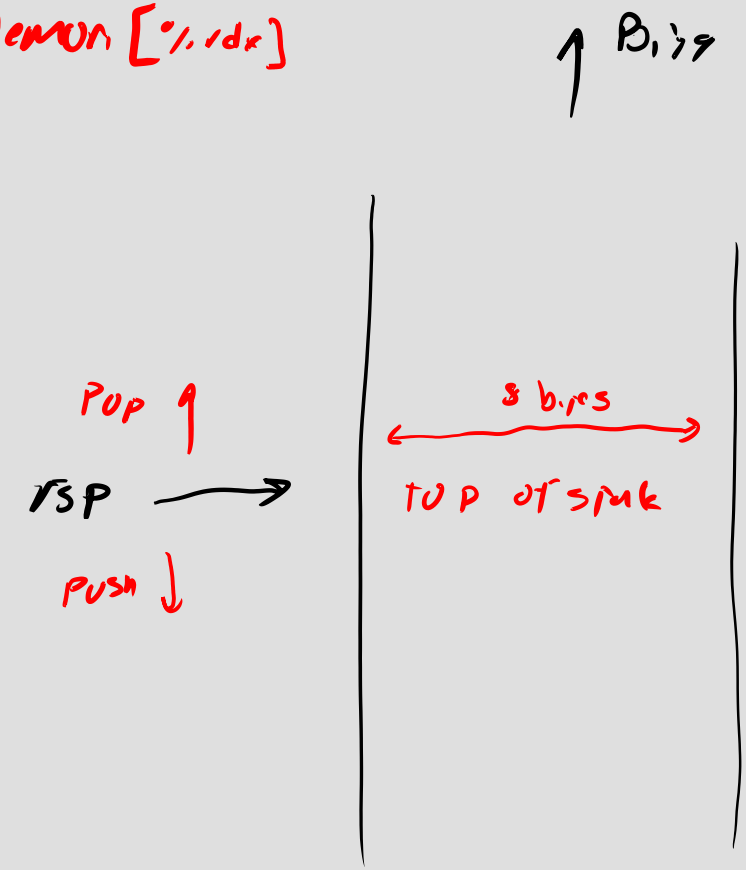
retq






FFF
OS
stack
puts
heap
du texte
0000

(%rdx) ↔ Memory[%rdx]



pushq x

```
subq $8, %rsp
movq x, (%rsp)
```

popq x

```
movq (%rsp), x
addq $8, %rsp
```

popq %rsi

Stack :

return address

```
push %rdi
push %rsi
:
movq 16(%rsp), %rax
```

```
foo() {
    push  push
    —
    =
    pop  pop
    ——————— ret
}
```

```
bar(){
    foo()  →  call q  foo

    foo()
}
```

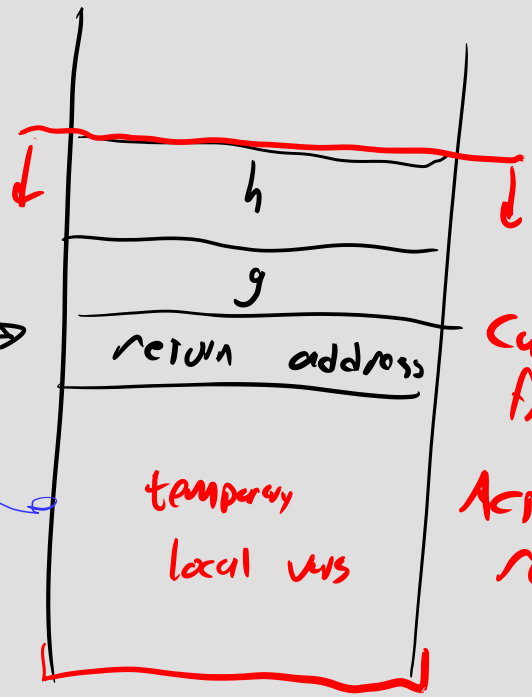
return value → %rax

Calling Convention

r8 r9 push Push

foo(a, b, c, d, e, f, g, h)

rdi rsi rdx rcx



Save reg → push

Caller Callee