CS208

F, 6 Feb 2024

function1: "caller-owned local var."

Callee pushq %rbx

Save the old value of rbx

.

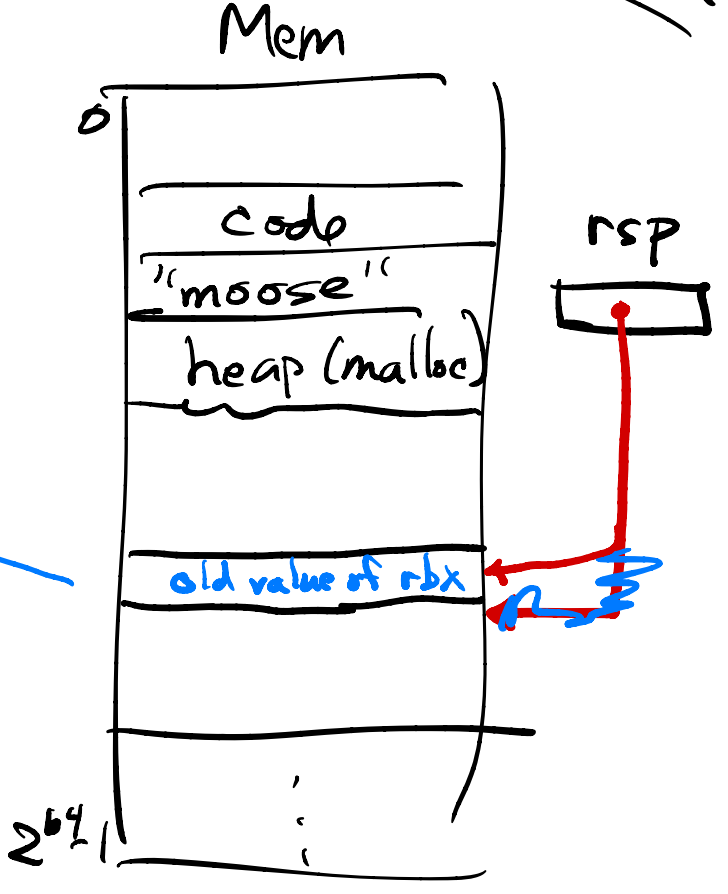
.

restore old value to rbx

→ popq %rbx

ret

Puzzle 2



function1(__*a, __*b)

Puzzle 3

```
testq %rdi, %rdi
sete  %al
testq %rsi, %rsi
sete  %dl
orb   %dl, %al
je    .L3
```

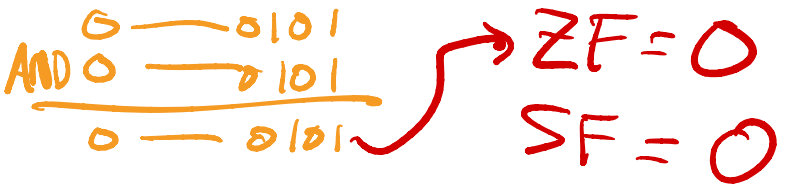
if ~~rdi~~ a == 0, al = 1, else al = 0

if ~~rsi~~ b = 0, dl = 1 else dl = 0

if !(a==0 || b==0) jump to L3

testq %rdi, %rdi

Pretend rdi == 5



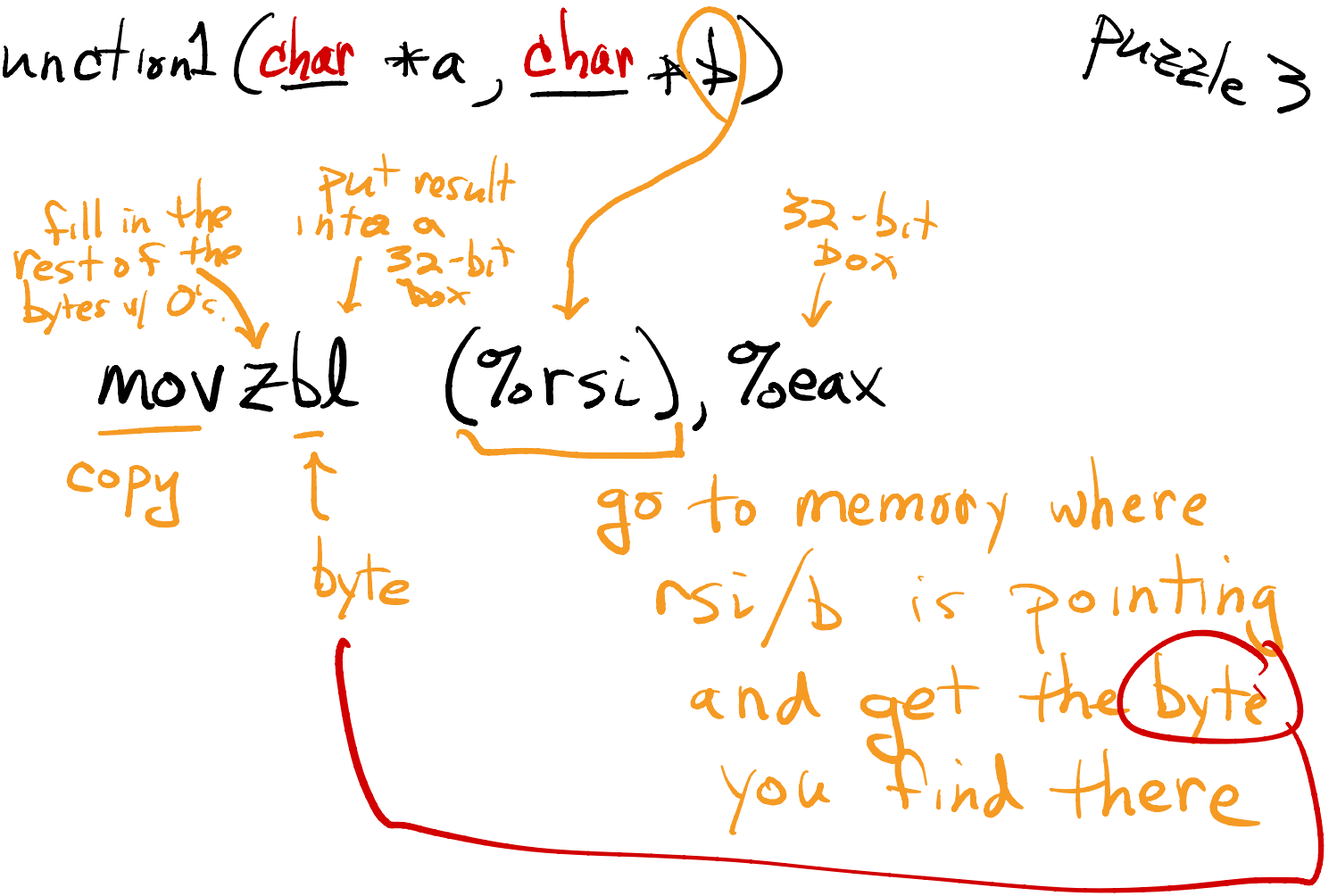
sete %al

if ZF = 0, set al = 0

else set al = 1

function1(char *a, char *b)

Puzzle 3



Puzzle 5

```
leal (%rcx, %rax), %esi
```

① ② ③

%rcx + %rax → 32 bit

64 bit

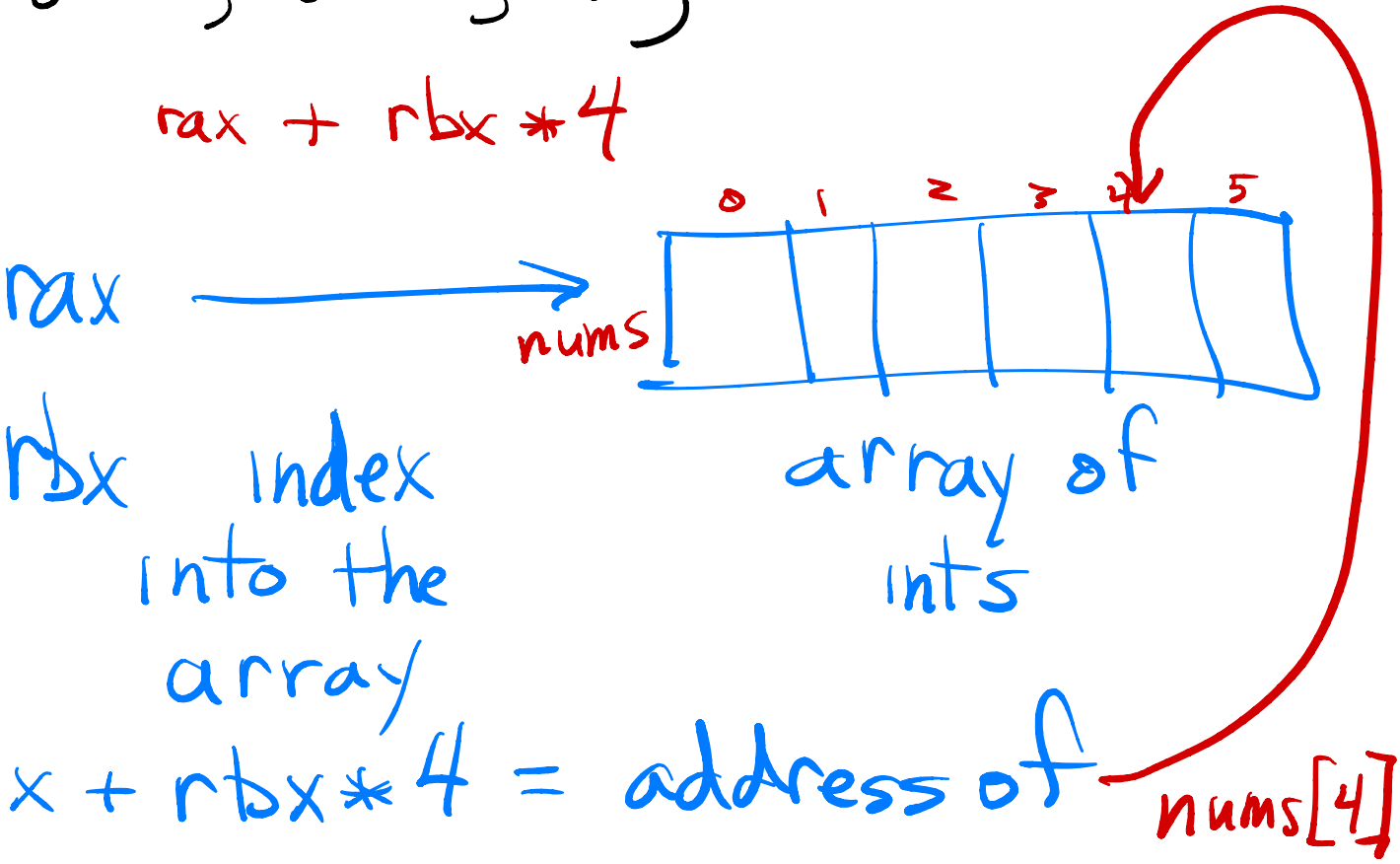
① LEA does computations only on 64-bit #s (thus rcx not ecx)

② Loads bottom 32 bits of sum into destination

LEA "load effective address"

(%rax, %rbx, 4)

rax + rbx * 4



0x8(%rsp, %rbx, 4)

rsp + 8 + 4 * rbx

(rsp + 8) is where the array of ints lives