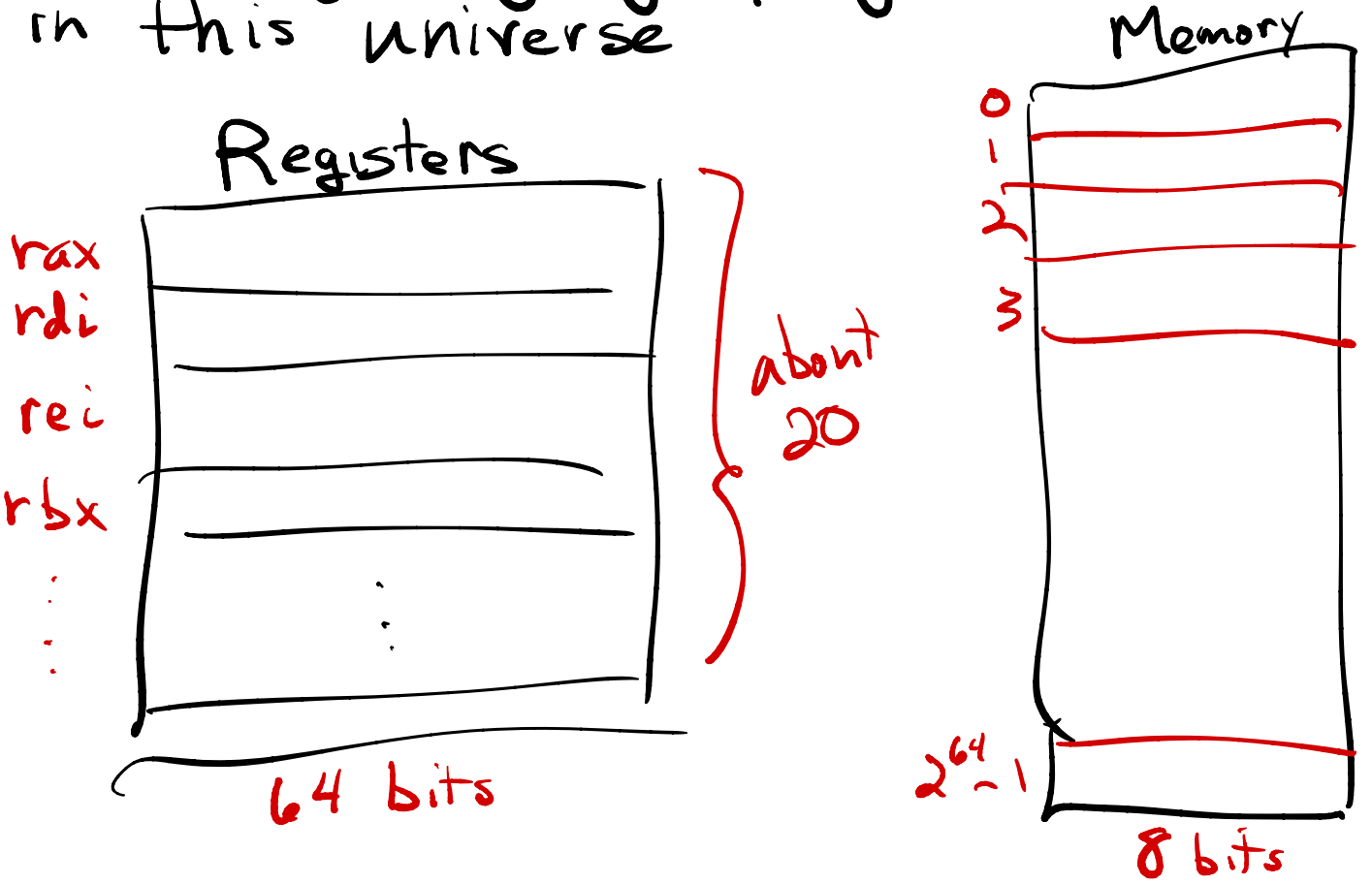
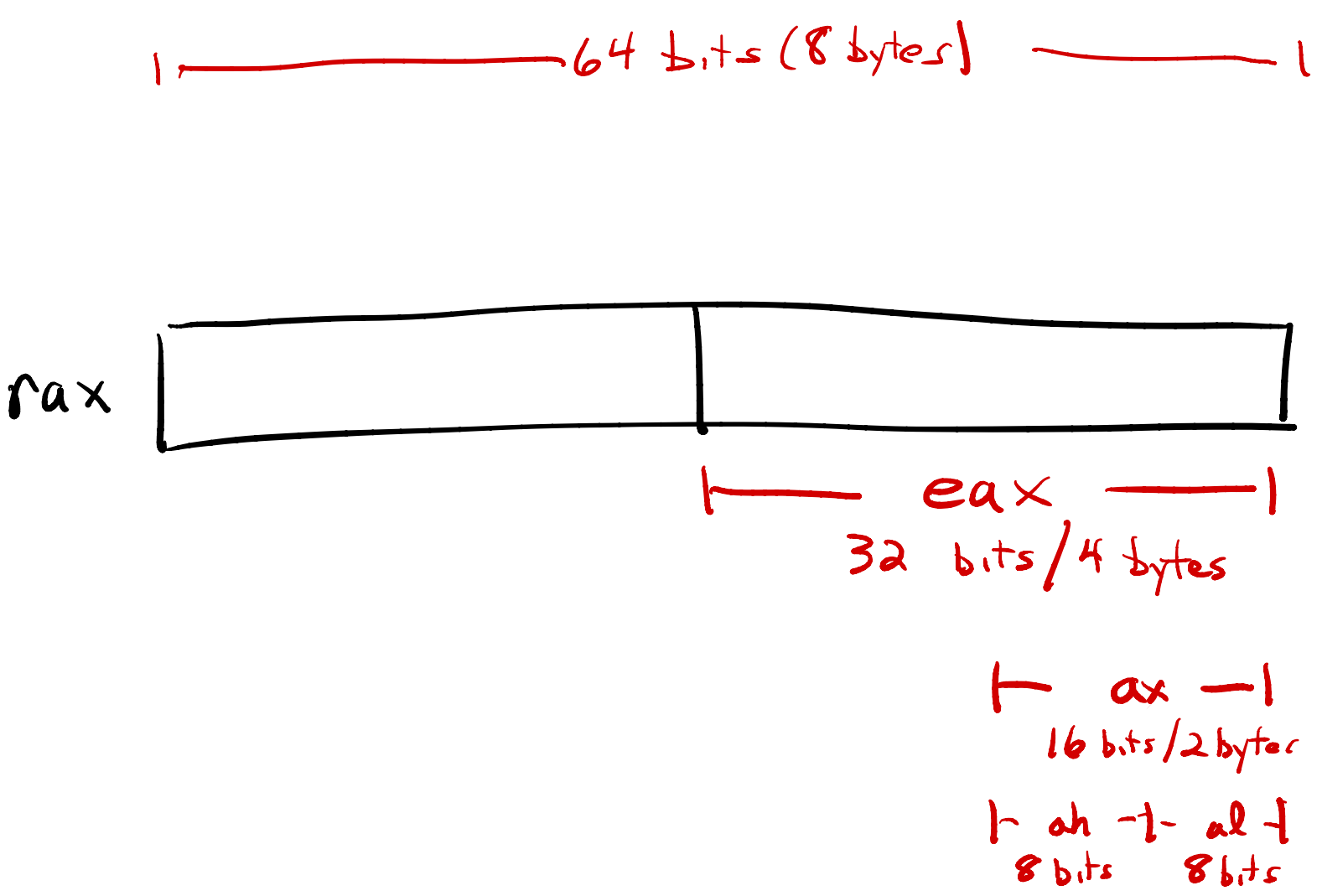
CS 208

M, 13 Oct 2025

Assembly language programmers work in this universe




64 bits (8 bytes)
rax
eax
32 bits/4 bytes
ax
16 bits/2 bytes
ah
al
8 bits
8 bits

movl — copies data from one place to another

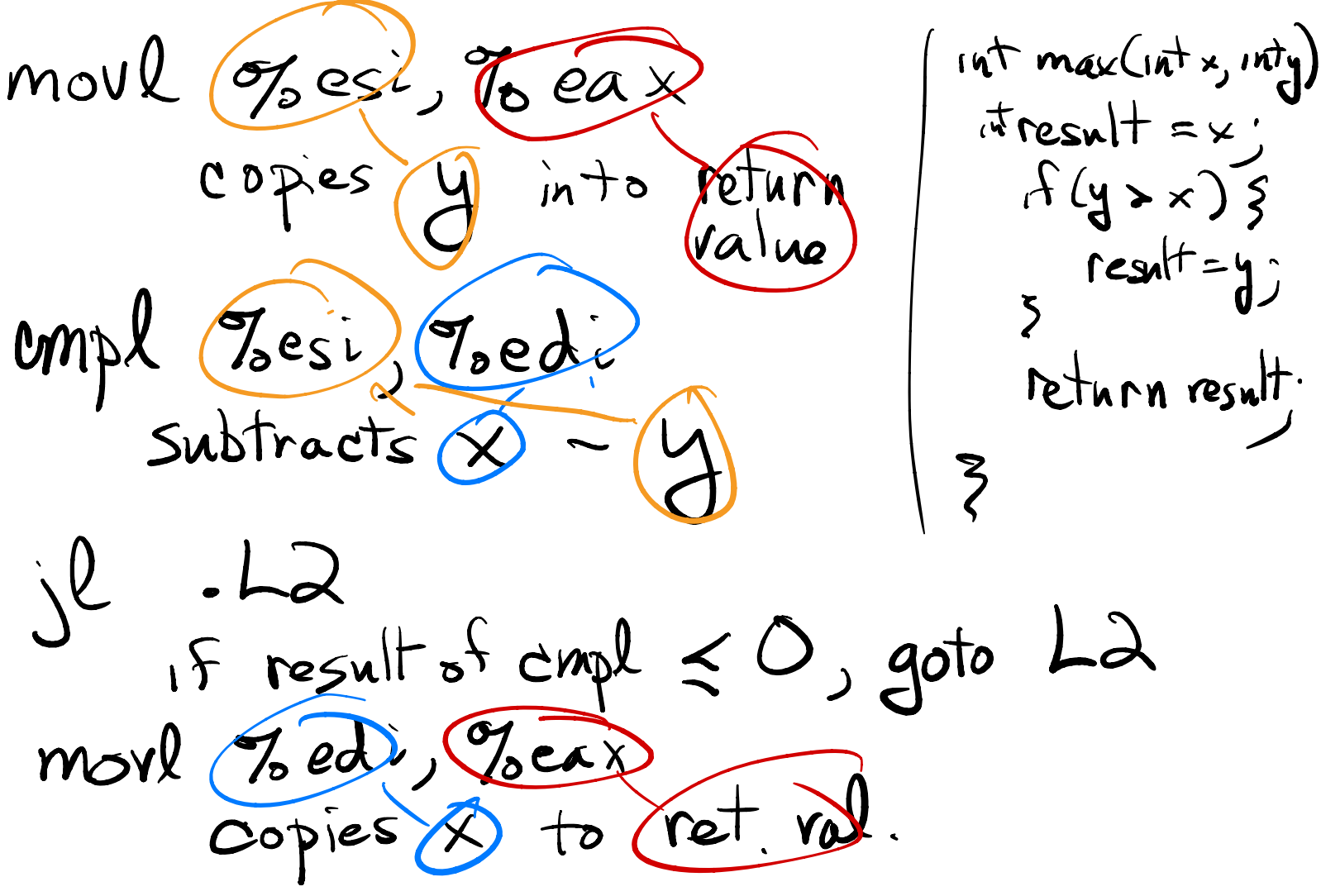
negl — negates an integer in-place

ret — returns from a function (this is complicated)

eax —— conventionally the return value (if return type is int)

edi — 1st parameter of a fn.

esi — 2nd parameter of a fn


movl %esi, %eax
copies y into return value
cmpl %esi, %edi
subtracts x - y
jl .L2
if result of cmpl < 0, goto L2
movl %edi, %eax
copies x to ret. val.
int max(int x, int y)
int result = x;
if (y > x) {
result = y;
}
return result;
}

max(3, 4)

movl %esi, %eax

cmpl %esi, %edi

subtracts 3 − 4

jl .L2

if subtraction was <0 (it was)

goto L2

ret

returns eax to the caller