Virtual Memory

- OS must manage physical memory
  One single physical resource
- all programs want to use same addresses
  - stack 0xff...
  - code 0x80....
  - portability
- What if physical memory is smaller than expected?
  - make memory seem larger than it is (swapping)
- ★ - protection
  - isolates your data
  - protects other's data
- Sharing between processes

1) how VM works

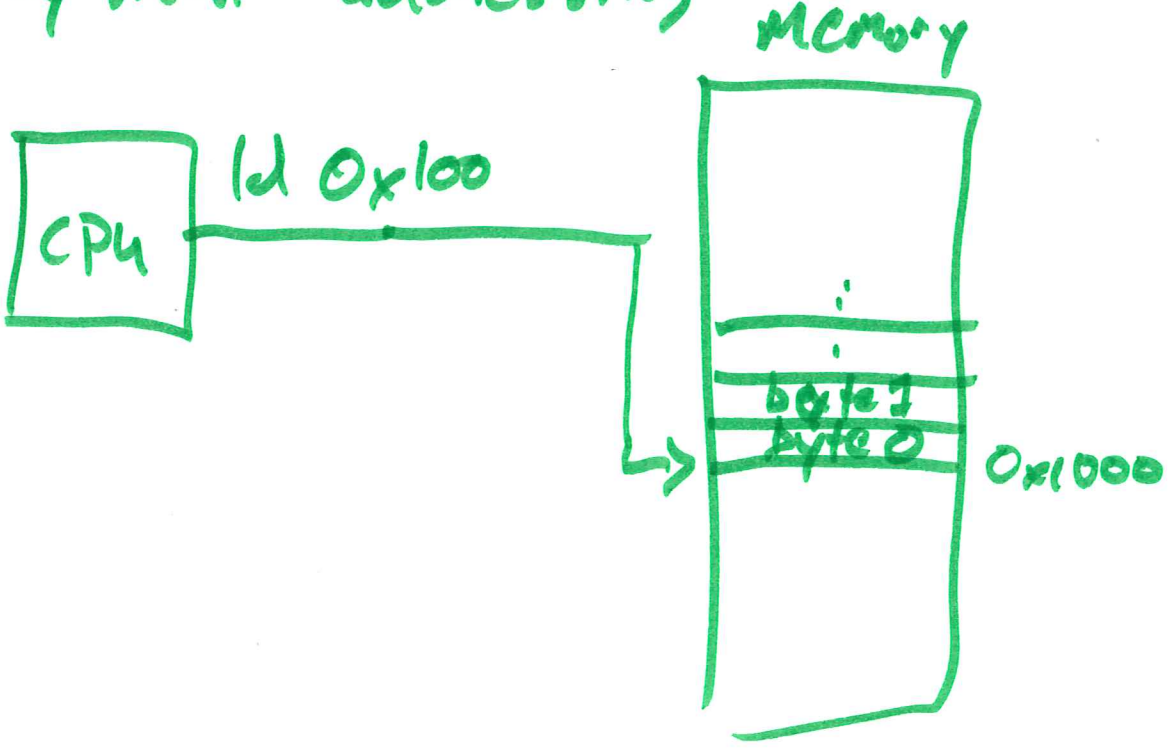
2) how VM is actually implemented

3) speeding up VM (caching)

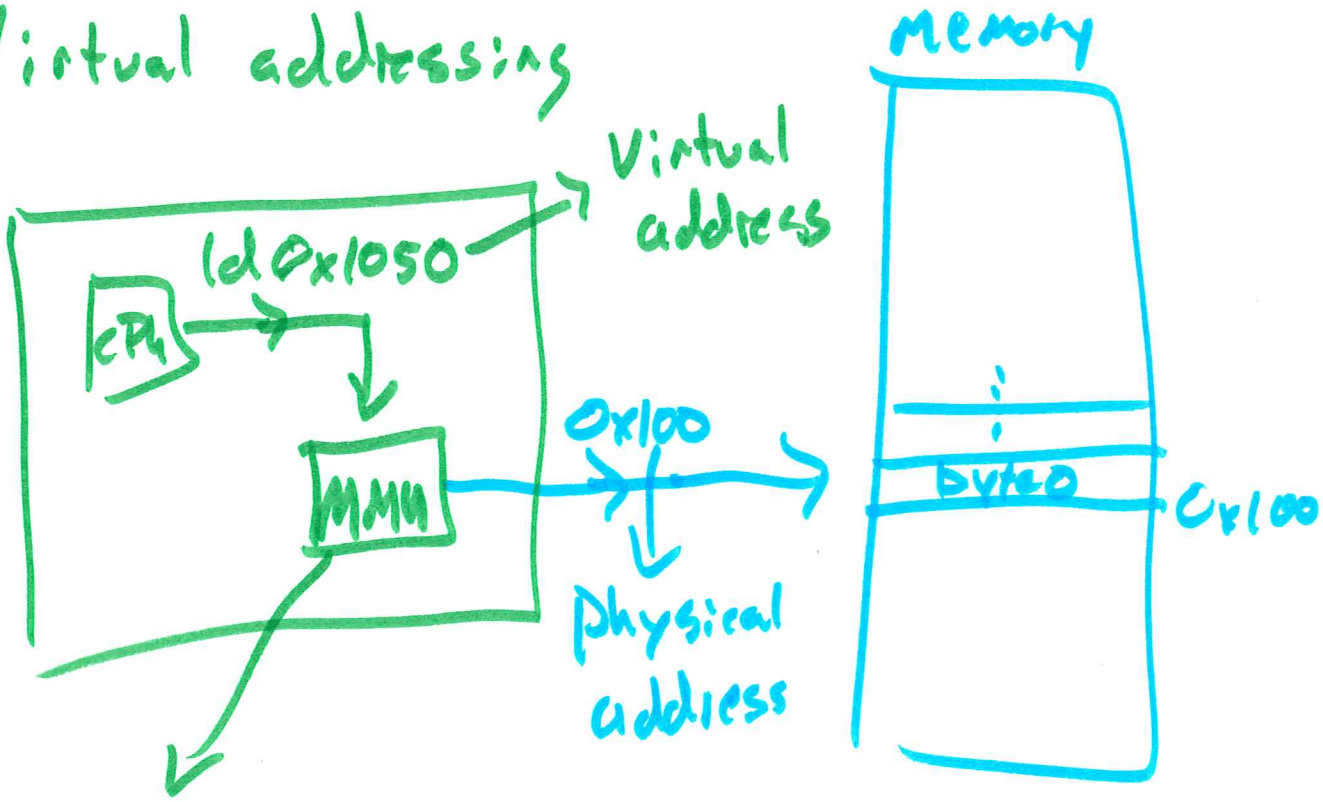
4) memory management user-level (malloc)

5) code + memory (load + link)

Physical addressing



Virtual addressing



MMU - Memory Management Unit

---

address space: Set of addresses (usually linear)

Virtual address space $\{0 \ldots N-1\}$ → 2^n → n → # of bits

Physical address space $\{0 \ldots M-1\}$ → 2^m

64-bits today

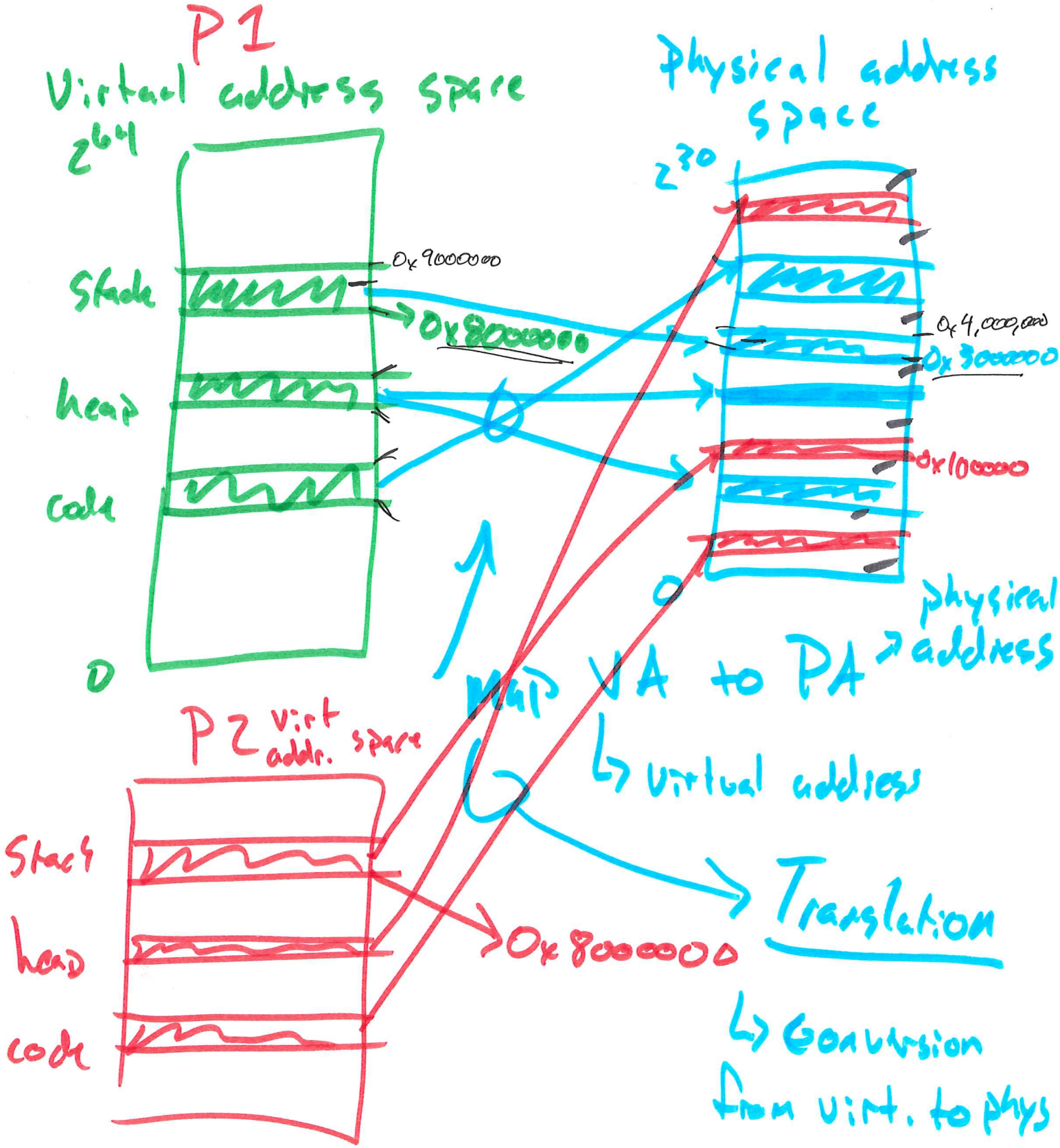
defined by H/W

40-48 bits

Program's address space
→ memory the program thinks it can access
Virtual address space

int a = 12;

&a → what's possible here?



Translation table

	Base	translation	size
P1	0x8000000	→ 0x3000000	1,000,000
P2	0x8000000	→ 0x1000000	1,500,000

↑ segmentation →

base
bounds (size)
translation

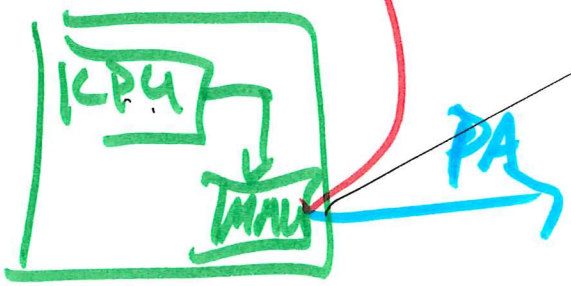
P1

P2

0x1000.. 0x3000

0x800000 → 0x000000

0x1000.. 0x1000..



Segmentation leads to fragmentation

Solution? →