L3: Operating Systems

Sam Madden

madden@csail.mit.edu

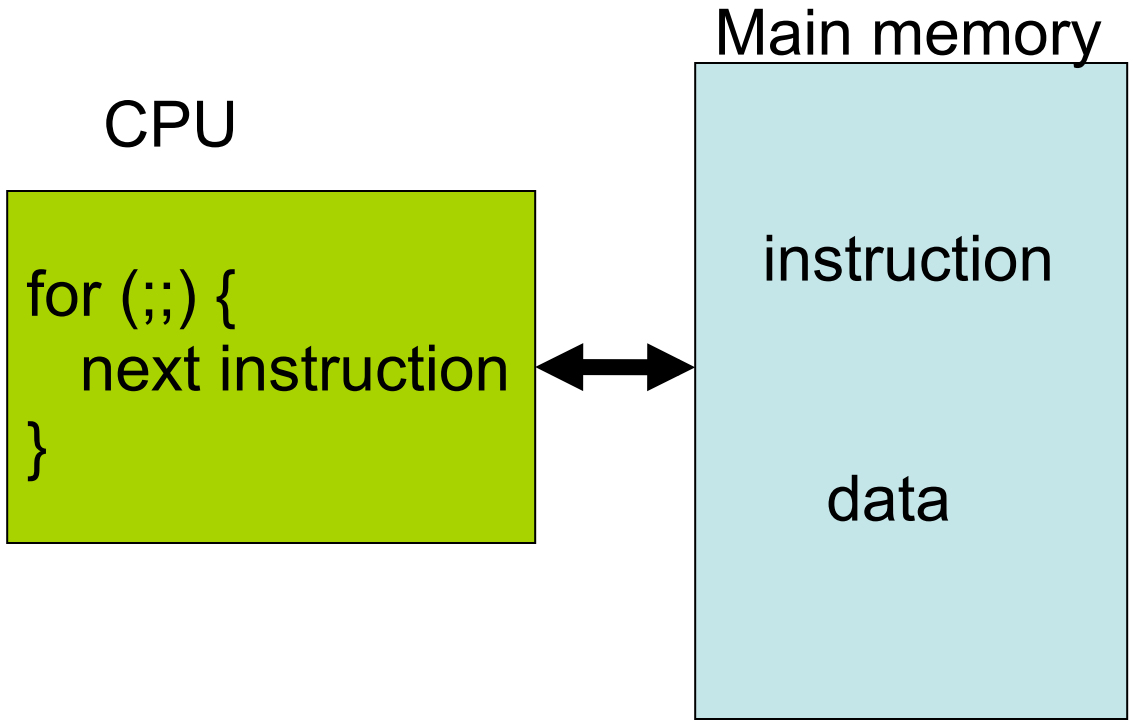
6.033 Spring 2014

OS: New topic [4 lectures]

- Case study of widely-used system
  - Virtual memory system, file system, processes,
- Illustrates ideas from first lectures:
  - OSs supports client/server computing within a single computer
  - OSs have a naming system
- Introduce new ideas and techniques: kernel, files, locks, etc.
- Example: UNIX (Linux, BSDs, etc.)

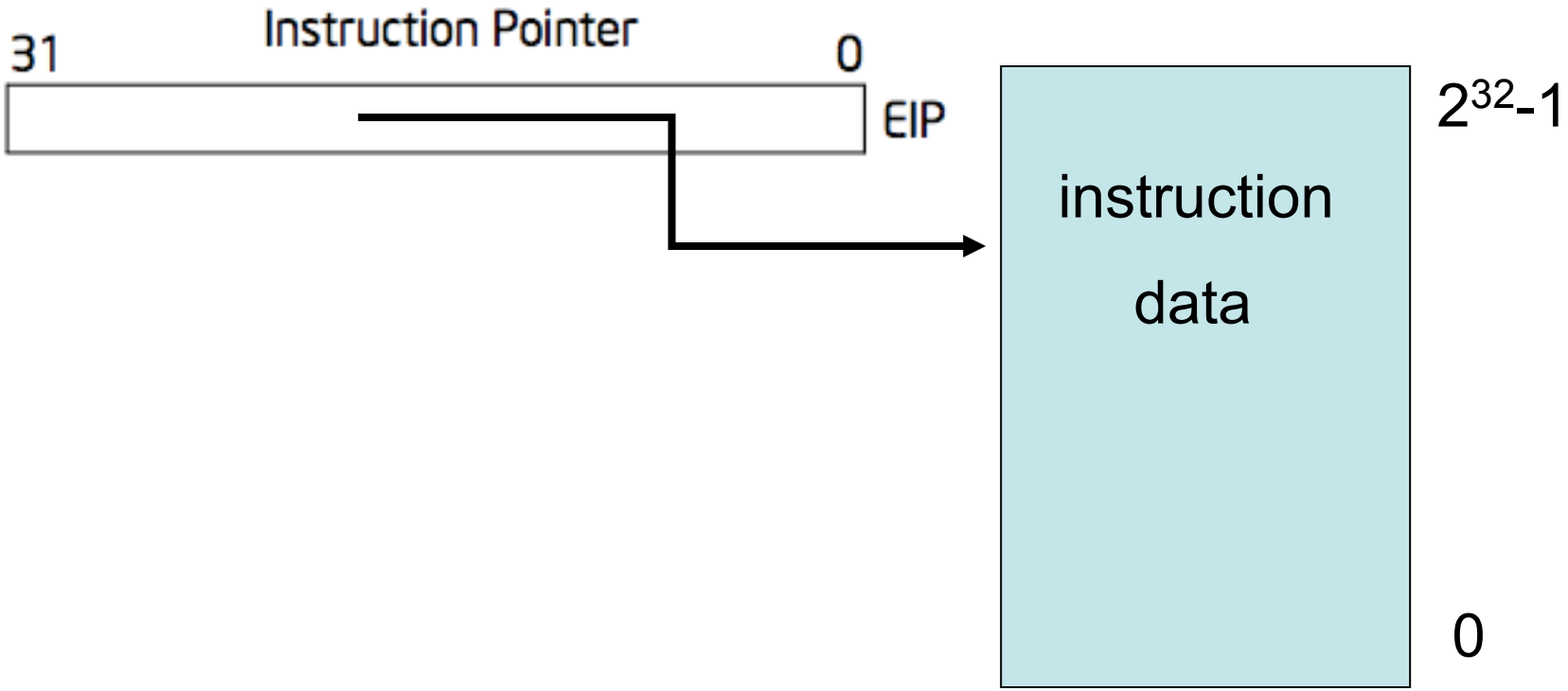
Virtualizing Memory

Running a single program



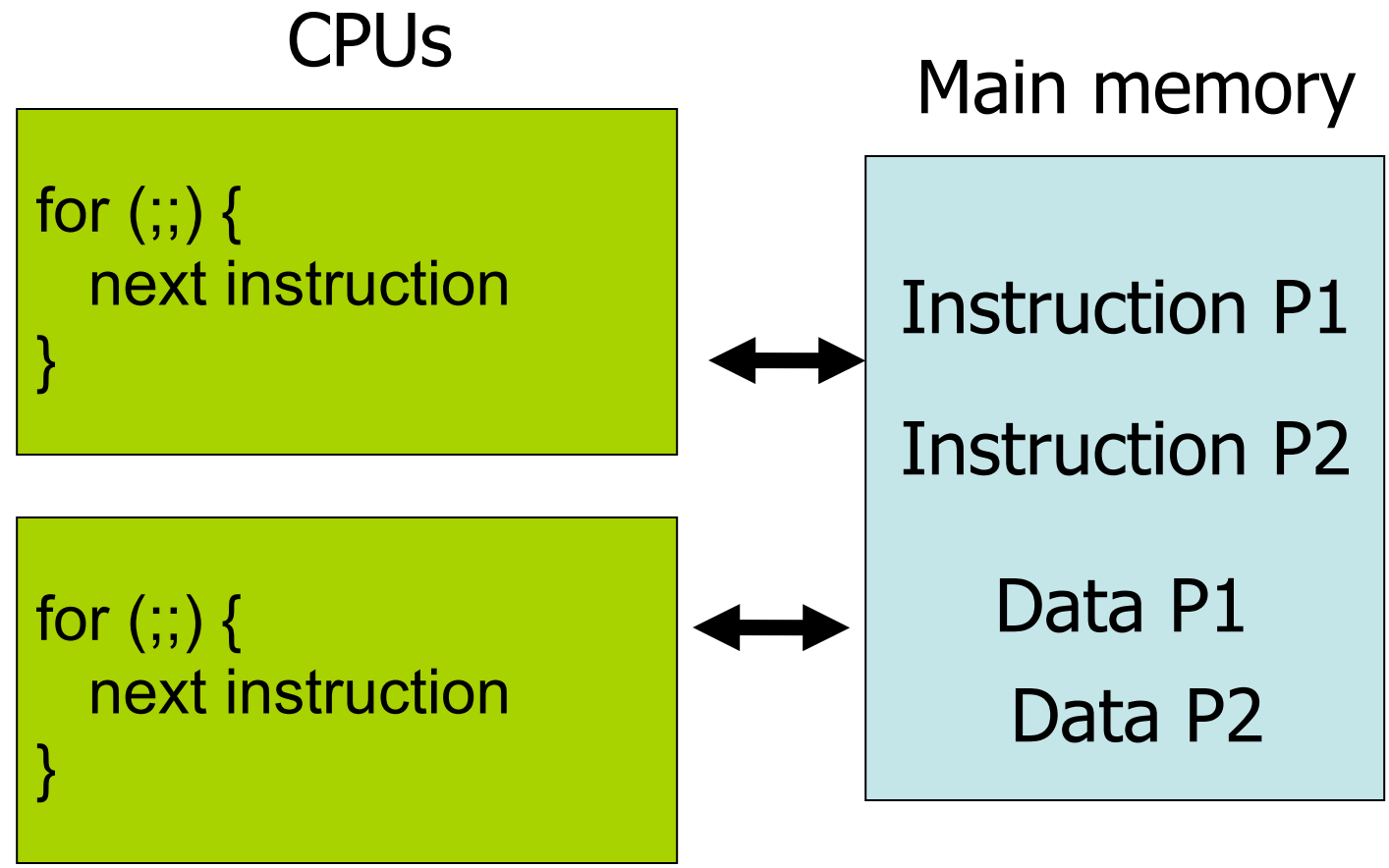
- Memory holds *instructions* and *data*
- CPU *interpreter* of instructions

32-bit x86 implementation



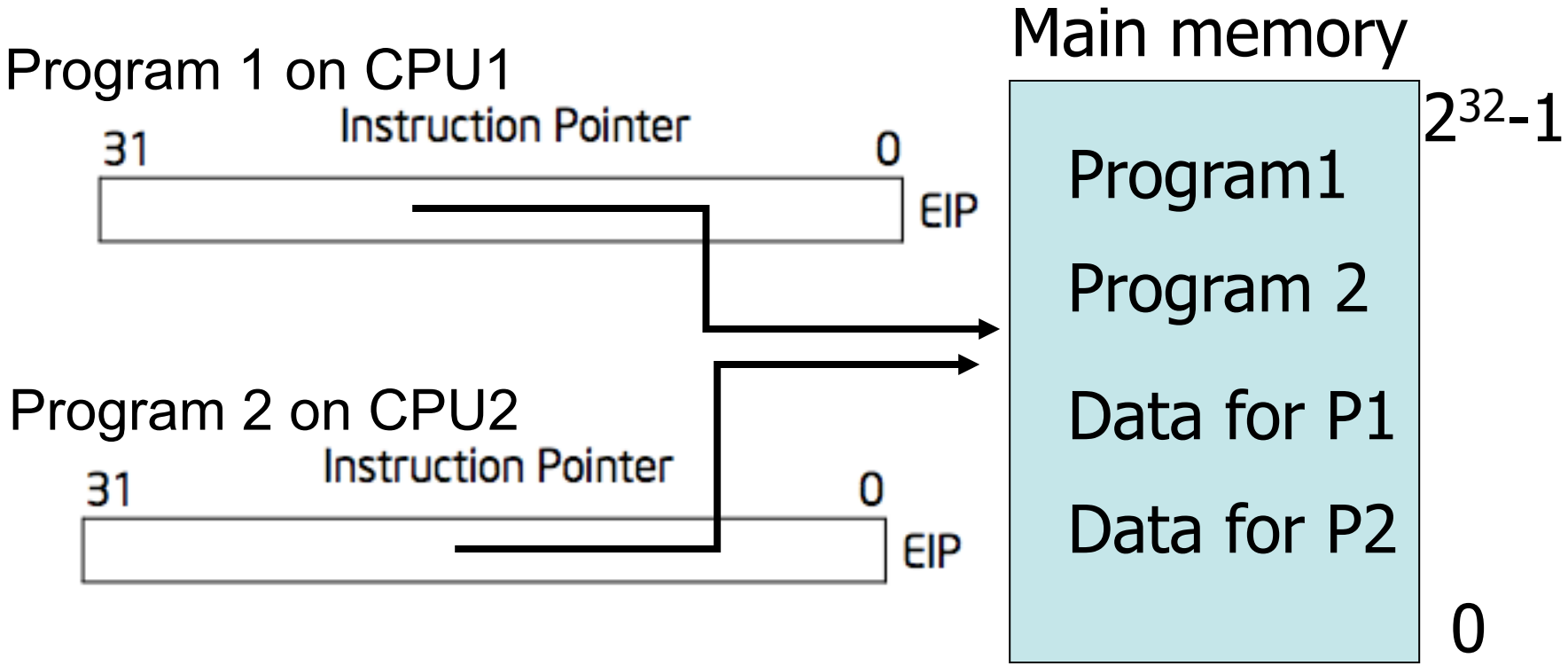
- EIP is incremented after each instruction
- Instructions are different length
- EIP modified by CALL, RET, JMP, and conditional JMP

Several programs



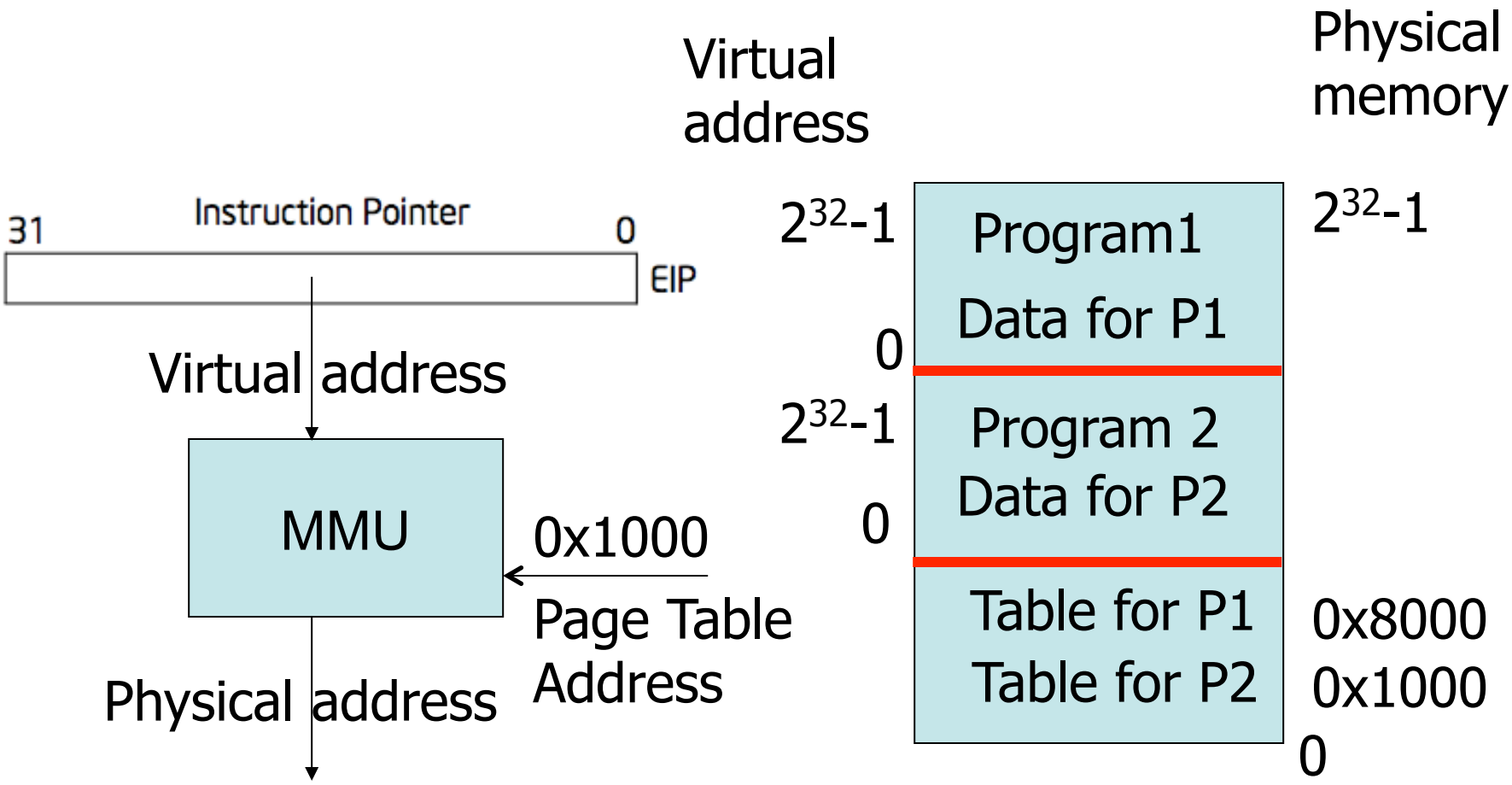
Problem: no boundaries!

Problem: no boundaries



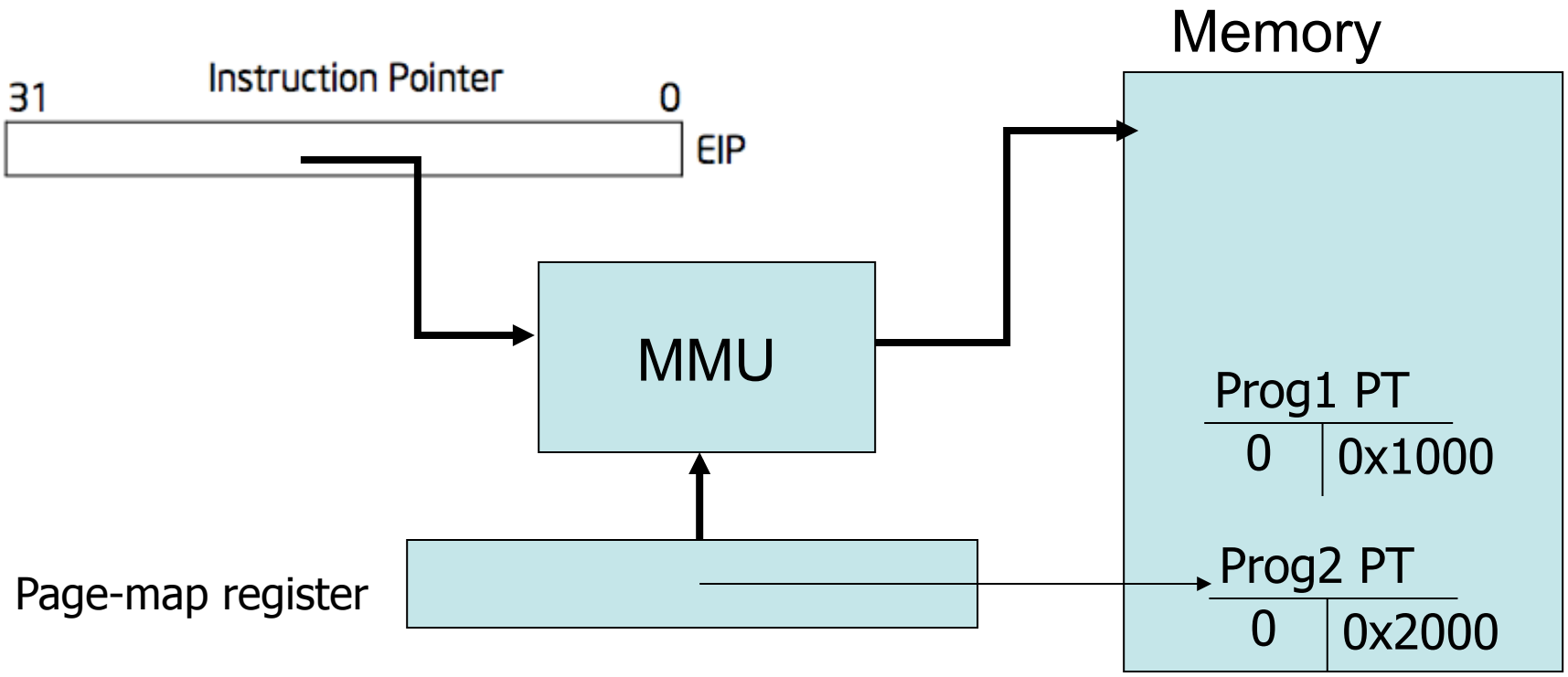
- A program can modify other programs data
- A program jumps into other program's code
- A program may get into an infinite loop

Approach: memory virtualization



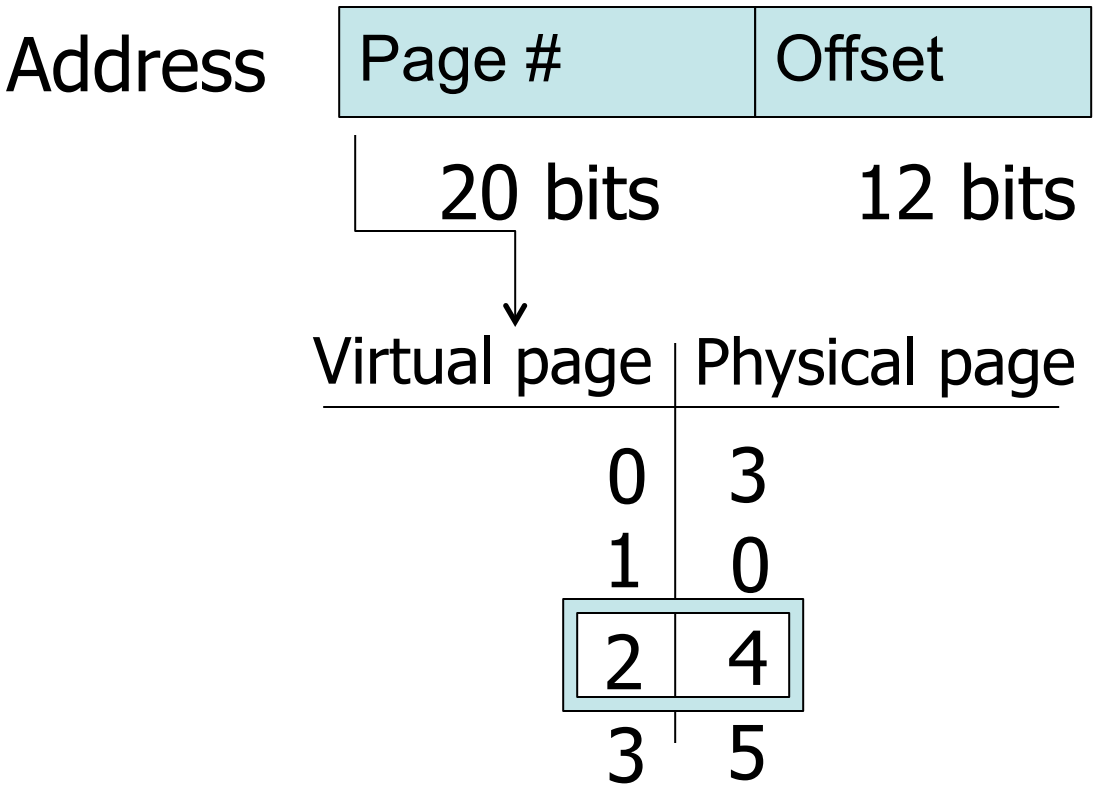
- P1: LD r0, 0x0000 translated with Table 1
- P2: LD r0, 0x0000 translated with Table 2

Page table records mapping



- Each program has its own translation map
  - Physical memory doesn't have to be contiguous
- Prog1: 0x0000 -> 0x1000
- Prog2: 0x0000 -> 0x2000

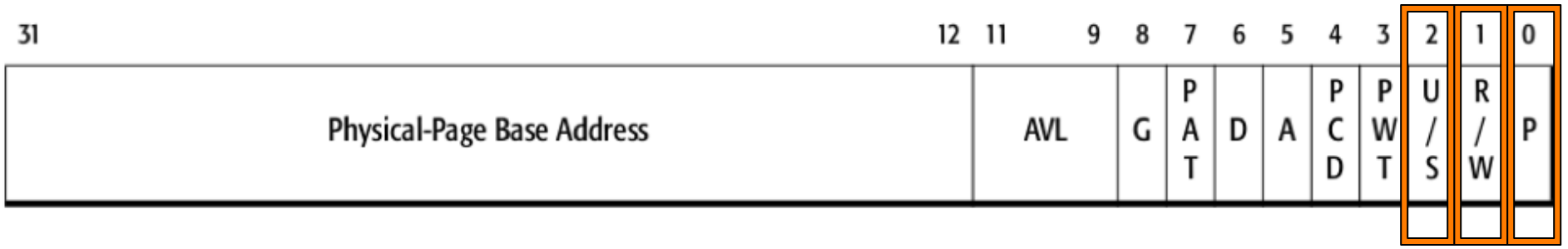
Space-efficient map



20 bits 12 bits

0x00002148 → 4 * 4096 + 0x148 = 0x00004148

x86 page table entry



- R/W: writable?
  - Page fault when W = 0 and writing
- U/S: user mode references allowed?
  - Page fault when U = 0 and user references address
- P: present?
  - Page fault when P = 0

Kernel Manages Page Maps, Interrupts

Handling Interrupts

```
Interrupt(x):
        u/k ← k
        handler = handlers[x]
        call x
        iret //sets u/k to u
```

Page faults are a kind of interrupt

Interrupts can also be called directly

OS Abstractions

File system is an abstraction of raw disk
Apps not allowed to access raw disk!

```
main() {
  int fd, n;
  char buf[512];

  chdir("/usr/kaashoek");

  fd = open("quiz.txt", 0);
  n = read(fd, buf, 512);
  write(1, buf, n);
  close(fd);
}
```

Summary

- Two key OS techniques
  - Virtualization allows programs to share hardware
  - Abstractions provide portability, cooperation
    - See Unix paper
- OS kernel enforces modularity
  - Program vs program
  - Program vs kernel