Lecture 22: Virtual memory 1

Friday, March 1, 2019 10:56 AM

Outline

- Virtualization
- Virtual memory
    - Translation mechanisms
    - Paging
    - Page tables
- RISC-V virtual memory

AN x64 PROCESSOR IS SCREAMING ALONG AT BILLIONS OF CYCLES PER SECOND TO RUN THE XNU KERNEL, WHICH IS FRANTICALLY WORKING THROUGH ALL THE POSIX-SPECIFIED ABSTRACTION TO CREATE THE DARWIN SYSTEM UNDERLYING OS X, WHICH IN TURN IS STRAINING ITSELF TO RUN FIREFOX AND ITS GECKO RENDERER, WHICH CREATES A FLASH OBJECT WHICH RENDERS DOZENS OF VIDEO FRAMES EVERY SECOND
BECAUSE I WANTED TO SEE A CAT JUMP INTO A BOX AND FALL OVER.
I AM A GOD.

RISC-V
"machine" mode

Virtualization

Downside of machine mode?

need A way to limit user code

Don't want programmer to do anything dangerous

Security w/ multiple apps

↳ isolation

↳ protection

It's complicated to manage memory

Want multiple users/applications running at "the same time"

↳ time sharing

Need to Virtualize

Seem like each application is running on its own CPU w/ its own memory

How to virtualize the CPU?

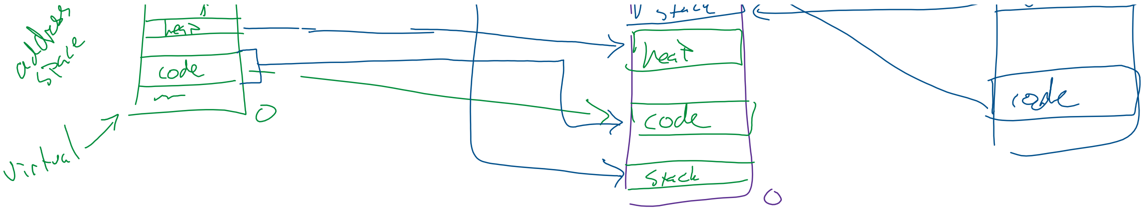
→ what state do you need?

- Current state of all registers → architectural registers X1 .. X31
- PC ↳ no need to save μarch state like pipeline regs
- other machine regs
- Save memory state

Virtualize memory?

1) isolation between procs
2) protection between procs
3) Seem like we're running alone
4) Appear like we have lots of memory → rv32 ↳ 4GB mem per process

Add a level of indirection!



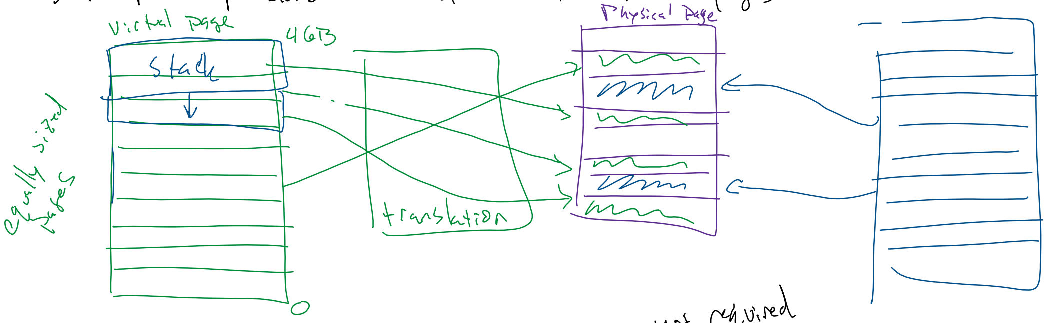
Direct mapping or segmentation

↳ sizes are mismatched → memory wasted → fragmentation

↳ each process map to same spot → solve w/ segmentation

↳ Need to be aware of other procs (and yourself) when resizing segments

To solve these problems → break down into smaller pages



not required

Virtual address size (eg 32 bit in rv32) ≠ Physical address size

How to represent translation in software?

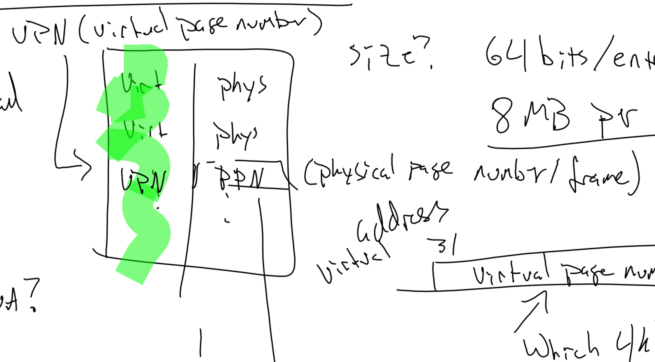
4GB VA
4kB pages

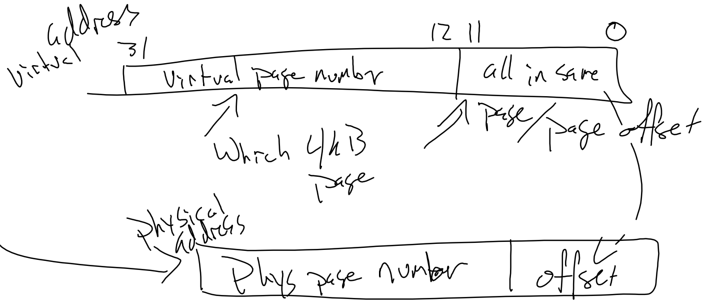
Table
↳ Big overhead

size? 64 bits/entry * apps * 1 million (# of pages)
8 MB per application

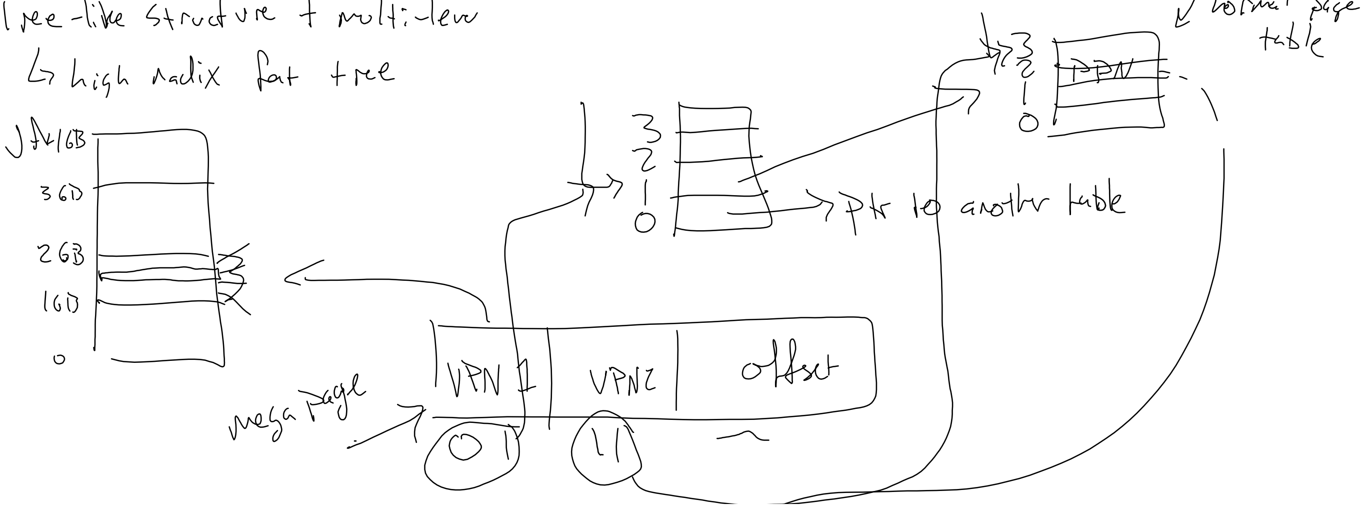
hash table
inverted page table

↳ What if 64 bit VA?
↳ 16 PB





Tree-like structure + multi-level
↳ high radix fat tree



PPN	offset

physical address

Sv32

↳ 2 levels

1024 4 MB megapages

1024 4 kB base pages/megapage