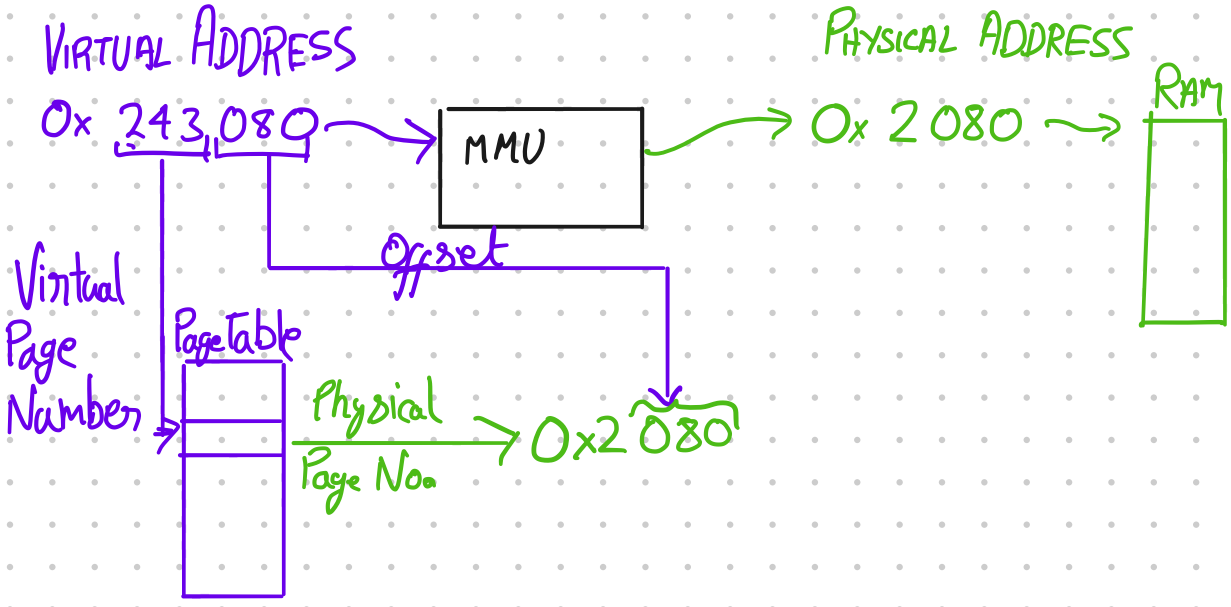
CS202 (-002) Lecture 12: Virtual Memory II

Announcement

- Next week: Midterm week
  - ↳ Tuesday 10/21: Review
    - ↳ Bring questions, etc.
  - Thursday 10/23: Midterm
    - ↳ Closed book, in-class
      - 1 double sided cheat sheet

Virtual Memory

↳ Where we were



Assumption: $4KB = 4 \cdot 2^{10} B$ pages

↳ Offset must be 12 bits

Concern: Pagetable size

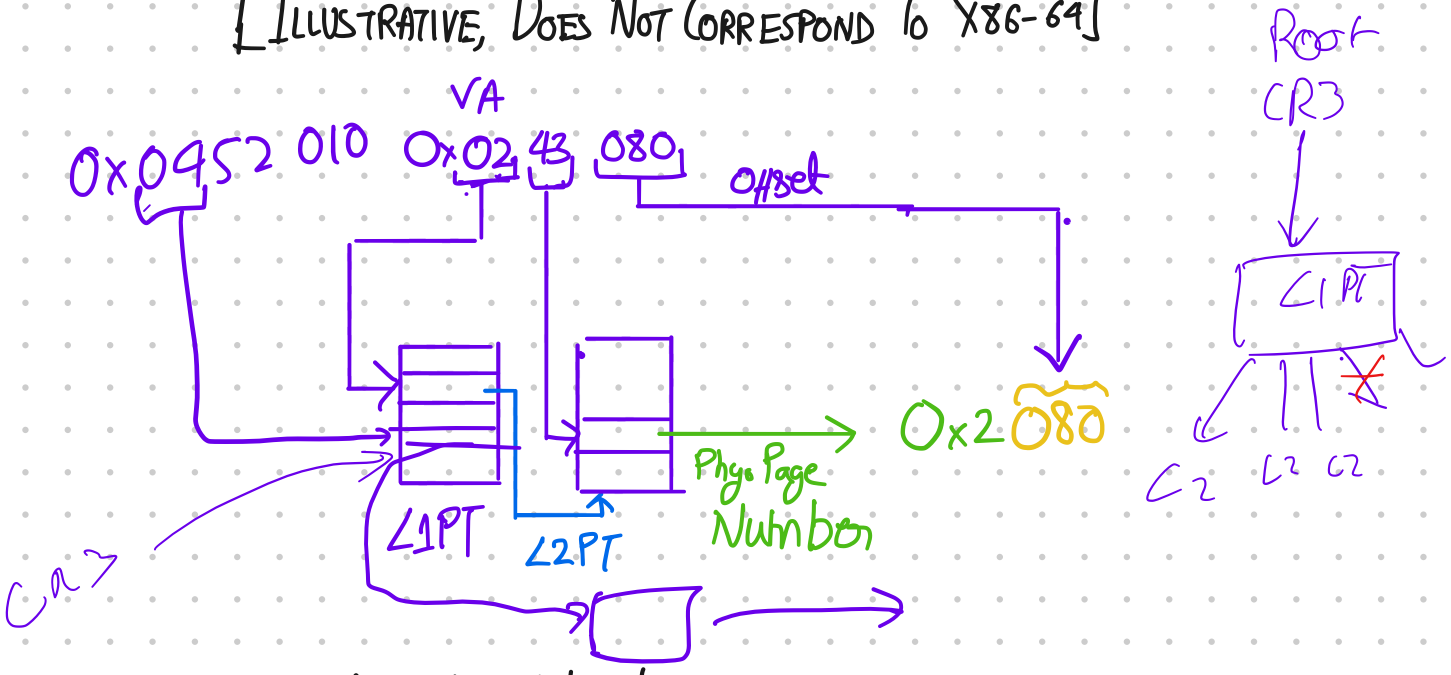
↳ 48-bit virtual address

⇒ 2^{36} pages per process

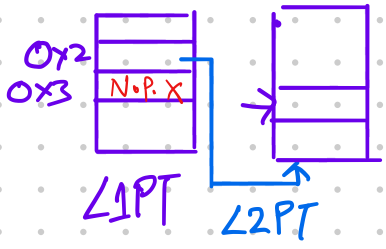
↳ Most unmapped

Answer: make a tree

[ILLUSTRATIVE, DOES NOT CORRESPOND TO X86-64]



Why does this take less space?

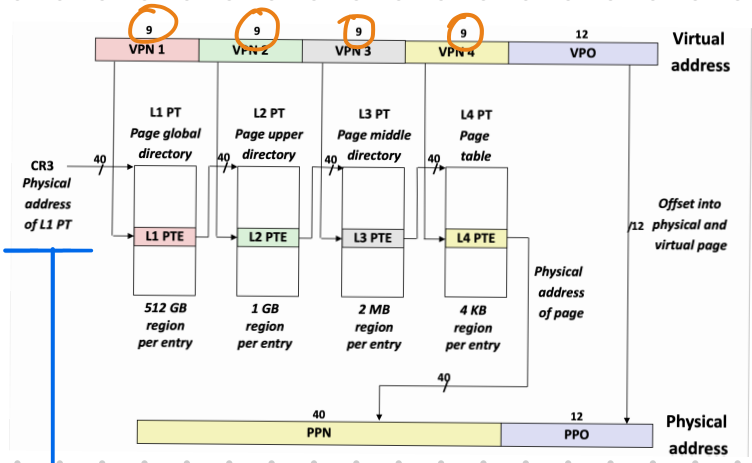


- Can mark that an entire subtree is not present

Q: In example, what happens if program accesses 0x0343 080?

Q. What type of addresses (VIRTUAL or PHYSICAL) does the L1 page table contain?

x86-64

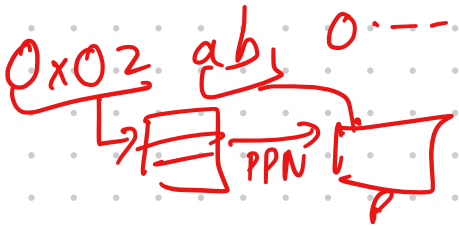


→ We talked about this last class

- Register
- Can only be accessed from kernel/supervisor mode
- PA of L1 Page Table

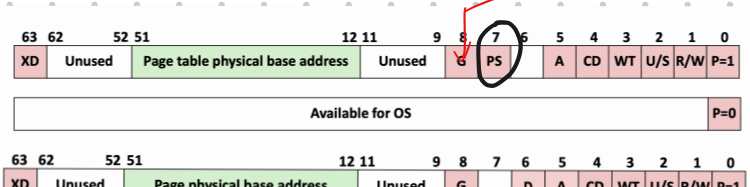
Differences from our example

- Tree has larger height (Can save more space)

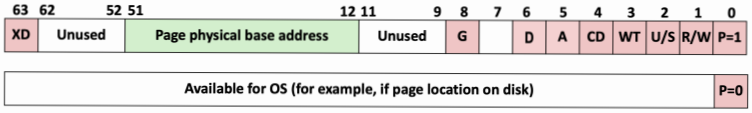


Contents

Don't worry (has to do with caching)



L1-L3



L4

Each entry references a 4K child page. Significant fields:

P: Child page is present in memory (1) or not (0)

R/W: Read-only or read-write access permission for this page } PERMISSIONS

U/S: User or supervisor mode access

WT: Write-through or write-back cache policy for this page

A: Reference bit (set by MMU on reads and writes, cleared by software)

D: Dirty bit (set by MMU on writes, cleared by software)

Page physical base address: 40 most significant bits of physical page address (forces pages to be 4KB aligned)

XD: Disable or enable instruction fetches from this page.

Don't worry

Note: 64bits = 8bytes in size

In 4K: $2^2 \cdot 2^{10}B / 2^3B$

$= 2^9$ entries

∴ 9-bit offset

P_0 Present (1) or Not

R/W : Read-write (1) or read-only

U/S : Access from user space (1) or not

A, D ← set when a page is accessed

A ⇒ page was read or written

D ⇒ Page was modified

} VM IV

Complexity of Address translation

Remember (Lec 08) memory access can be up to 100ns

[OR ~100-400 cycles]

↳ # of memory accesses when going from VA to PA?

Need to reduce memory translation overheads

↳ CACHE entries

↳ VA → PA mapping

→ Permissions

→ ...

} TRANSLATION LOOKASIDE BUFFER

Fixed size

Ice Lake 64 items L1 } For 4K pages.

2048 items L2

PROBLEM

↳ Keeping cached entries CONSISTENT with page table contents

↳ Need to flush TLB when page table changes

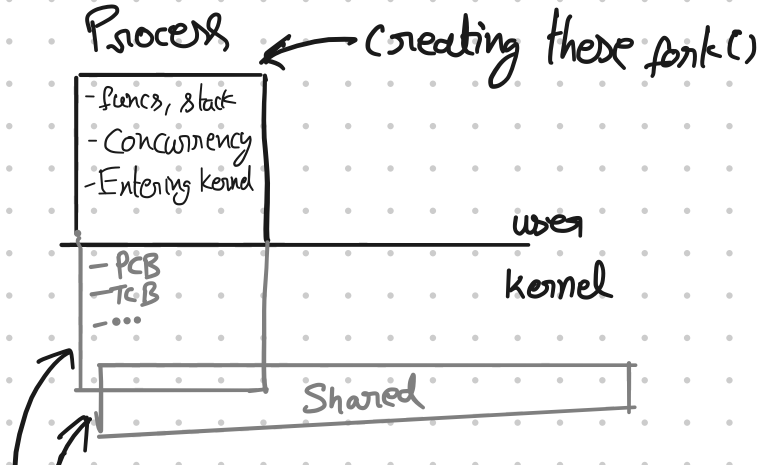
Can be expensive if

Core A changes Page table for process P

while core B is executing thread t

for process P

↳ TLB shootdown ☹ Not covered in this class.



Set U/S bit to limit access!