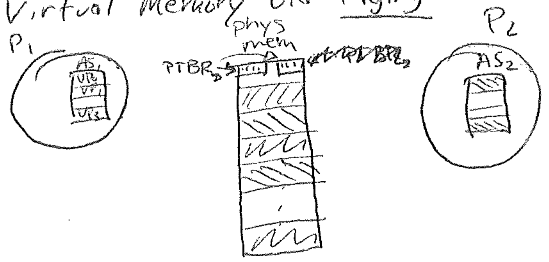
LECTURE 6:
TLBs

VM:

Last Time:

⇒ Virtual Memory via Paging



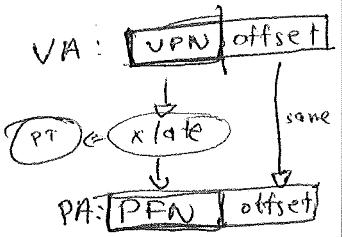
PageTable: per process
VP → PF mapping info

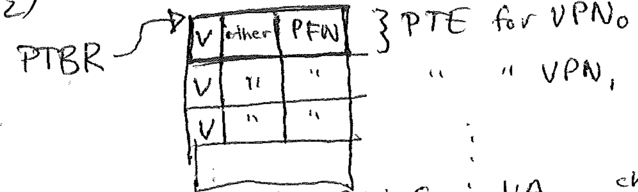
⇒ when P1 running:

PTBR: address of P1's page table

⇒ upon context switch (to P2)
OS sets PTBR to P2's Page Table

⇒ upon each mem reference (inst fetch, explicit load/store)
H/W accesses page table to complete addr. translation





⇒ why good?
→ supports flexible AS (valid bit)
⇒ (doesn't waste memory w/ unused AS)
→ easy to manage free space (simple free list) [fixed size]
(no ext. frag)

H/W
→ extract VPN from VA — cheap
→ calculate addr of PTE — cheap
→ fetch: PTE — EXPENSIVE
⇒ extract PFN, form PA: — cheap
⇒ fetch PA ⇒ register — EXPENSIVE

⇒ but, some problems
1) Page tables: too big
2) Paging: too slow
for each mem refs must first access PT!
(another mem ref)
[today's focus]

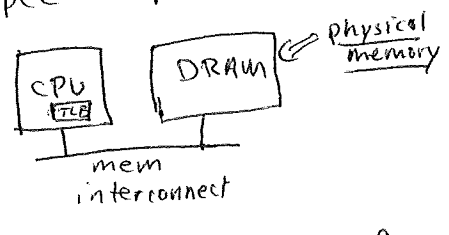
e.g. 32-bit VA space w/ 1K, 2K, 4K pages
how big is PT?

why not just use really big pages?

e.g. load VA, R1
how many mem refs?

⇒ To speed up: Translation Lookaside Buffer (TLB)

Better name: Address Translation Cache



cache: smaller, faster memory
in this case, on chip

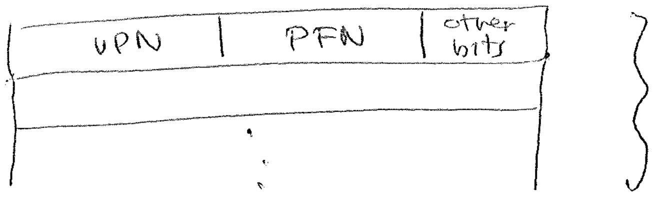
TLB: keep some "popular" translations in TLB ⇒ speed up translation

some*: 32, 64, 128? 256?

h/w:
upon each mem reference:
⇒ extract VPN from VA — cheap
⇒ check TLB: is this VPN in TLB? — cheap
⇒ if yes [TLB hit]
~~get~~ extract PFN from TLBEntry — cheap
form PA — cheap
fetch PA (EXPENSIVE)

⇒ if no [TLB miss]
do same as before:
get PTE from PT (using PTBR)
update TLB w/ translation info
retry instruction

TLB contents:



32, 64, ..., 512 entries

searched: in parallel

usually fully-assoc
(entry can be anywhere)

other: valid does this entry have valid info: [what if no such?]
prot. for quick prot check, such as R, W

what about context switch?
what should OS do? (flush, or ASID)

Issue: H/W has to know so much about (PT structure)
innovation: S/W (or OS) managed TLB
idea: decouple H/W from PT structure

h/w: upon each mem reference
- ⇒ extract VPN from VA
- ⇒ do check TLB
- ⇒ if hit:
  VPN → PFN
  fetch PA
- ⇒ if miss:
  raise [TLB miss exception]
  (trap to OS)

OS: run TLB miss trap handler
check PT for PTE
if valid, etc:
extract PFN
update TLB w/ special instruction(s)
return from trap
(retry current instruction)

privileged ?! ⇒

why bother?
- ⇒ h/w is simpler
- ⇒ s/w has more flexibility

Summary:
Addr Trans w/ Paging: too slow
H/W support to fix: TLB