

Lecture 7

CS 695

A1 - done.

A2 - out.

19th Feb. 2024

Monday

7th Feb.

Wed.

Quiz 1

830 am

using the KVM
interface

to setup &
control

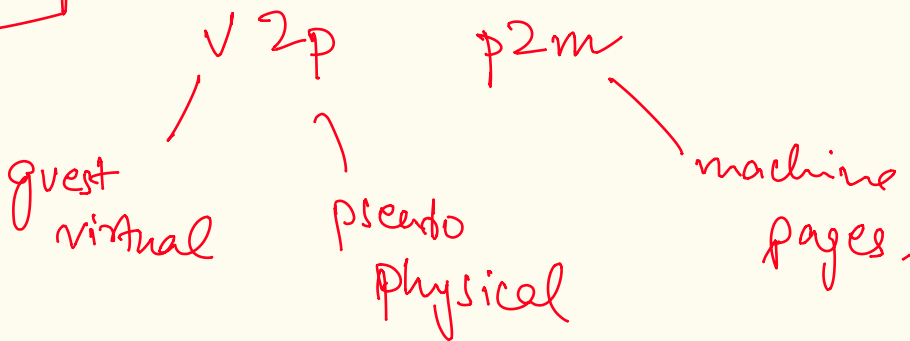
guest VMS.

② WSL — ?

③ n/w accessible
Linux mls.

④ memory virt. w/ VMS.

relap.

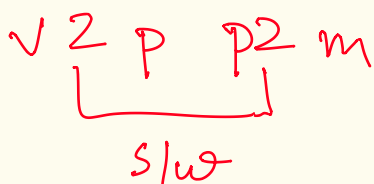


Constraint

① single walk
mmu.

① direct mapping (pv approach)

— all page table updates are via hypercalls.



hypercall (V2P)

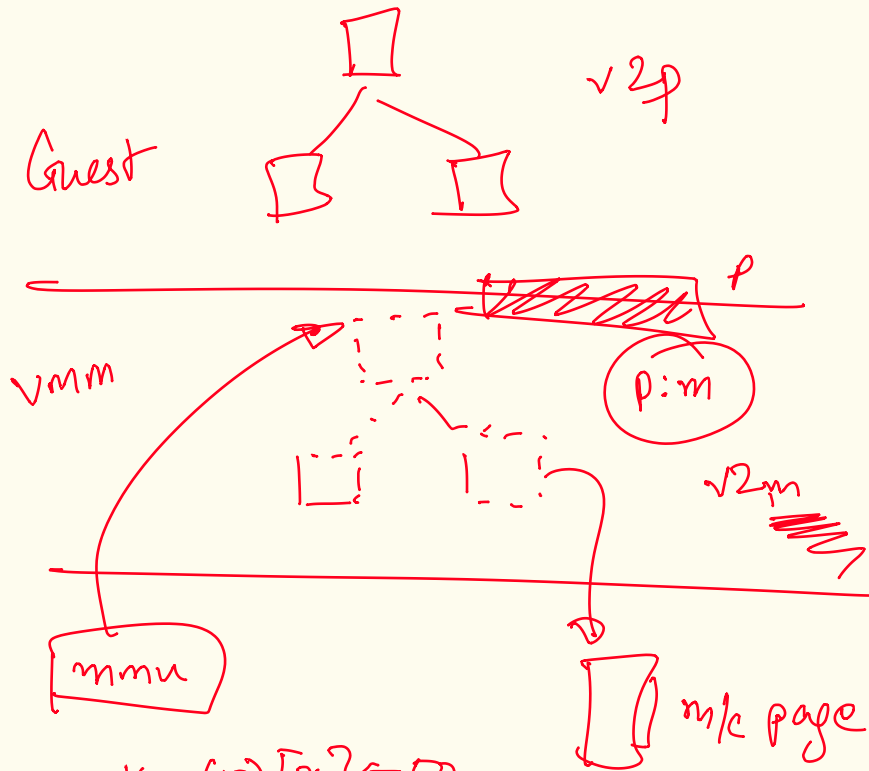
p:m

request

update: V2M

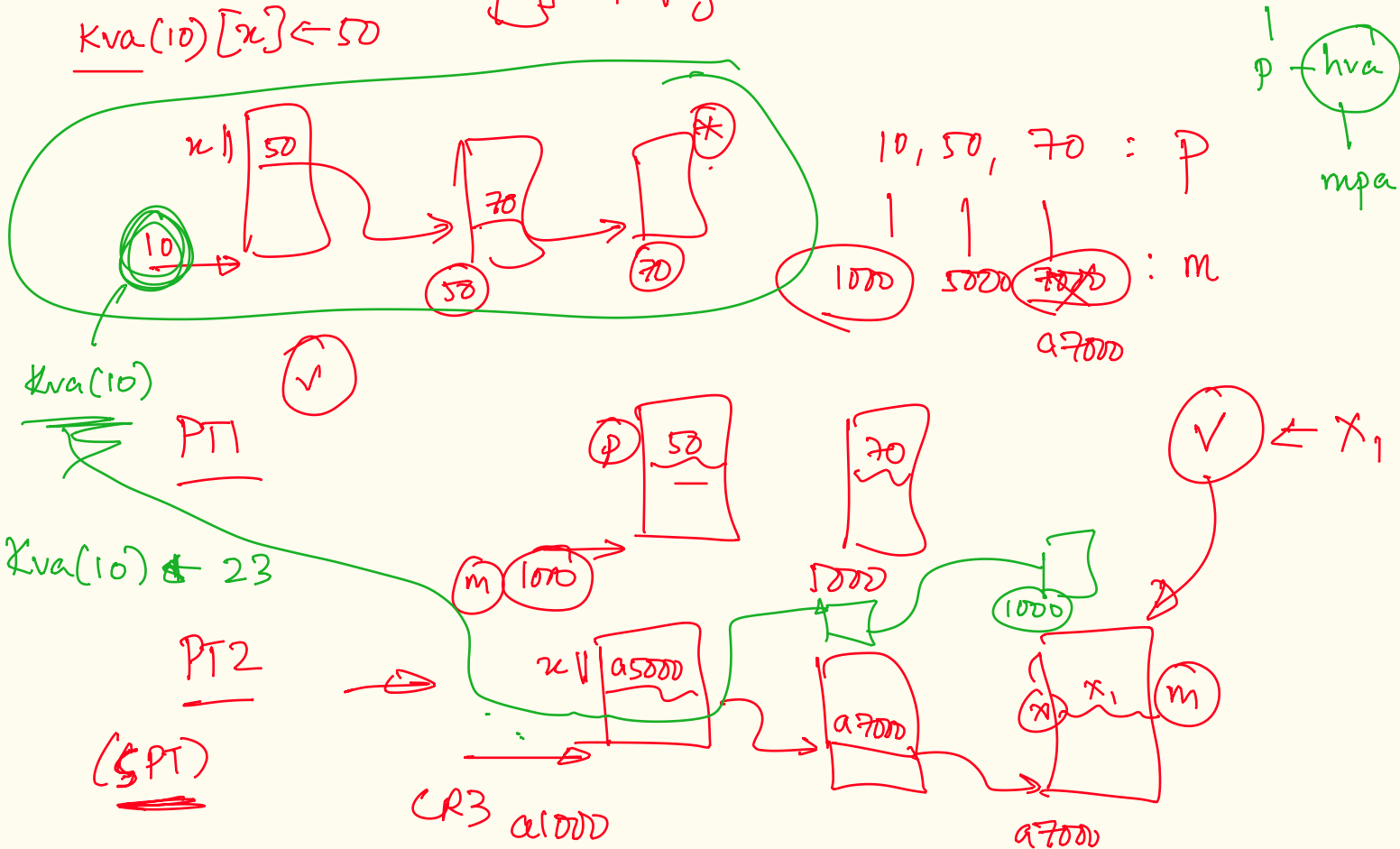
② shadow paging (full virtualized VMs).

- guest unaware of the VMM



--- Shadow page table (SPT)

guest is unaware of the SPT



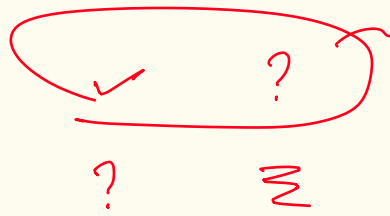
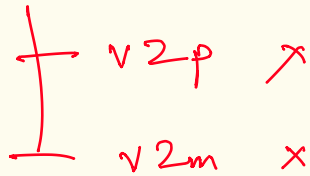
- Key: is to ride on trap on CR3 update.
- mark all guest page table pages as read-only.

SPT: per-process per guest (v2m)

Cost:

- maintain
- ① 2 page tables $\leftarrow$ update cost
 $\leftarrow$ memory cost

- ② page fault handling is costly.



walk the guest page table

& v2m page table

to decide how to handle fault.

- ① - guest fault no v2p

$\Rightarrow$ inject fault & ~~let~~ let guest page handler set up v2p mapping.

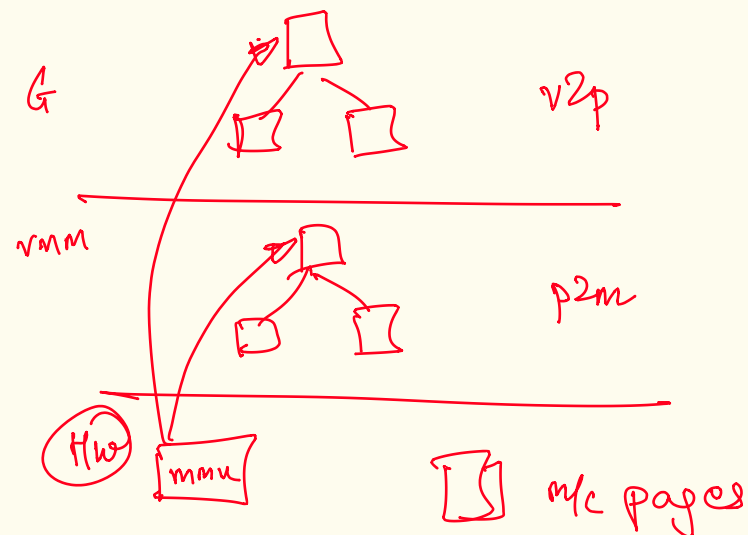
$\Rightarrow$ update to GTS will result in trap & update SBT.

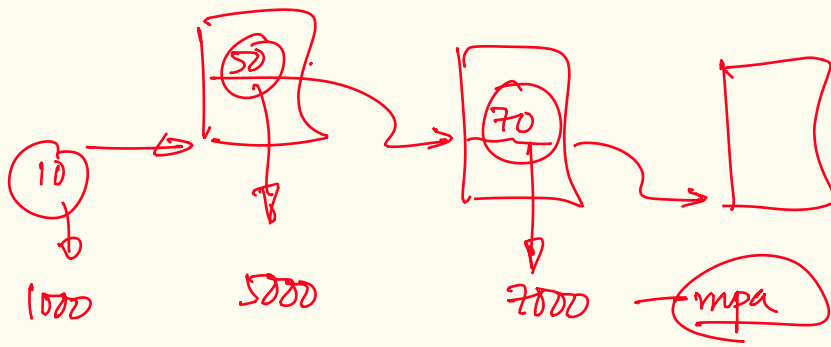
- ② vmm's fault

- swap-in page via vmm pg. fault handler.

- update v2m

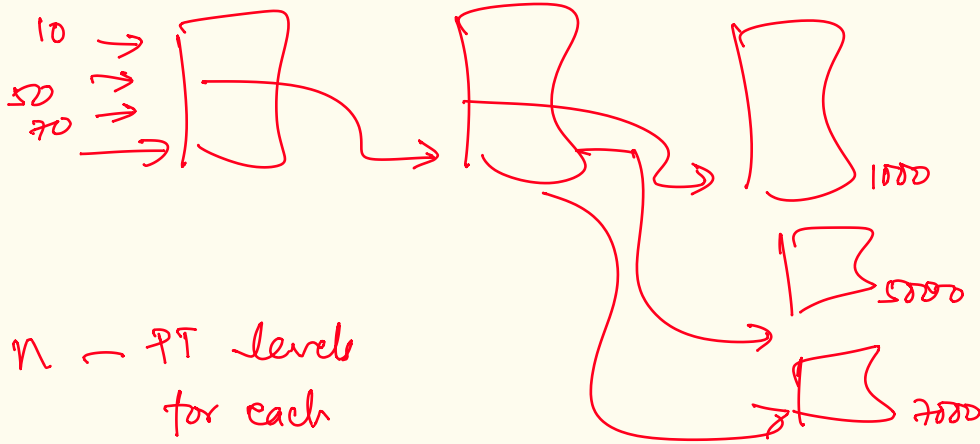
Design 3: H/w assisted memory virtualization
EPT/NPT $\longrightarrow$ nested page tables-extended





Guest.

v2p



p2m

vmm

n - PT levels
for each
PT.

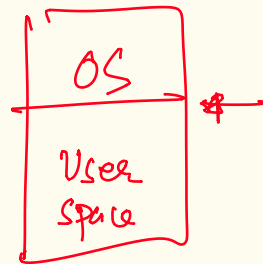
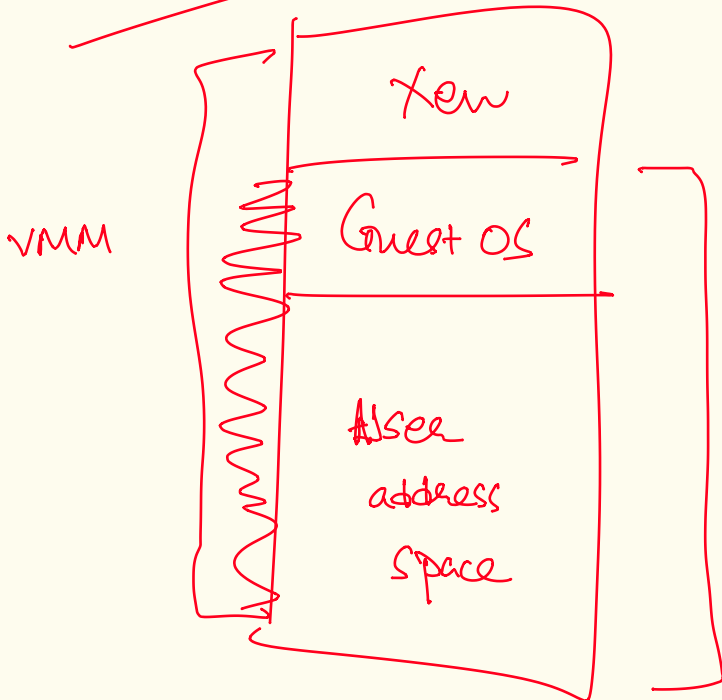
each pte entry of level needs a p2m walk.

$O(n \times n)$



VMM-work

Native



va range
for the guest

segmentation.

base address + offset