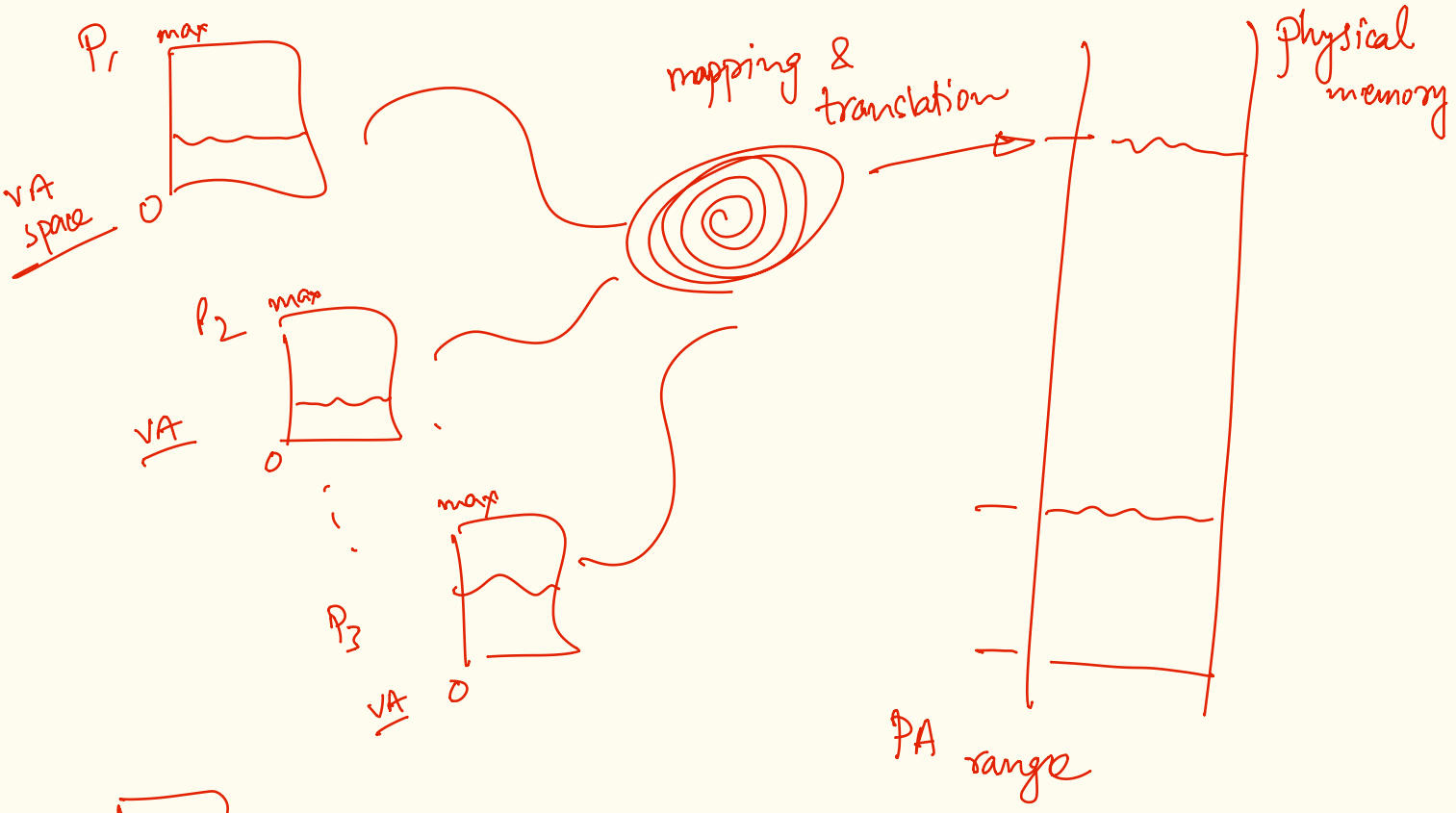
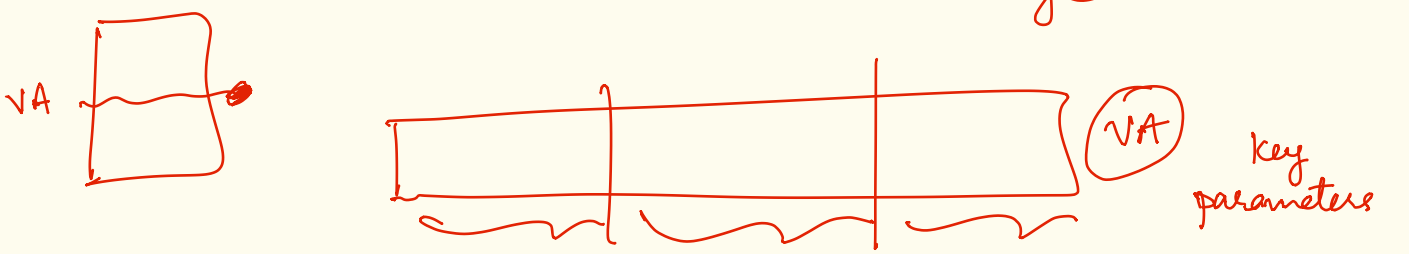
Lecture #13

memory virtualization & memory management.

abstraction ~ process (virtual) address space.

scan mechanism : v2p translation mapping



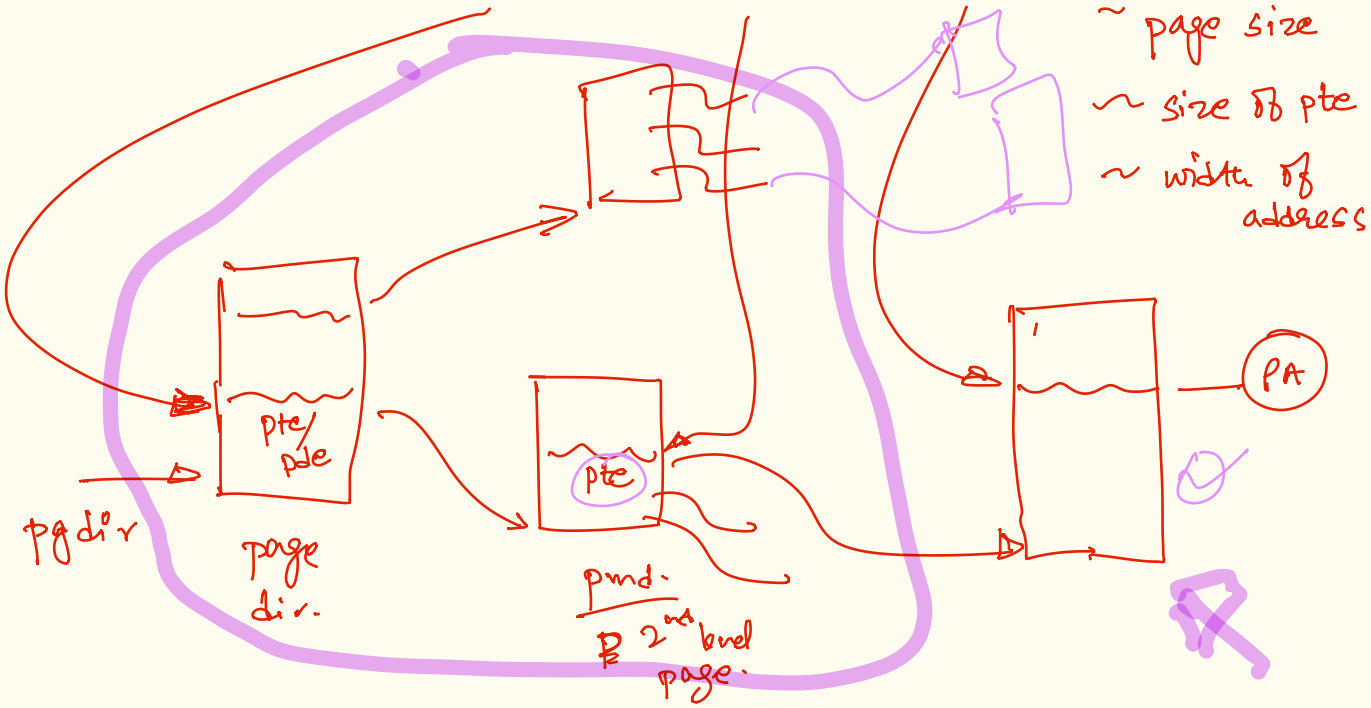


key parameters

~ page size

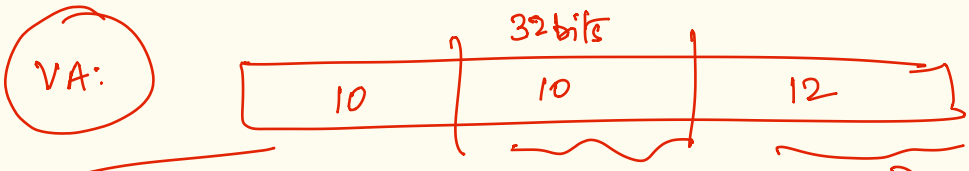
~ size of pte

~ width of address



e.g. (i) 32-bit address range

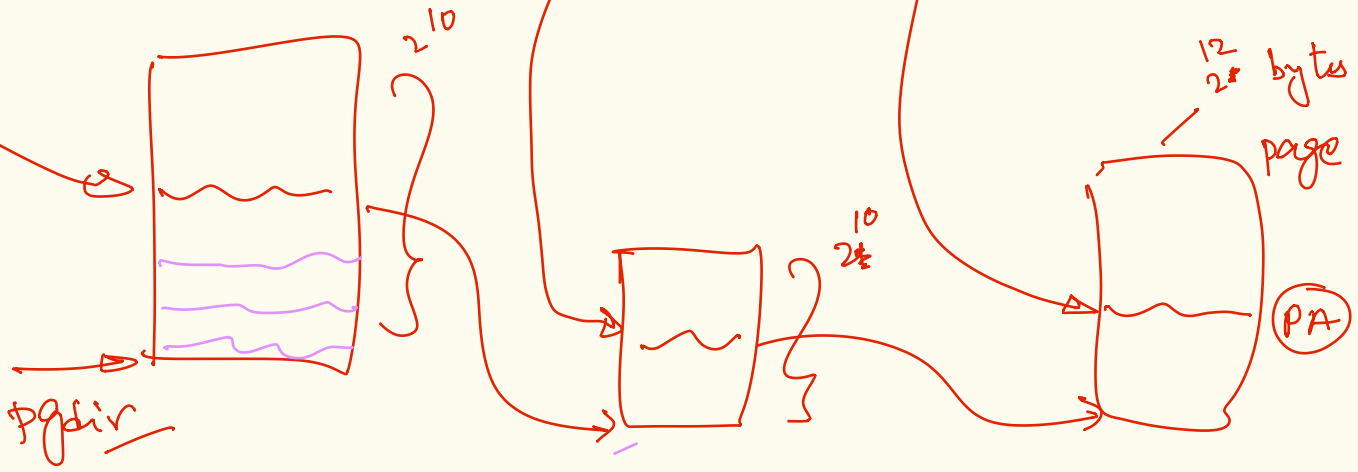
4096 byte pages 4 byte PTEs



PTE points to a physical page. (for access)

1 page = 2^{12} bytes

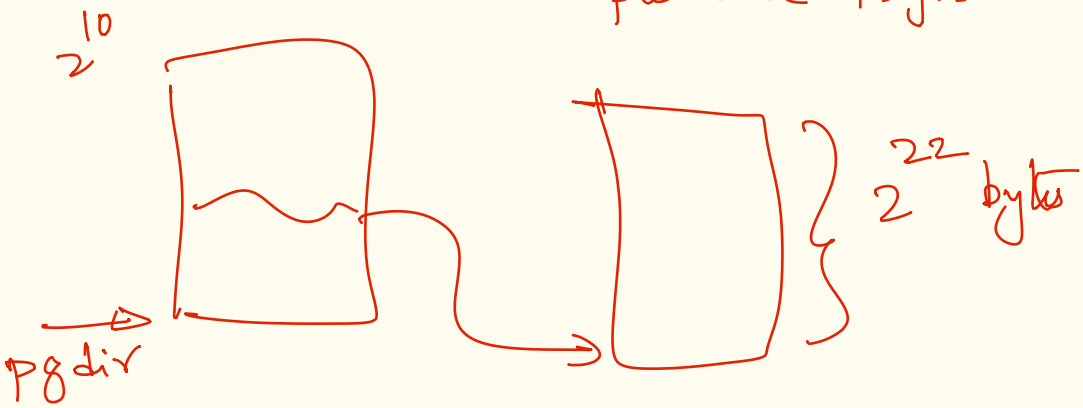
2^{32} address ⇒ 2^{20} pages ⇒ 2^{20} PTEs



(ii) page size — ~~2^{12}~~ 2^{22} ~ 4MB page.

2^{10} ptes for 2^{22} bytes per page

pte size 4 bytes



(2) 32 bit address space

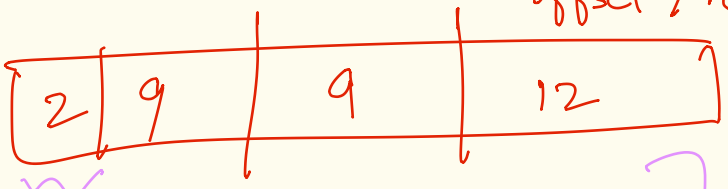
4096 bytes pages $\frac{4096}{2^3} \sim 2^9$

8 byte ptes. $\sim 2^3$

512 entries per page

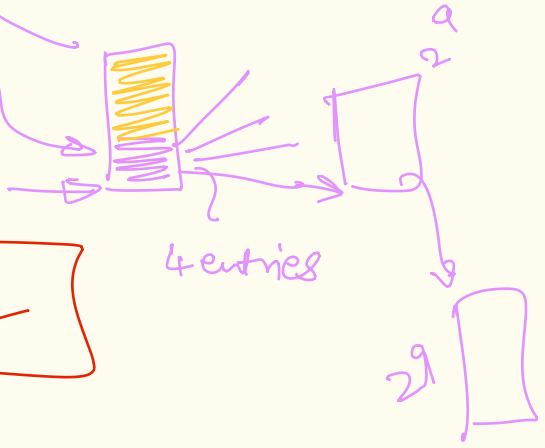
VA:

(i)

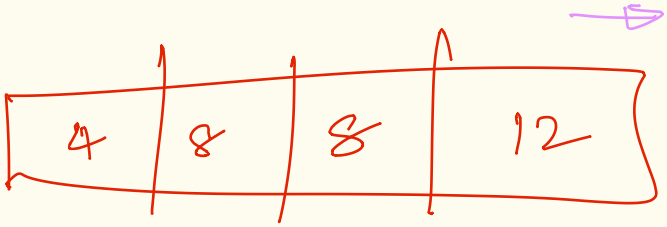


offset / to reach byte on mapped page.

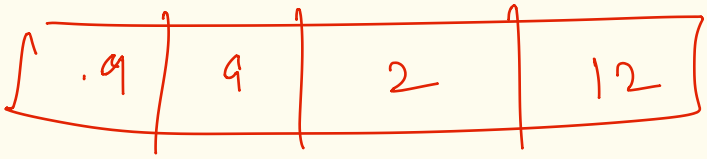
2^9 entries per page table ptes



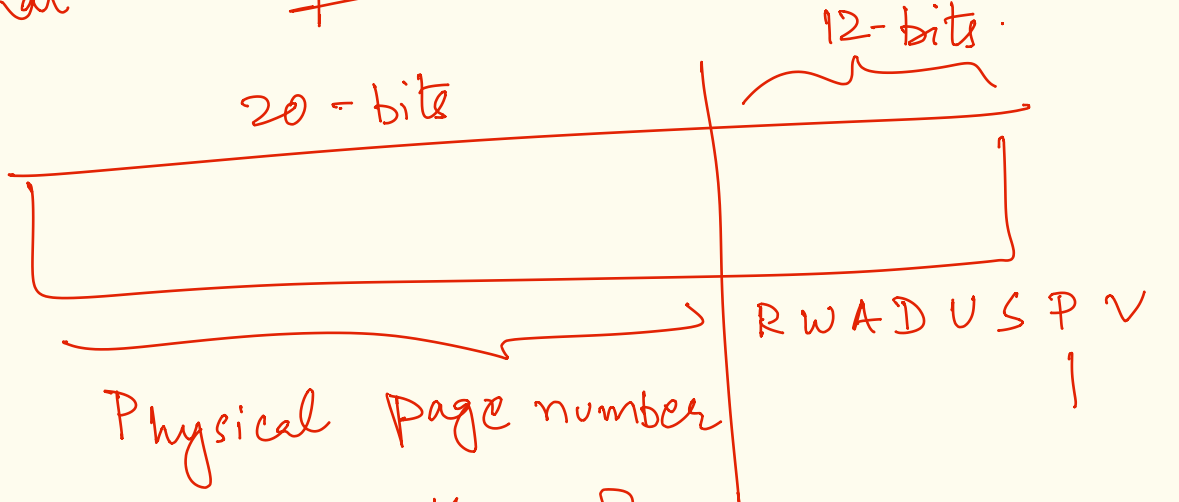
(ii)



(iii)



What is a pte.



⇒ starting address of a page.

overheads of paging / page table based virtualization.

(i) OS needs non-process memory to store page tables.

- incrementally build page tables
- page tables pages can be swapped!
- Sharing of page tables!

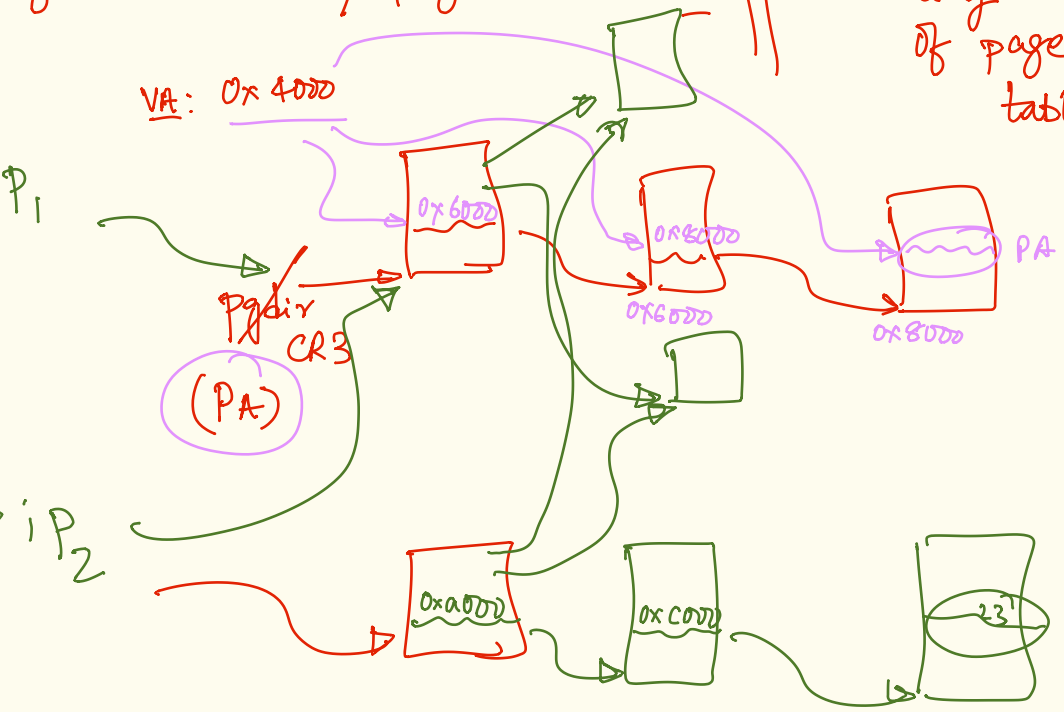
(ii) cost of translation / page table walk.

TLB

```
a = 23;
a = 23;
a = 23;
```

P_1

*(0x4000) = 23; P_2

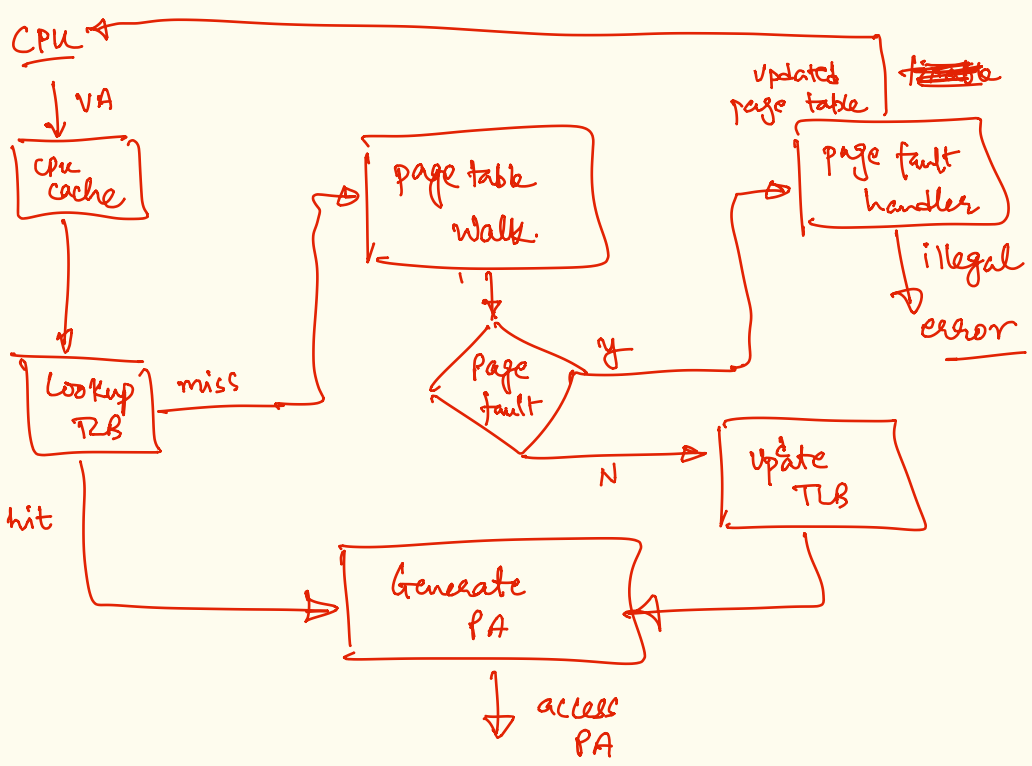


TLB: translation lookaside buffer.

cache: to resolve VA → PA mappings.

store: VPN → MPN entry.

- part of the CPU.



— hardware (MMU), OS, process