

the virtual world enables

Q who can walk the page table?

- ① - mmu (hw page walker)
 - VA → @ → PA
 - pgdir → @ → page fault

② kernel ~ via kernel virtual addresses

↙ because

once the CPU has PAGING enabled in protected mode every address is a virtual address ⇒ translation!

$$pte = *(0x8000 + offset);$$

$$*(\underbrace{(Pa2Va)(0x8000)} + offset);$$

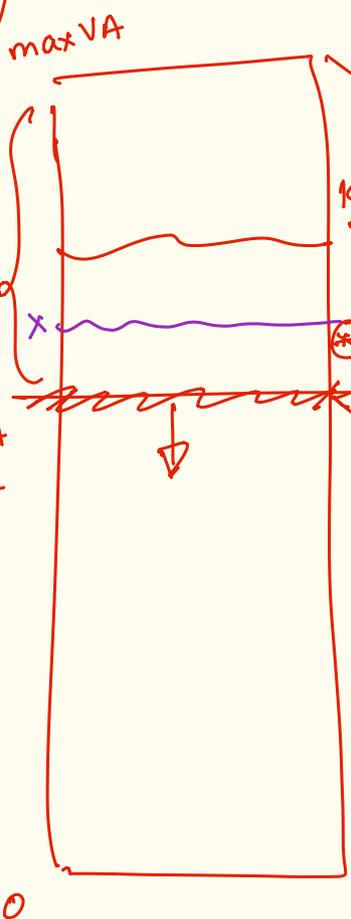
real mode vs protected mode

e.g. 1

VA is mapped for the kernel

real max VA for a user process

P₁

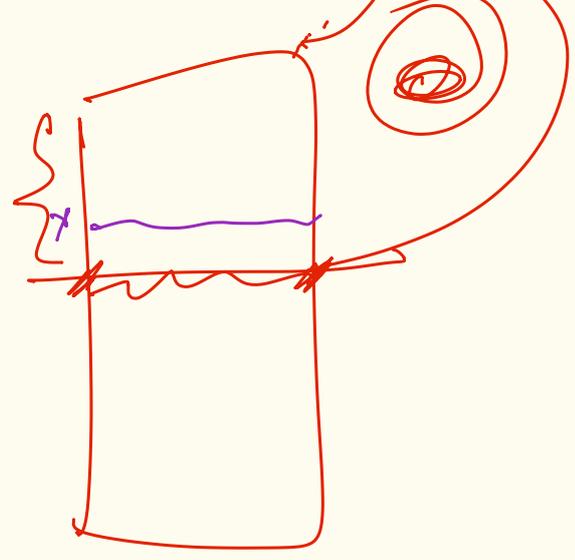


mapping / kernel page table is setup at boot time

when in real mode.

physical address space

pa range



P₂

```
int *p = malloc (sizeof(int) * 10);
```

```
    sbrk(1)  
    *p = 23;
```

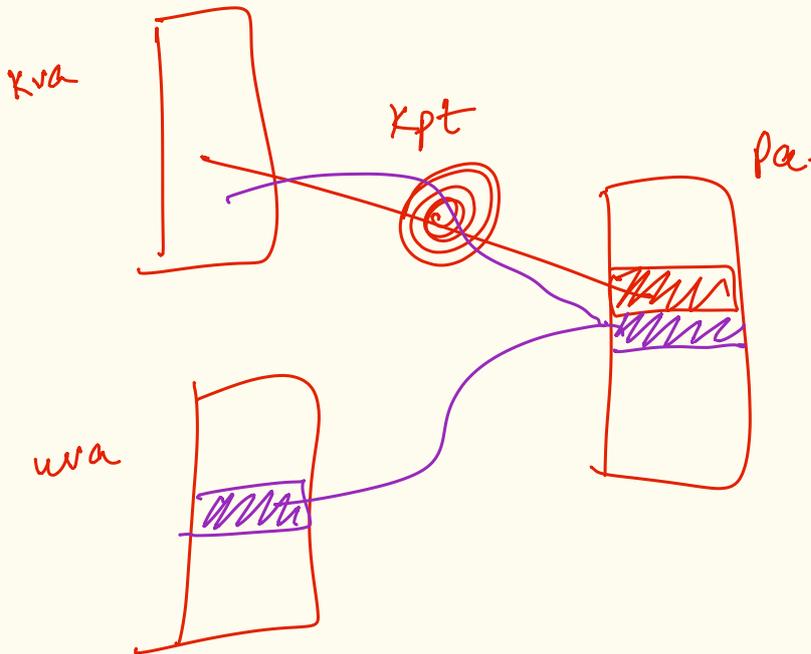
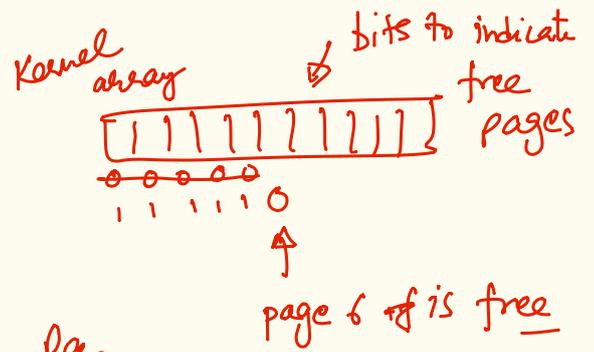
Kernel.

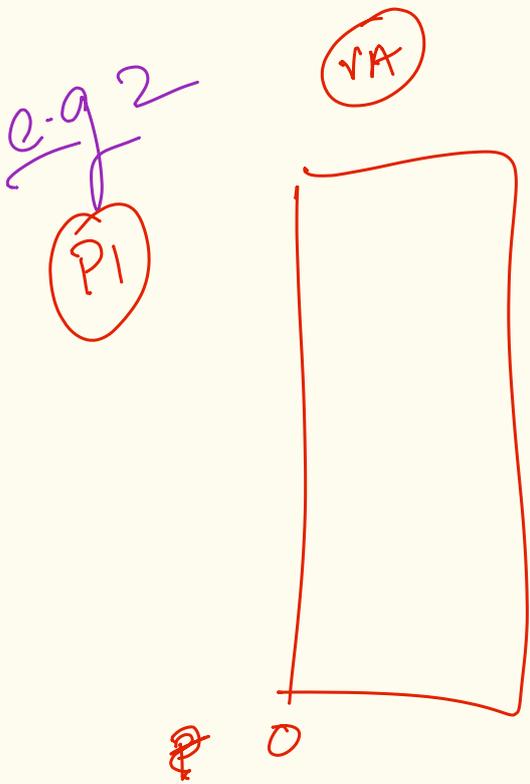


find va-range
build v2p if needed
return va
in page table

- find free page (pa)

- map page (p → pgdir, va, v2p (vx));



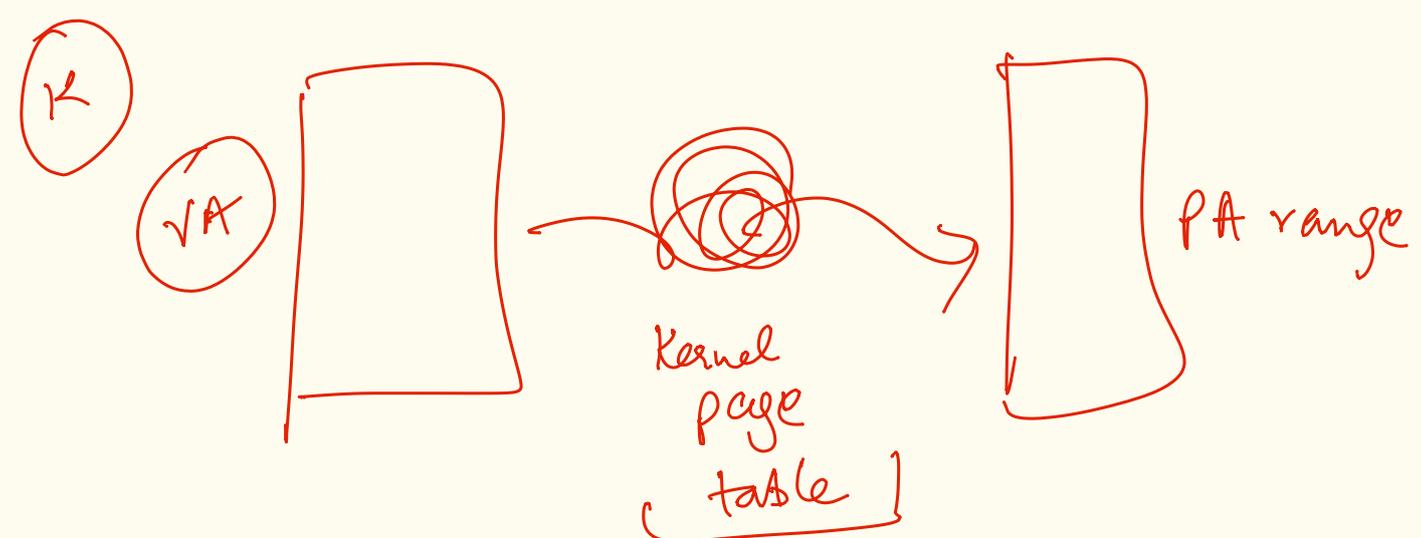


— user page table

&
Kernel page table
are separate.

— on interrupt/ SW to
kernel
mode

switch page tables.



— x86 rise v

separate page tables

1:1 mapping between
kva & (K)pa

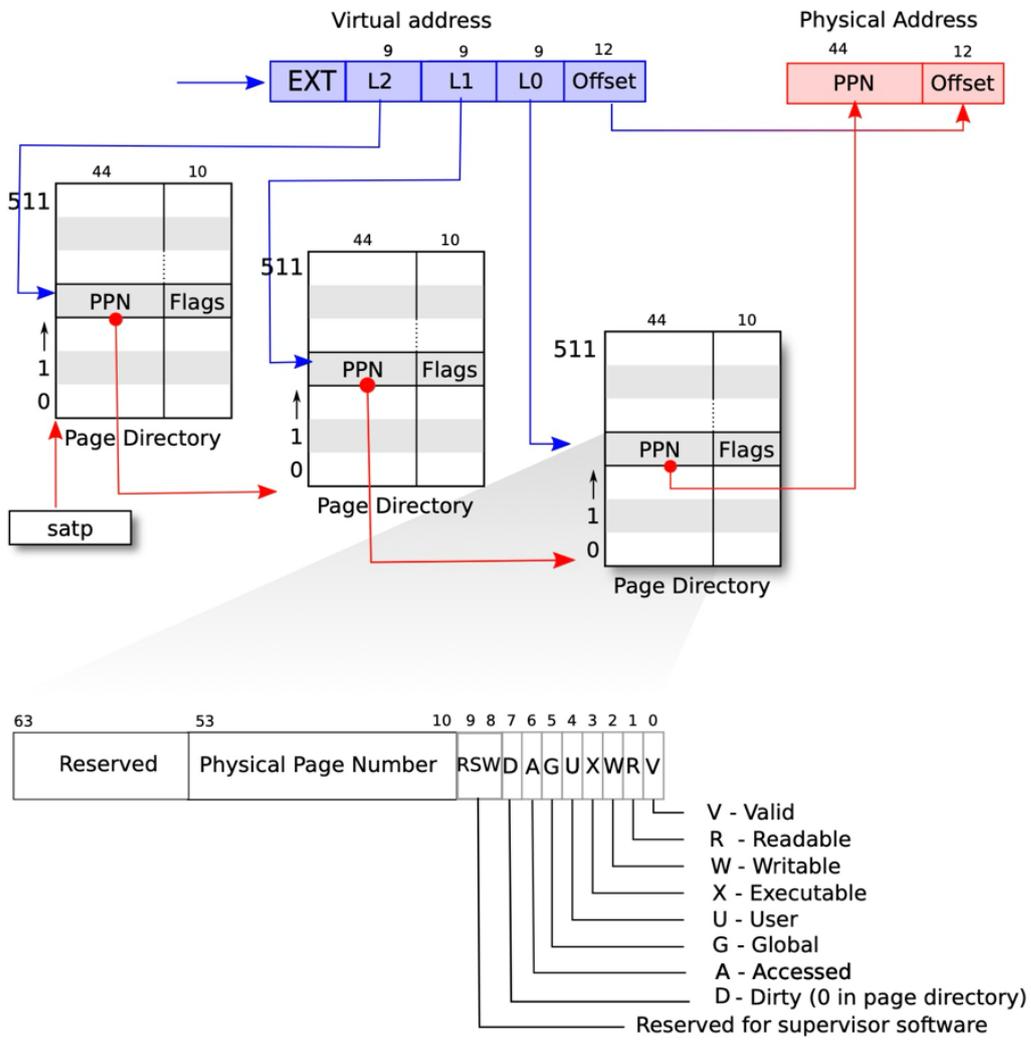


Figure 3.2: RISC-V address translation details.

