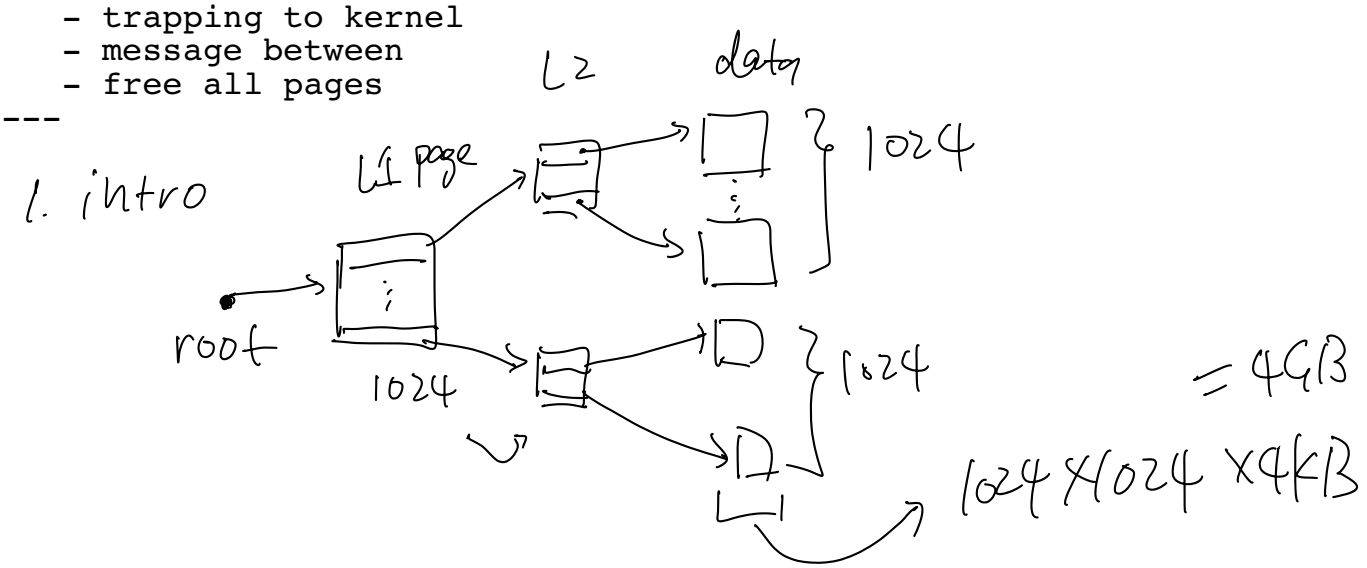
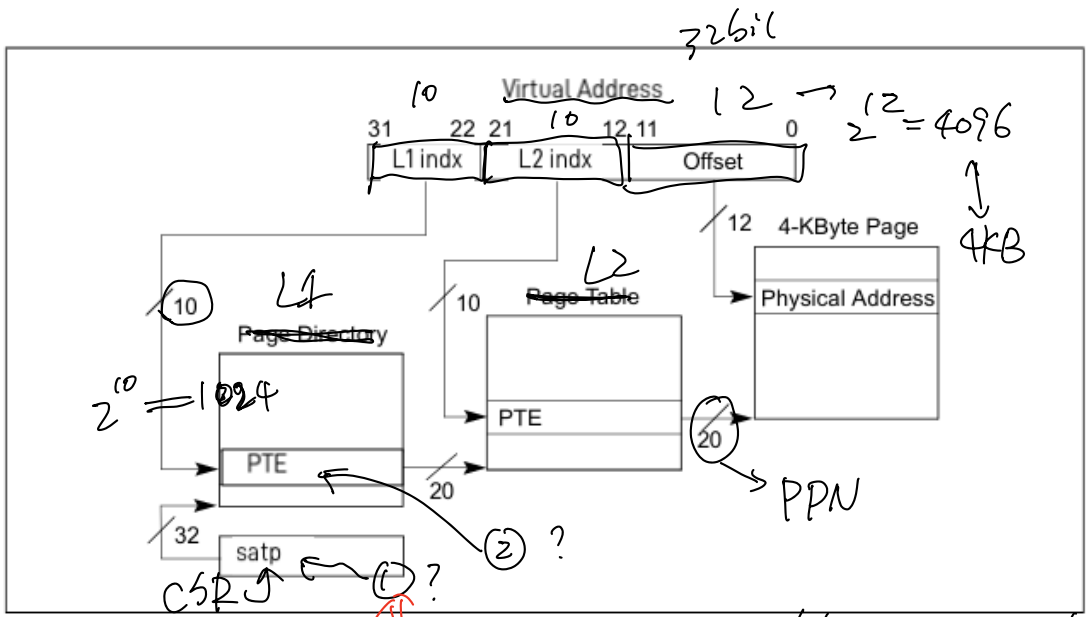
Week 10.a
CS6640
11/07 2023
https://naizhengtan.github.io/23fall/

1. RV32 page table intro
2. (manually) walking page table
3. OS implementation
   - trapping to kernel
   - message between
   - free all pages





VA: 0x80001fec

Q: L1 index, L2 index, offset

10bits 10bits 12bits

L1 index: 10...06 (9) = 0x200

L2 index: 0x1

offset: 0xfec

CS6640 Week10.a

1. CSR satp (page table root)

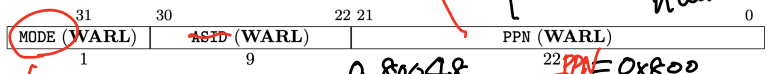


Figure 4.14: Supervisor address translation and protection register satp when SXLEN=32.

(1-bit MODE)

SXLEN=32		
Value	Name	Description
0	Bare	No translation or protection.
1	Sv32	Page-based 32-bit virtual addressing (see Section 4.3).

2. Page table entry (PTE)

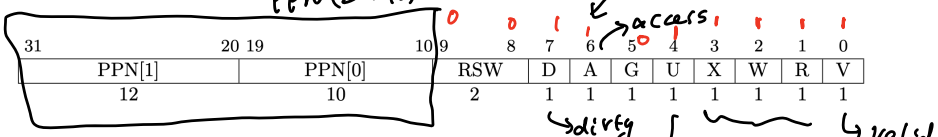


Figure 4.18: Sv32 page table entry.

3. Virtual address (VA)

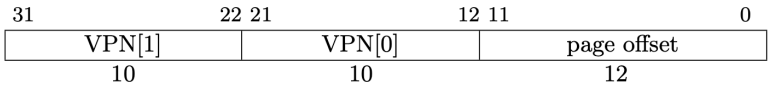
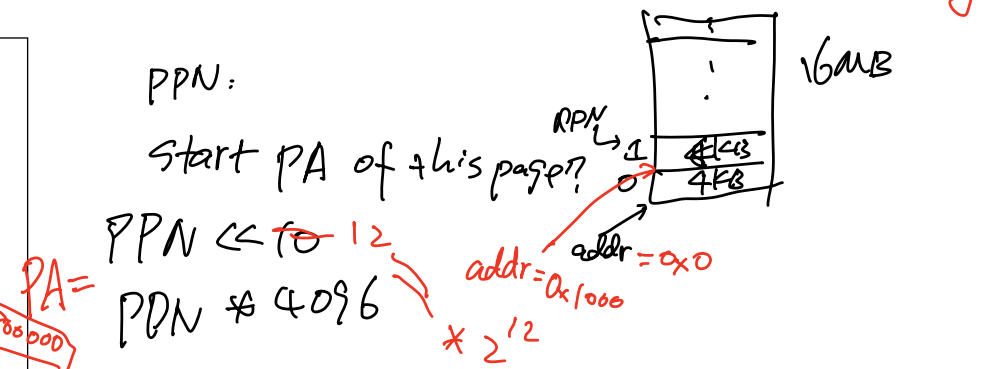
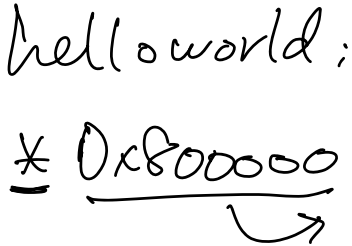


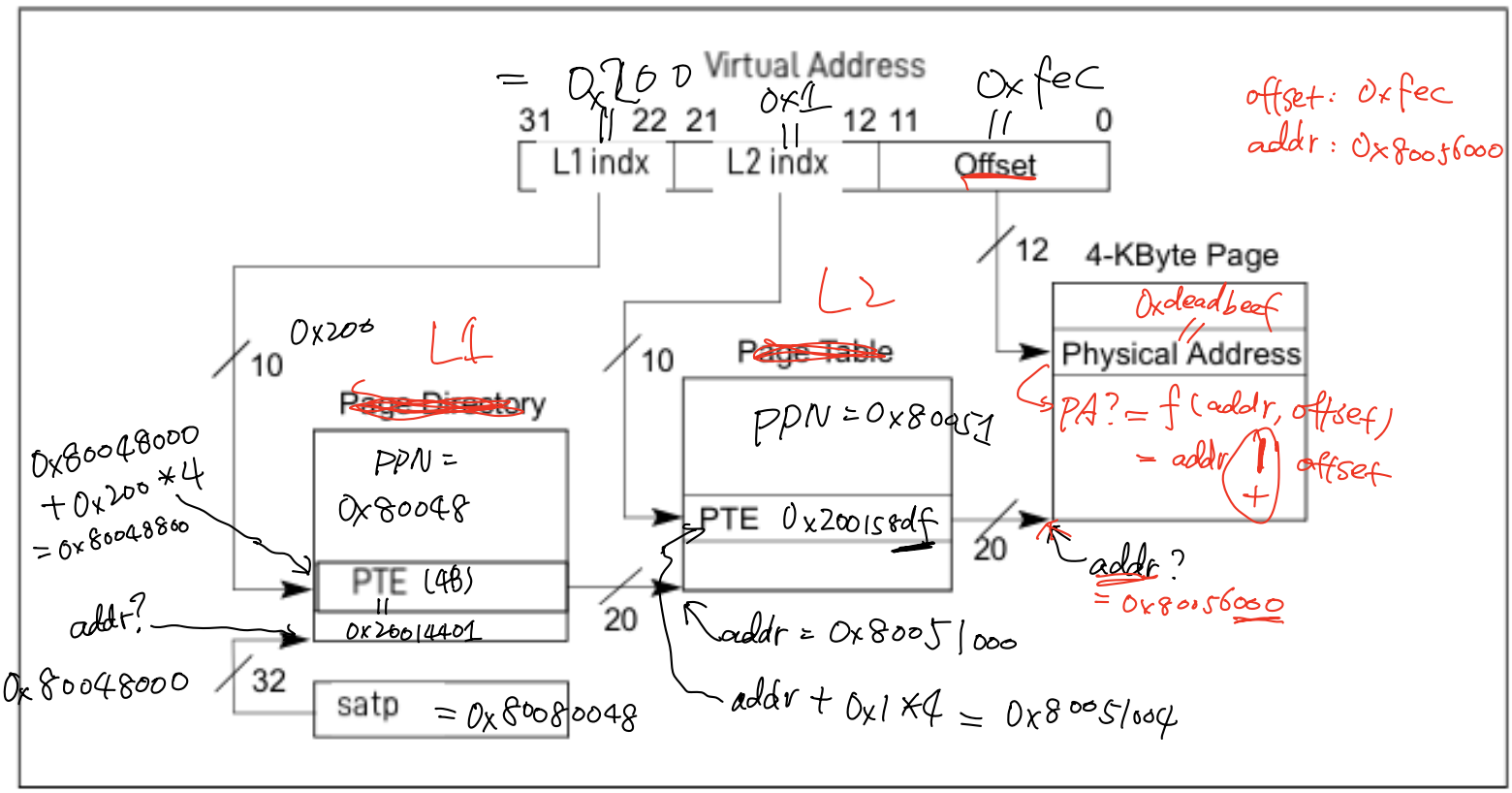
Figure 4.16: Sv32 virtual address.





* simulate CPU: manual page walk

  Steps:
  (1) split the VA to l1/l2 indexes and offest
  (2) get the root of page table (satp)
  (3) calc the L1 page
  (4) calc the L2 page
  (5) calc the physical address



3. OS implementation
   - trapping to kernel
   - message between ≥ processes
   - free all pages when exit