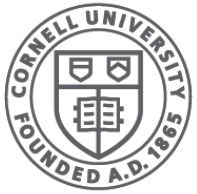
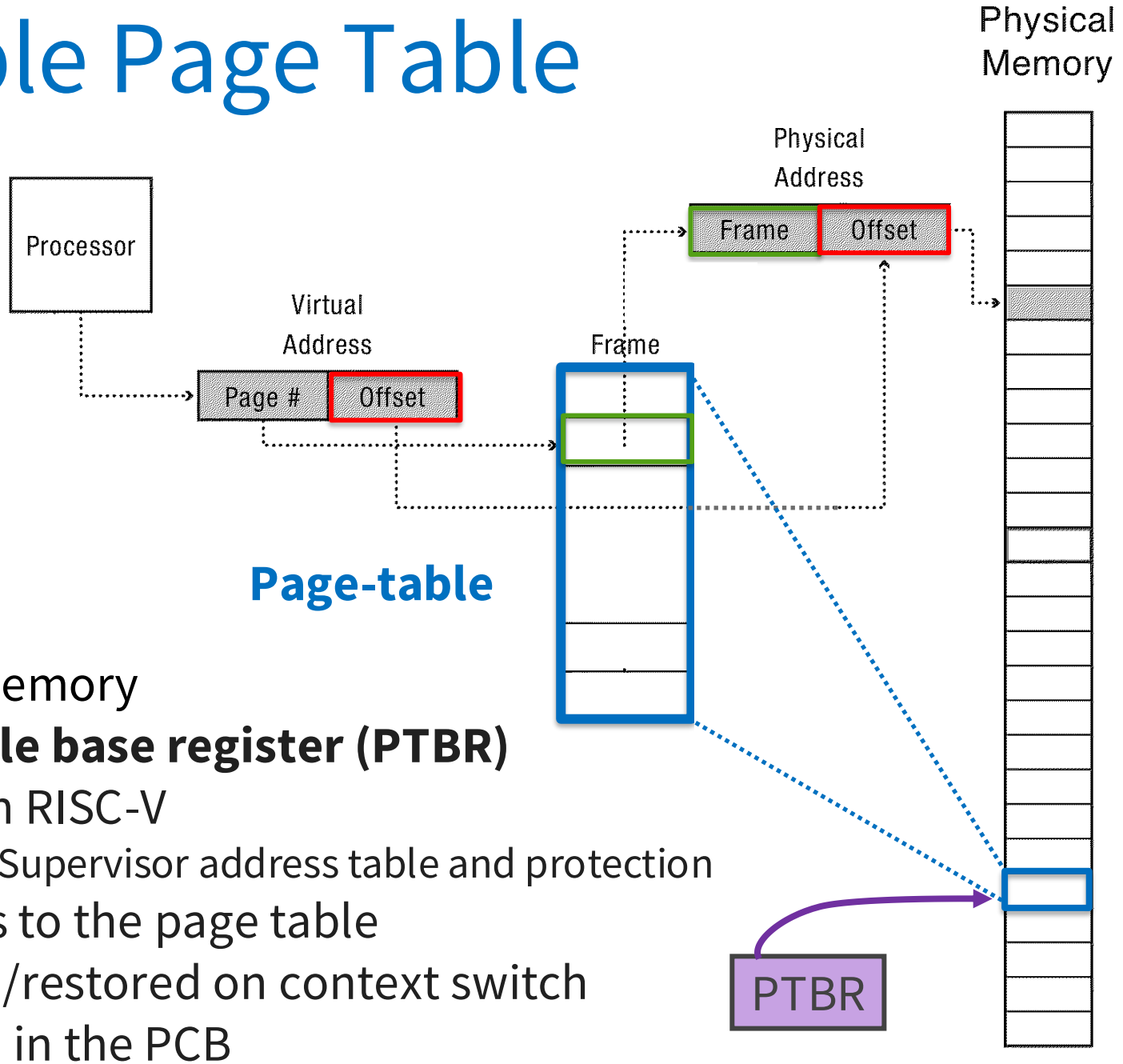
# P5: Paging

CS 4411
Operating Systems




Cornell CIS
COMPUTING AND INFORMATION SCIENCE

# Simple Page Table

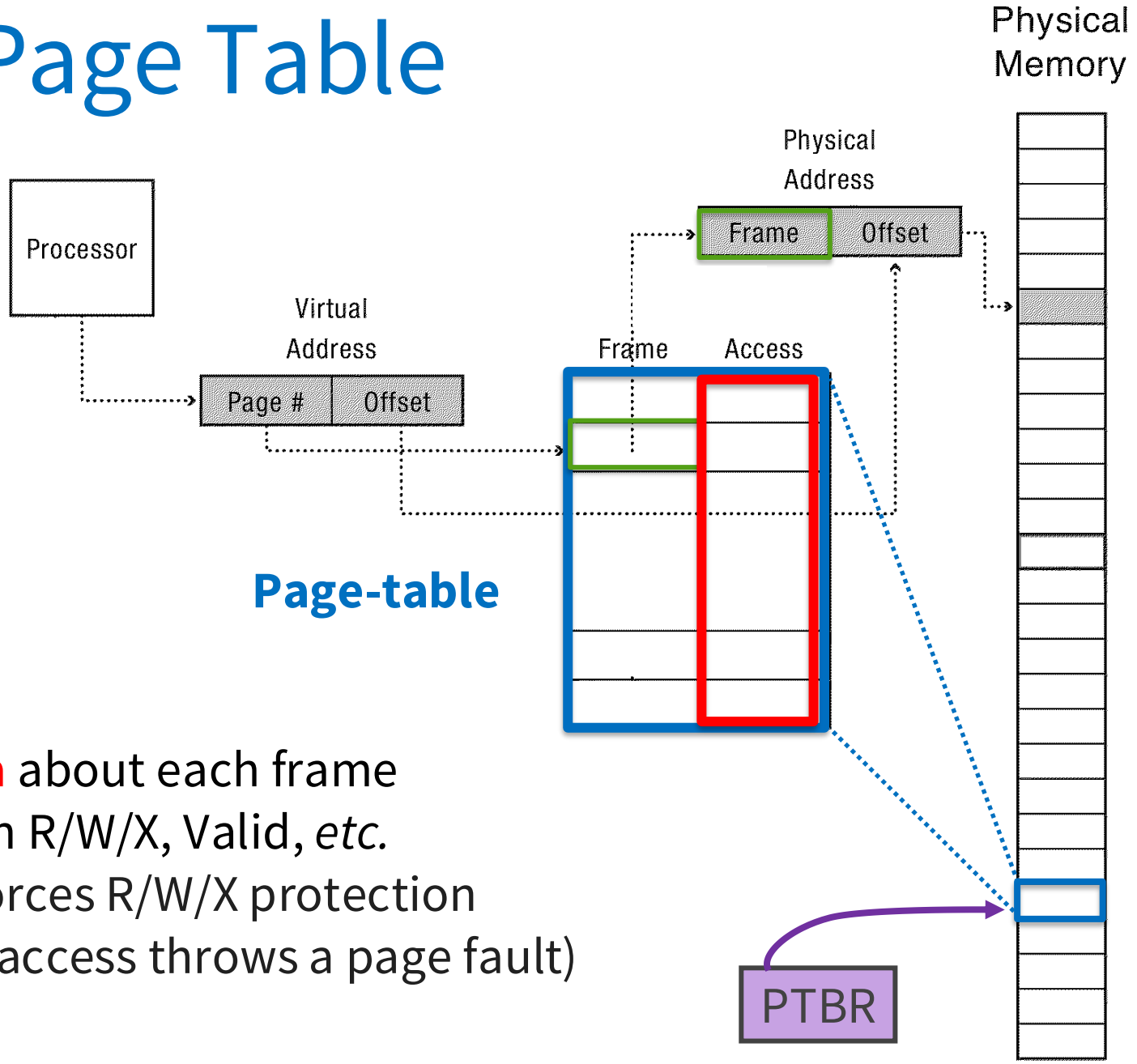



Lives in Memory

**Page-table base register (PTBR)**

- *satp* in RISC-V
  - Supervisor address table and protection
- Points to the page table
- Saved/restored on context switch
- Saved in the PCB

# Full Page Table




Meta Data about each frame

Protection R/W/X, Valid, *etc.*

MMU Enforces R/W/X protection

(illegal access throws a page fault)

# Complete Page Table Entry (PTE)


***Index*** is an index into (depending on Present bit):

- frames
  - physical process memory
- backing store
  - if page was swapped out

Synonyms:

- Present bit == Valid bit
- Modified bit == Dirty bit
- Referenced bit == Accessed bit
- Index == frame number

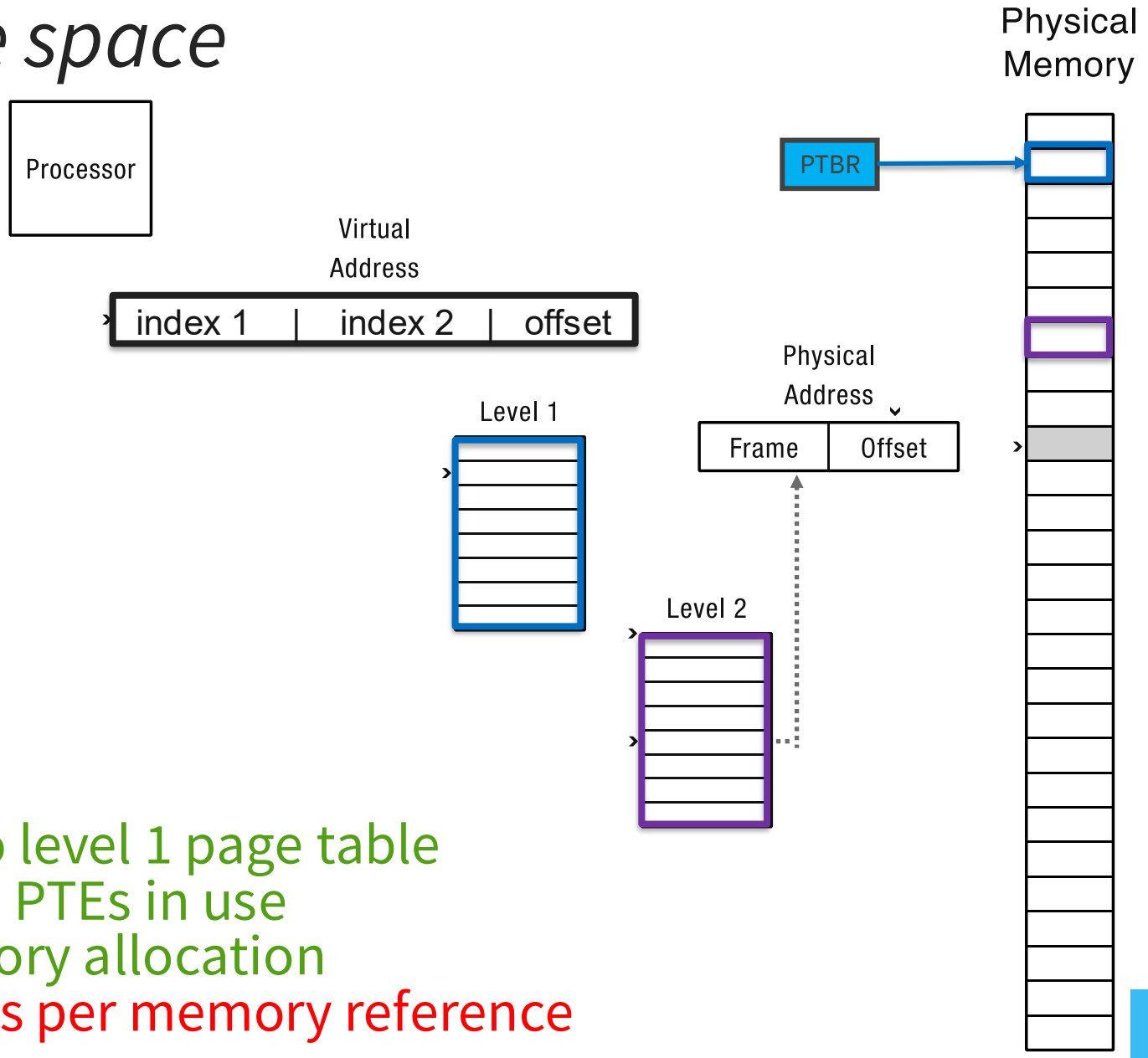
# How large must the page table be?

- Suppose:
  - 32-bit address space
  - 32-bit PTEs (4 bytes)
  - 12-bit offset (4K pages)

→

- 32 – 12 = 20-bit page number
  - That is, $2^{20}$ = approx. 1 million PTEs
- Page table is $2^{20}$ x 4 = $2^{22}$ bytes (4 M)
  - For each process!
  - Does not fit in a frame…

# *Multi-Level Page Tables to reduce page table space*

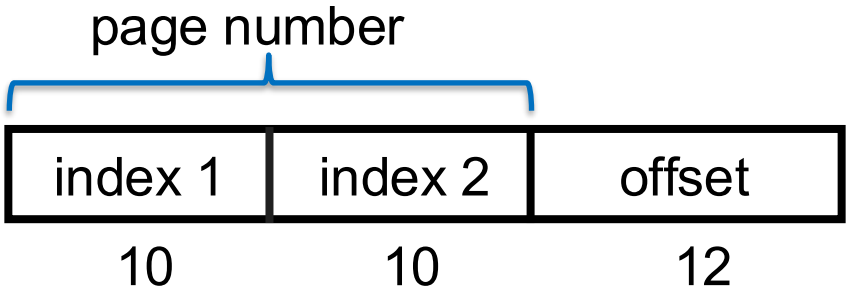



PTBR points to level 1 page table
+ Allocate only PTEs in use
+ Simple memory allocation
– ***more*** lookups per memory reference

# Two-Level Paging Example

32-bit machine, 4KB page size

- Logical address is divided into:
  - a page offset of 12 bits ($4096 = 2^{12}$)
  - a page number of 20 bits (32-12)
- Since the page table is paged, the page number is further divided into (say):
  - a 10-bit first index
  - a 10-bit second index
- Thus, a logical address is as follows:



# Let’s run the numbers

- Suppose 32-bit virtual address, 2-level page table
  - address is 10+10+12 bits
- Page Table Entry (PTE) is 32 bits (4 bytes)
  - convenient: PTE is the size of a word
- How many pages in the virtual address space?
  - Answer: $2^{10} \times 2^{10} = 2^{20}$
- How many PTEs in a page table?
  - Answer: $2^{10} = 1024$
- How large is a page table?
  - Answer: $2^{10} \times 4 = 4$ KB
  - conveniently fits in a frame

# How many levels for 64-bit RISC-V?

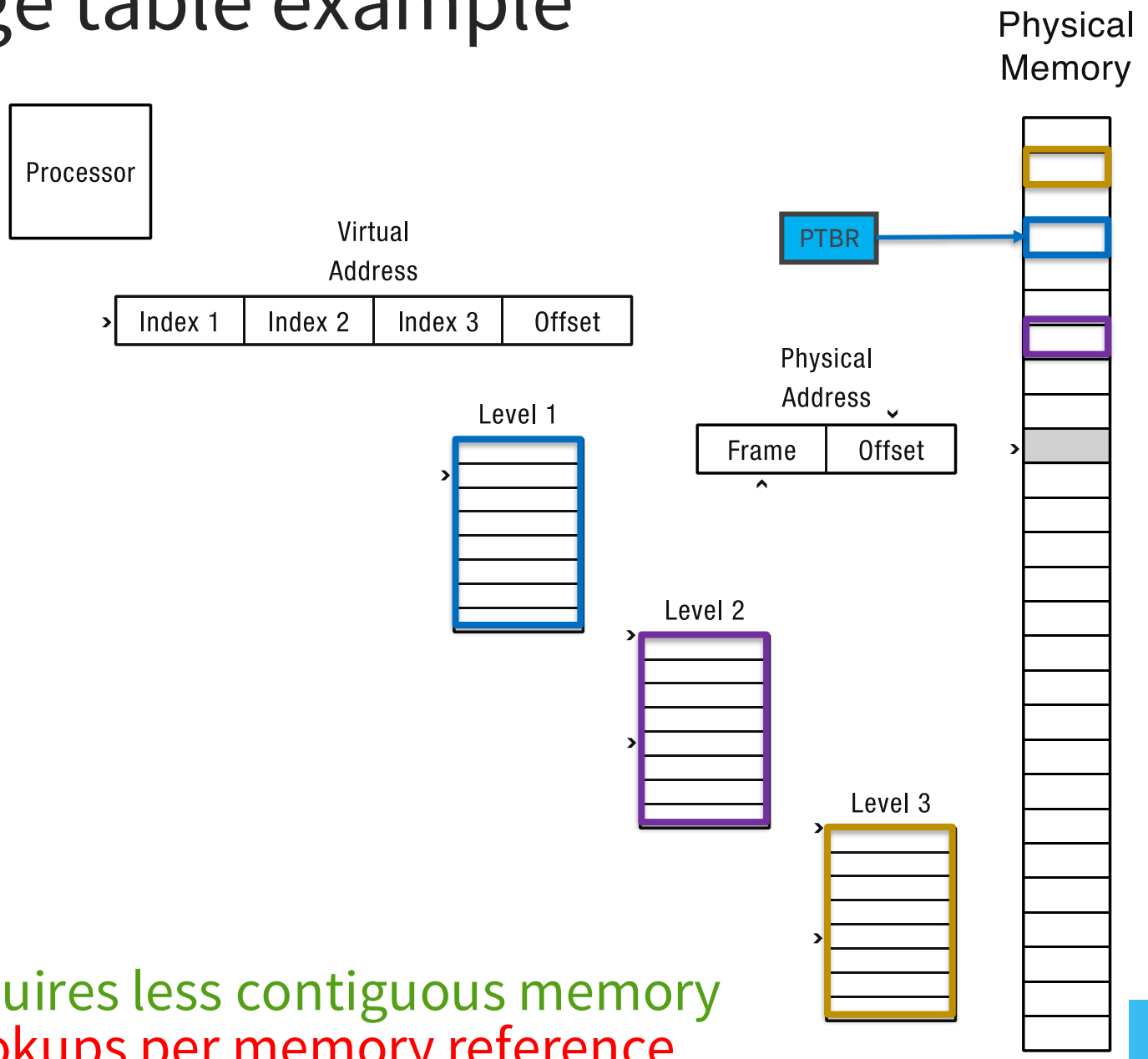
- 39-bit address (25 bits unused)
- 12-bit page offset (4K pages)
- 64-bit PTE (8 bytes)

→

- A page table can fit 4K/8 = 512 ($2^9$) PTEs
  → An index must be 9 bits
- A page number is 39 – 12 = 27 bits
- 27 / 9 = 3 levels

| unused | index 1 | index 2 | index 3 | page offset |
|---|---|---|---|---|
| 25 | 9 | 9 | 9 | 12 |

# 3 level page table example




+ First Level requires less contiguous memory
– ***even more*** lookups per memory reference