COS 417 Precept 4

Paging

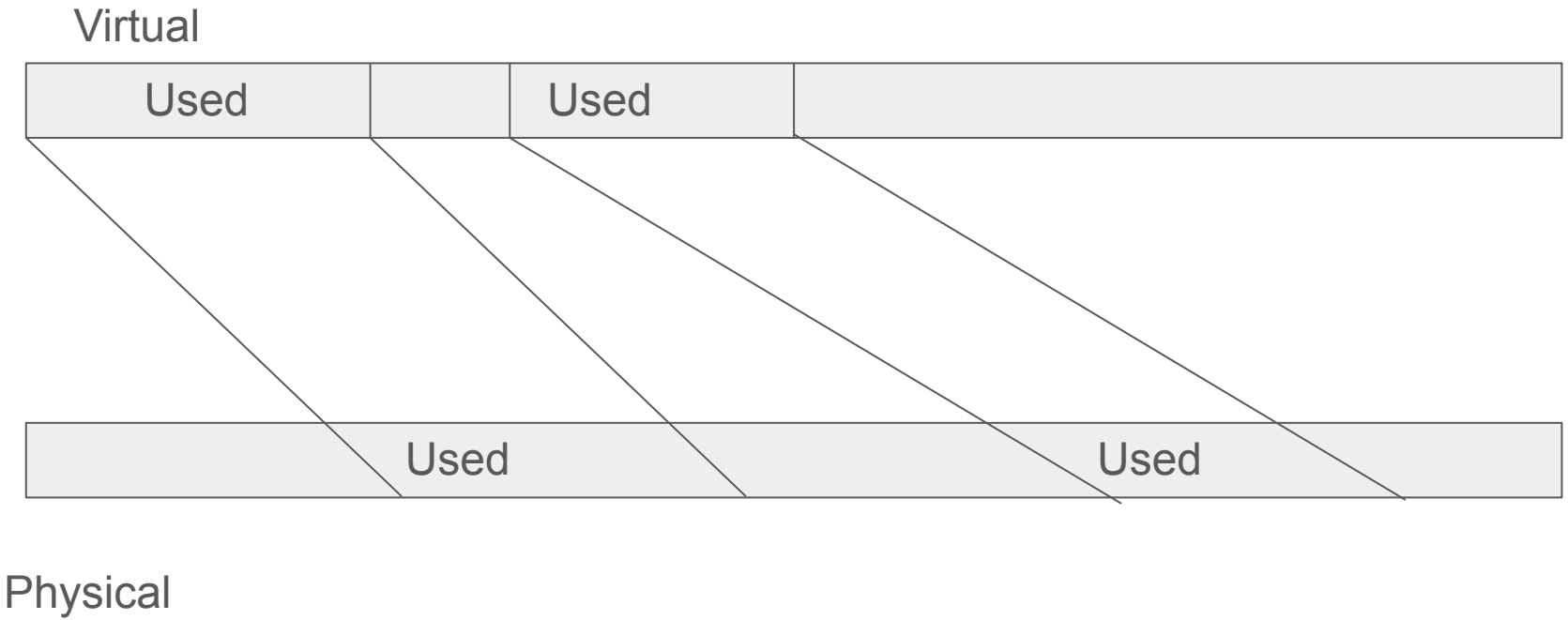
High level idea:

- Break the virtual memory up into constant-sized pages
- Memory Management Unit handles allocation of virtual pages to physical pages
  - To keep track of mappings, MMU uses page tables.
  - Multiple potential page table structures, xv6 uses 2 tier paging.

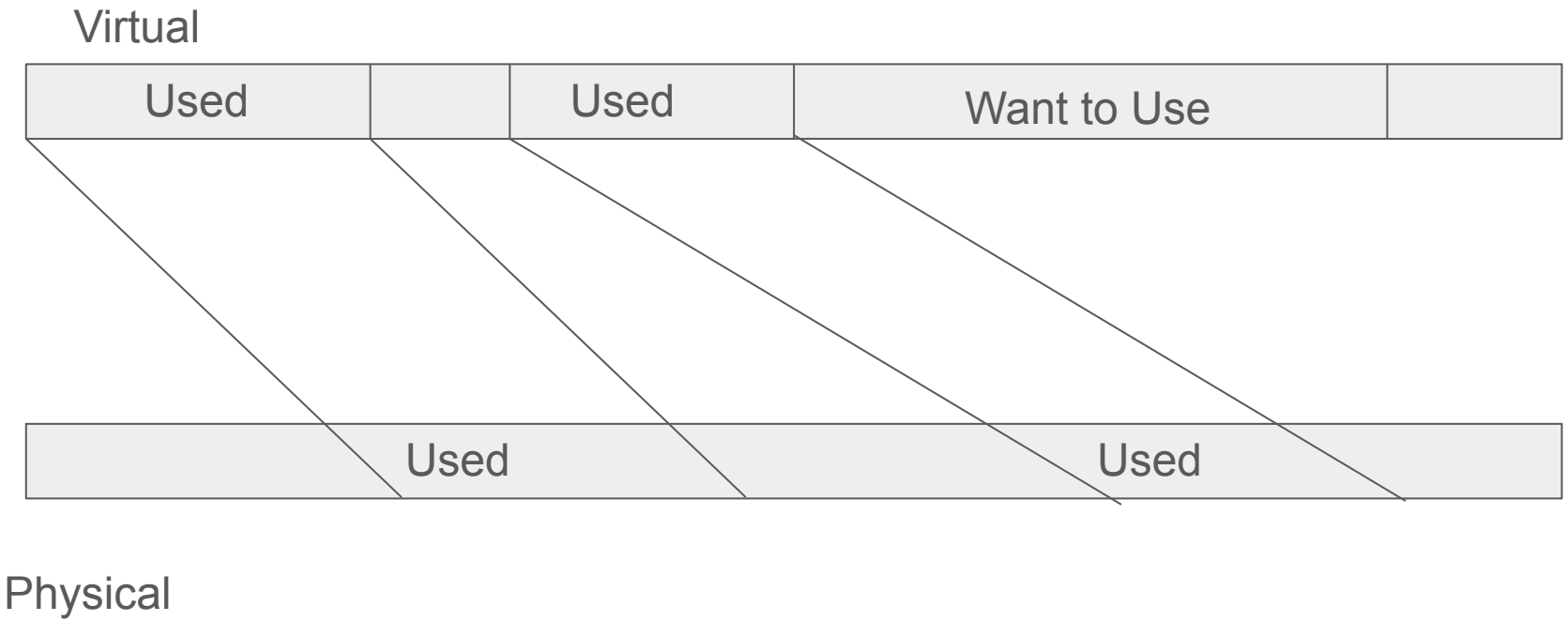
Reference:

https://pages.cs.wisc.edu/~remzi/OSTEP/vm-paging.pdf

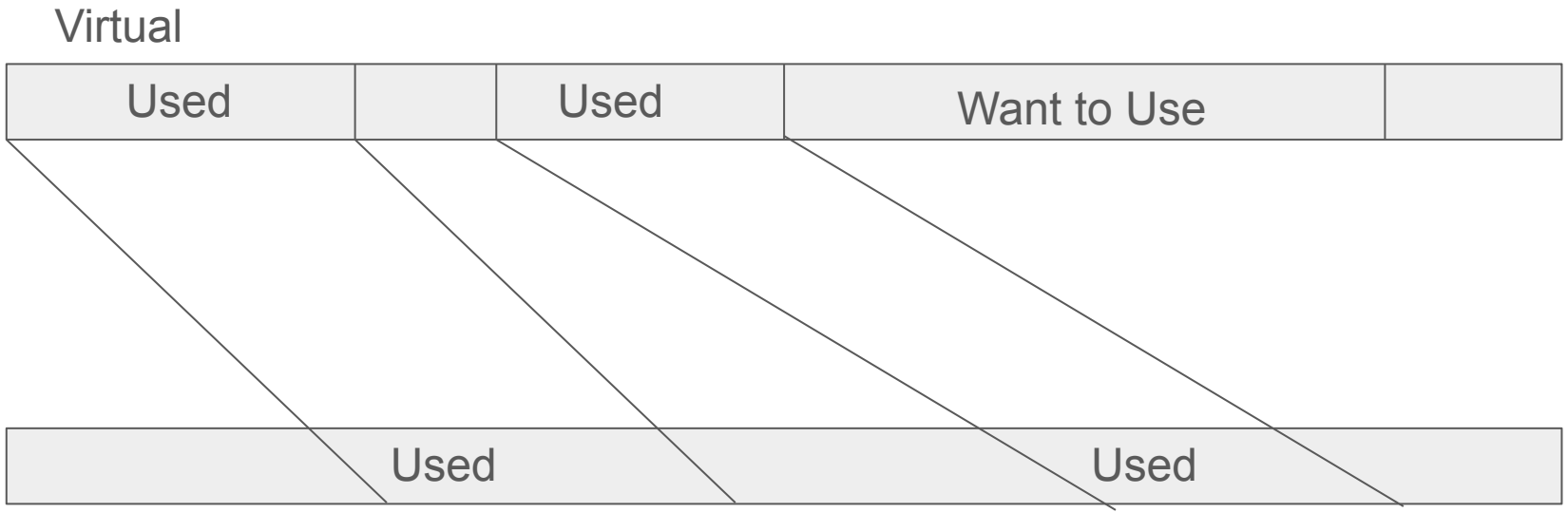
Without Page Tables



Without Page Tables



Without Page Tables



There exists enough physical memory, but fragmentation prevents allocation!

Why Page Tables?

- An MMU that allocates physical memory with dynamic sizes runs risk of fragmentation
  - Non-contiguous space leads to inefficient use of physical memory.
- In the example, there was enough space for the new virtual memory, but fragmentation prevented allocation.
- Instead, have the MMU focus on allocating constant sizes of memory.
  - Higher-level functions (malloc, free) handles dynamic sizes.

Page Table Addresses

- Each address contains two parts:
  - Virtual Address
  - Offset

va	va	va	offset	offset	offset	offset

Page Table Addresses

- Each address contains two parts:
  - Virtual Address
  - Offset

va	va	va	offset	offset	offset	offset

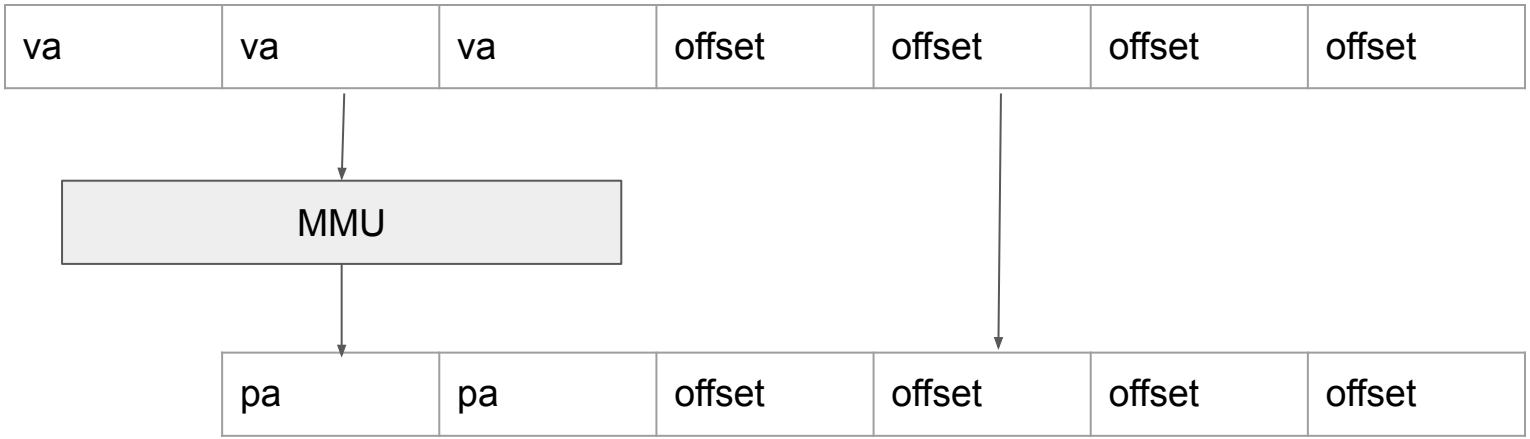
Page Table Addresses

Logarithmic relationship between page size and memory size

# Virtual Pages	Size of Page	Offset Bits	Page Bits	Total # Bits
64	64	6	6	12
4kb	1kb	10	12	22
1mb	1kb	??	??	??

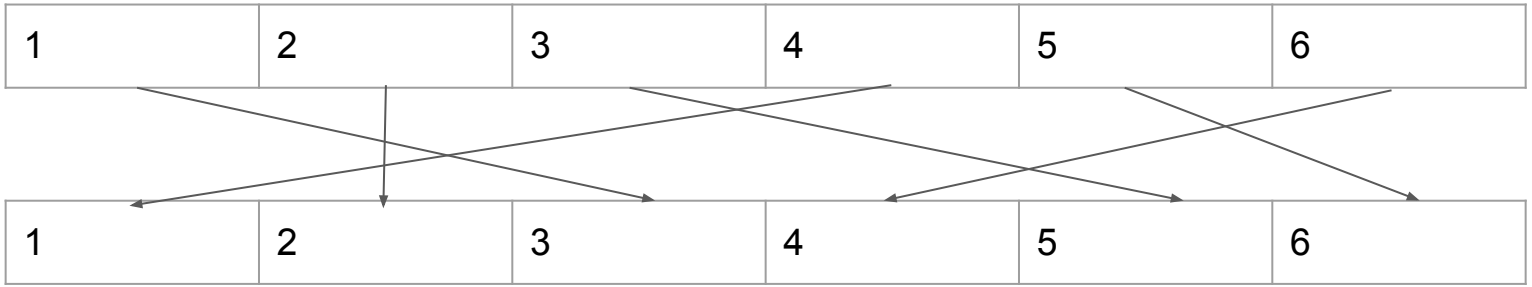
VA to PA

MMU takes care of the translation



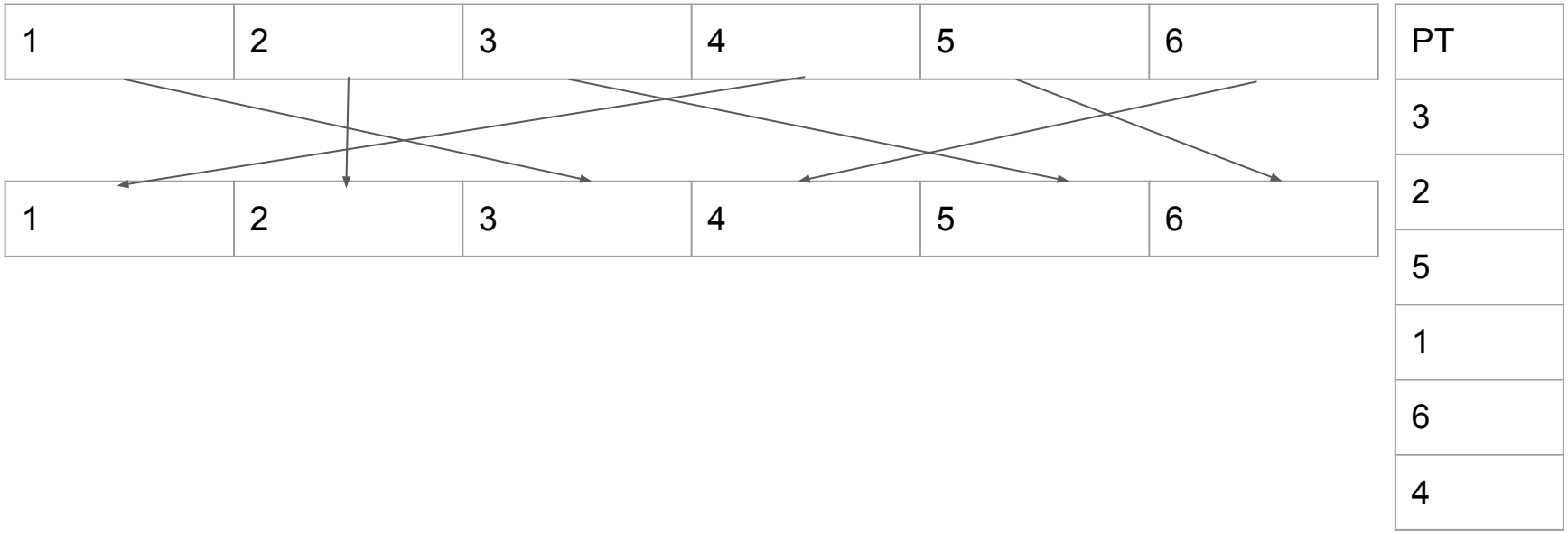
VA to PA

Virtual address to physical address is handled by page tables.



VA to PA

Virtual address to physical address is handled by page tables.



VA to PA

Steps to read physical from virtual memory

1. Load page table into memory
2. Look up PA using page table address
3. Compute physical address using PA + offset
4. Read from physical address

1. is quite expensive!

Multilevel Paging

Solution: Multiple Page levels

1. Instead of loading the entire page table, load each segment individually.

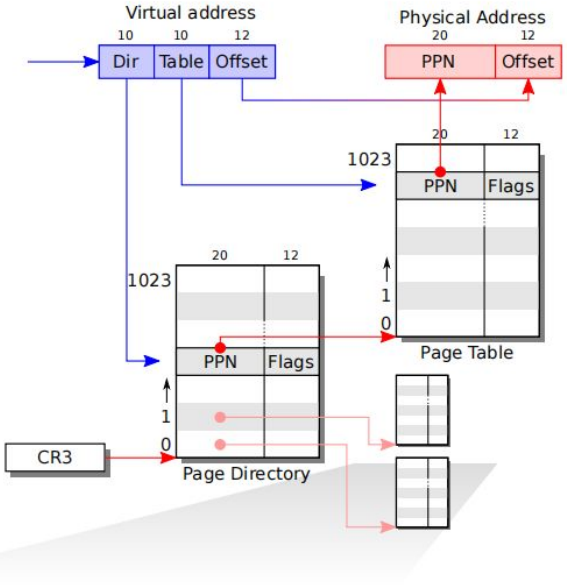
Multilevel Paging

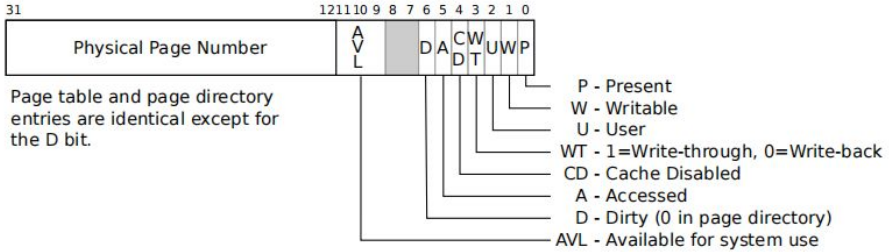
Solution: Multiple Page levels

1. Instead of loading the entire page table, load each segment individually.

It’s used in XV6!

Paging in xv6





Page table entries in xv6

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Address of page directory[1]																				Ignored							PCD	PWT	Ignored			CR3
Bits 31:22 of address of 4MB page frame										Reserved (must be 0)					Bits 39:32 of address[2]				PAT	Ignored			G	1	D	A	PCD	PWT	U/S	R/W	1	PDE: 4MB page
Address of page table																				Ignored				0	Ign	A	PCD	PWT	U/S	R/W	1	PDE: page table
Ignored																															0	PDE: not present
Address of 4KB page frame																				Ignored				G	PAT	D	A	PCD	PWT	U/S	R/W	1	PTE: 4KB page
Ignored																															0	PTE: not present

Page address in xv6

3 components:

- Page directory address
- Page Table address
- Offset

10 bit directory	10 bit address	12 bit offset