Betriebssysteme und Systemnahe Programmierung

Kapitel 8 • Paging

Winter 2016/17

Marcel Waldvogel

Paging (1)

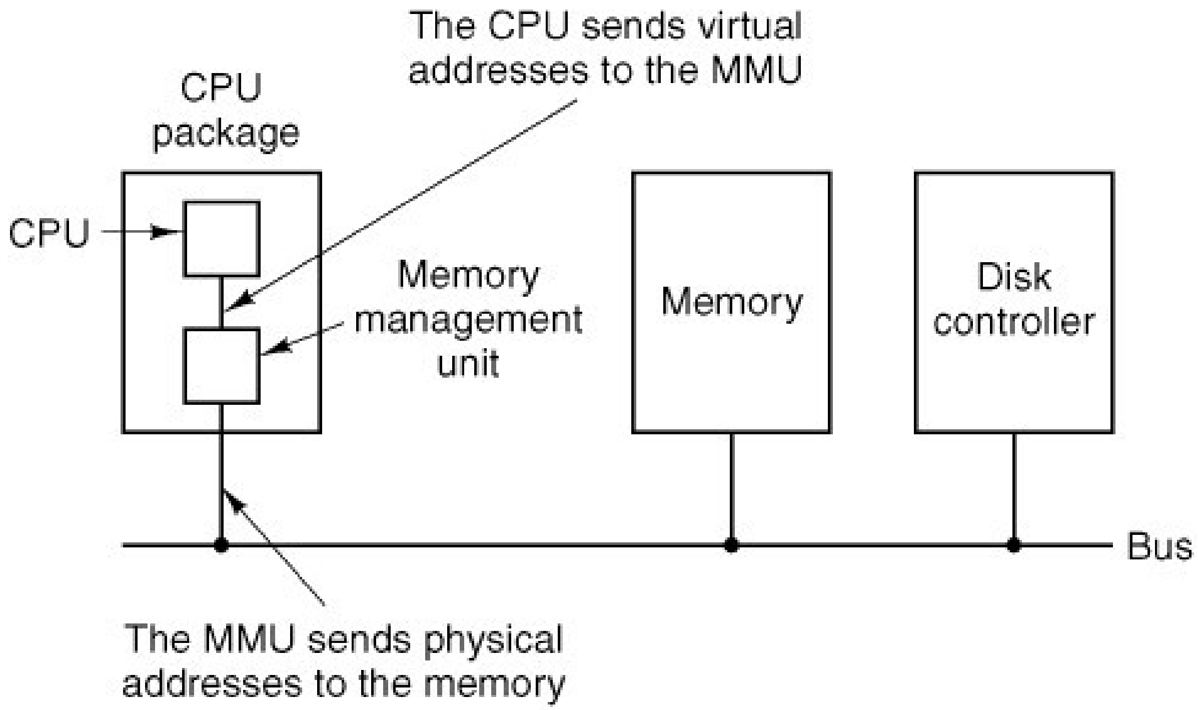

The CPU sends virtual addresses to the MMU
CPU package
CPU
Memory management unit
Memory
Disk controller
Bus
The MMU sends physical addresses to the memory

Figure 4-7. The position and function of the MMU. Here the MMU is shown as being a part of the CPU chip because it commonly is nowadays. However, logically it could be a separate chip and was in years gone by.

Tanenbaum & Woodhull, Operating Systems: Design and Implementation, (c) 2006

Paging (2)

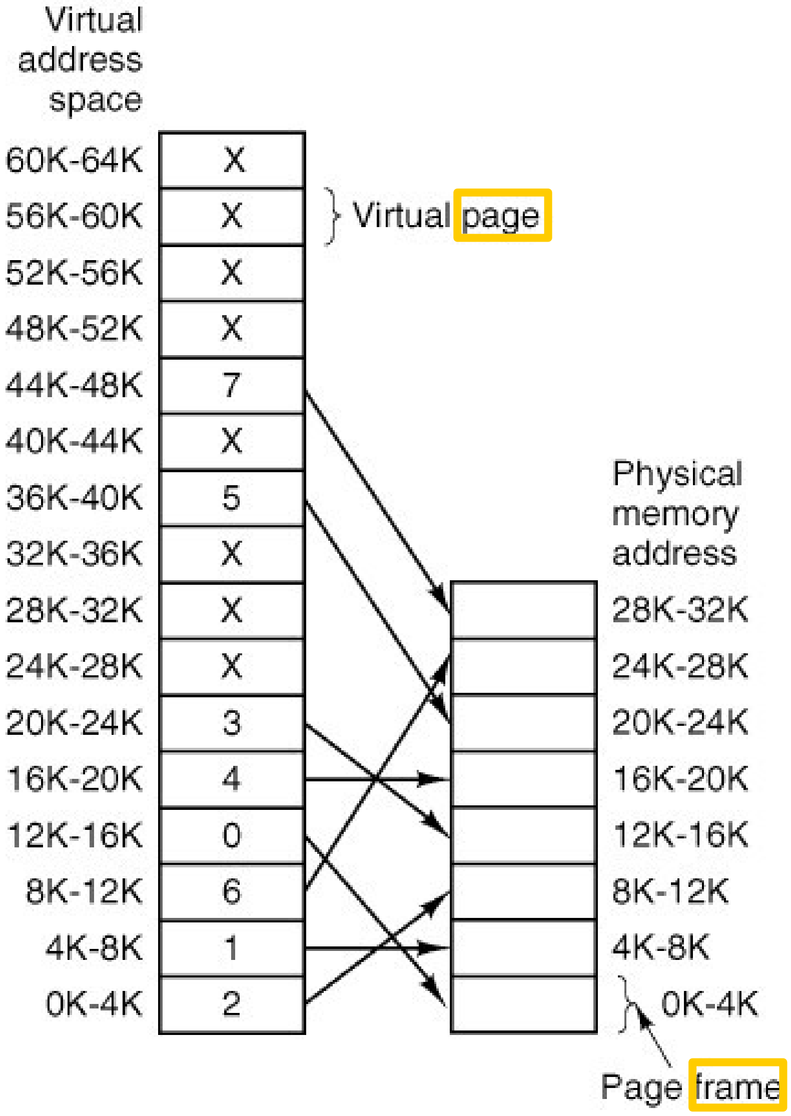
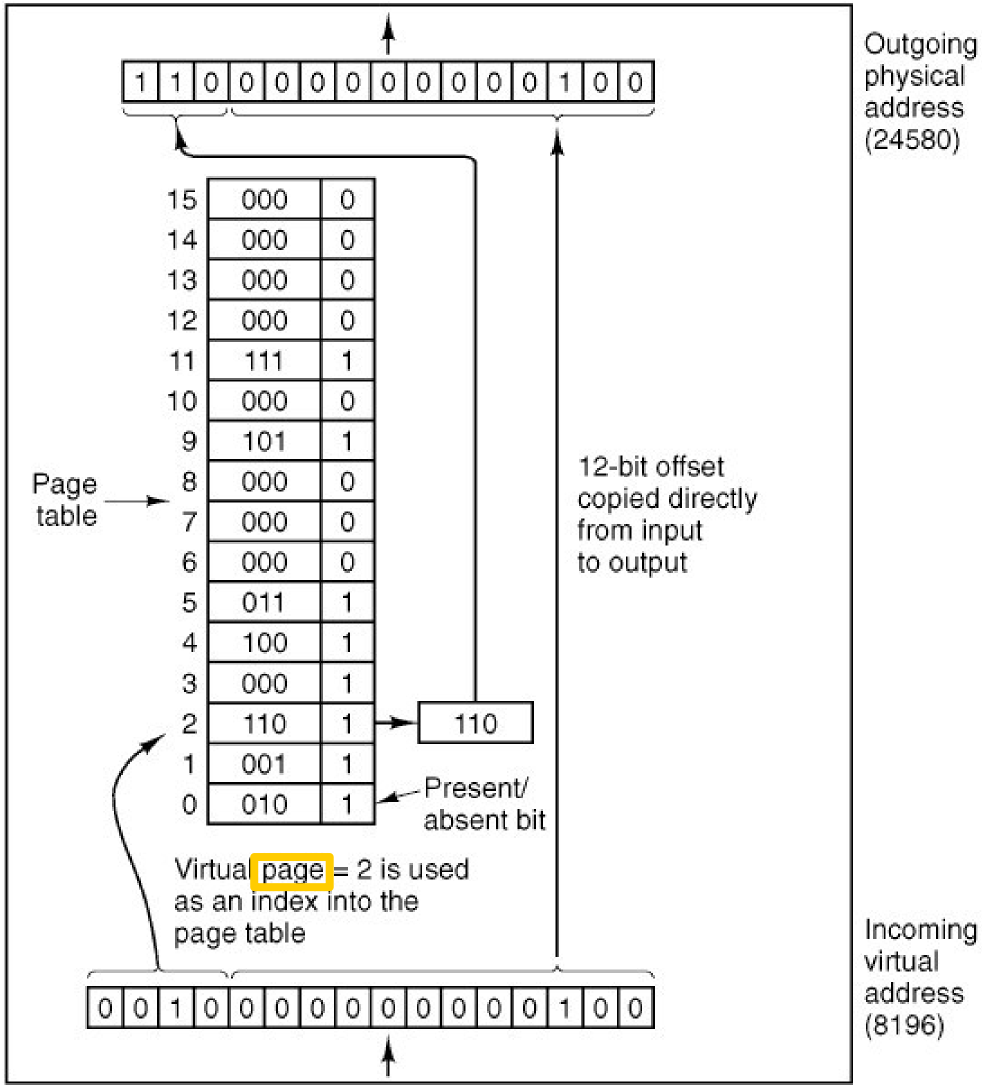


Figure 4-8. The relation between virtual addresses and physical memory addresses is given by the page table.

Tanenbaum & Woodhull, Operating Systems: Design and Implementation, (c) 2006

Paging (3)


1 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0
Outgoing physical address (24580)
12-bit offset copied directly from input to output
Page table
110
Present/ absent bit
Virtual page = 2 is used as an index into the page table
0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0
Incoming virtual address (8196)

15	000	0
14	000	0
13	000	0
12	000	0
11	111	1
10	000	0
9	101	1
8	000	0
7	000	0
6	000	0
5	011	1
4	100	1
3	000	1
2	110	1
1	001	1
0	010	1

Figure 4-9. The internal operation of the MMU with 16 4-KB pages.

Page table: Translation (process#, page#) → frame#

Tanenbaum & Woodhull, Operating Systems: Design and Implementation, (c) 2006

Page Tables

- Purpose: map virtual pages onto page frames

- Major issues to be faced

1. The page table can be extremely large
2. The mapping must be fast.

How to shrink?
How to make fast?
(Data structures involved?)

Multilevel Page Tables

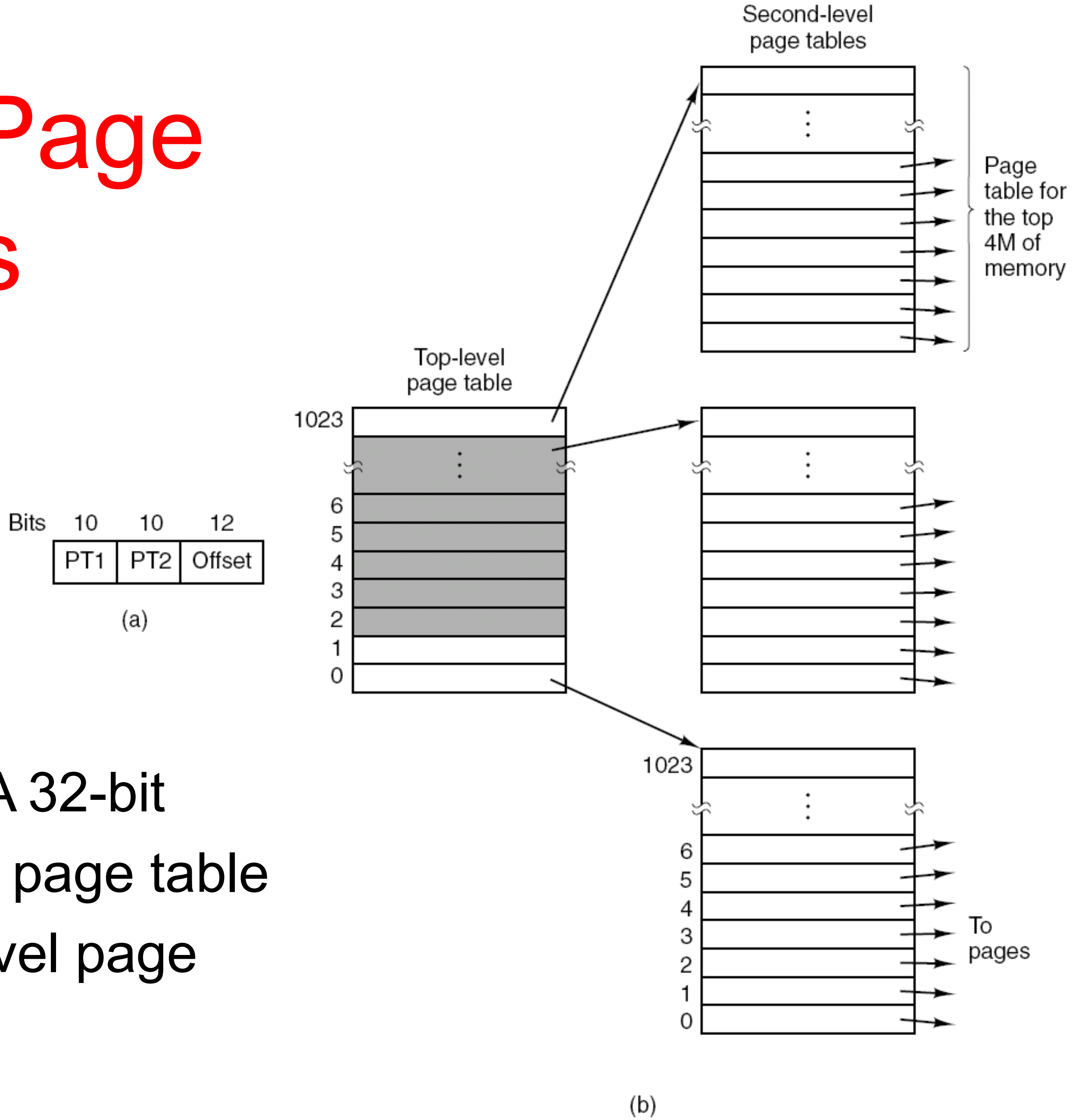
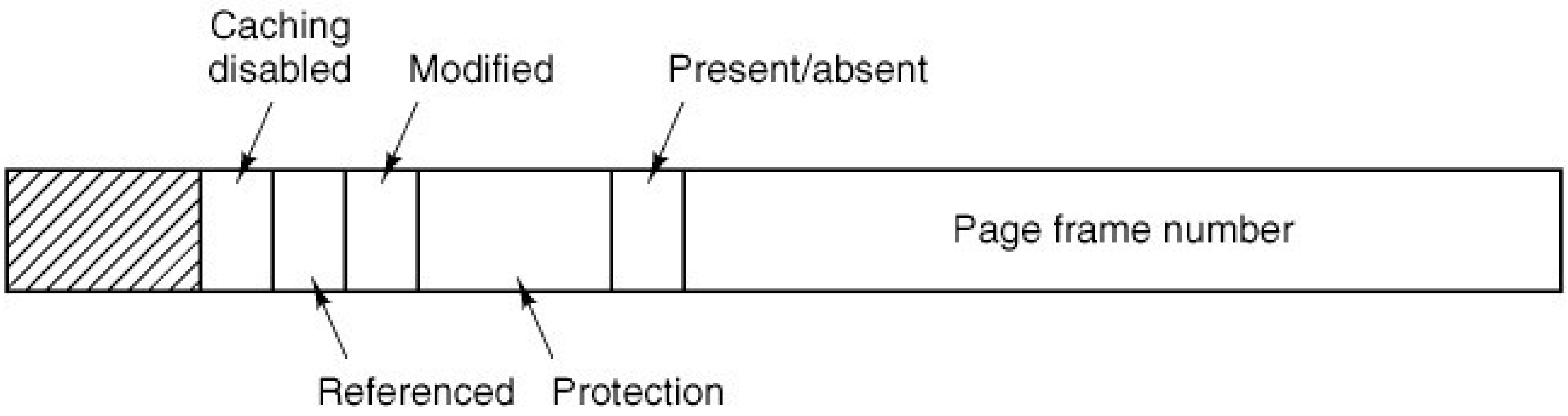


Figure 4-10. (a) A 32-bit address with two page table fields. (b) Two-level page tables.

Structure of a Page Table Entry

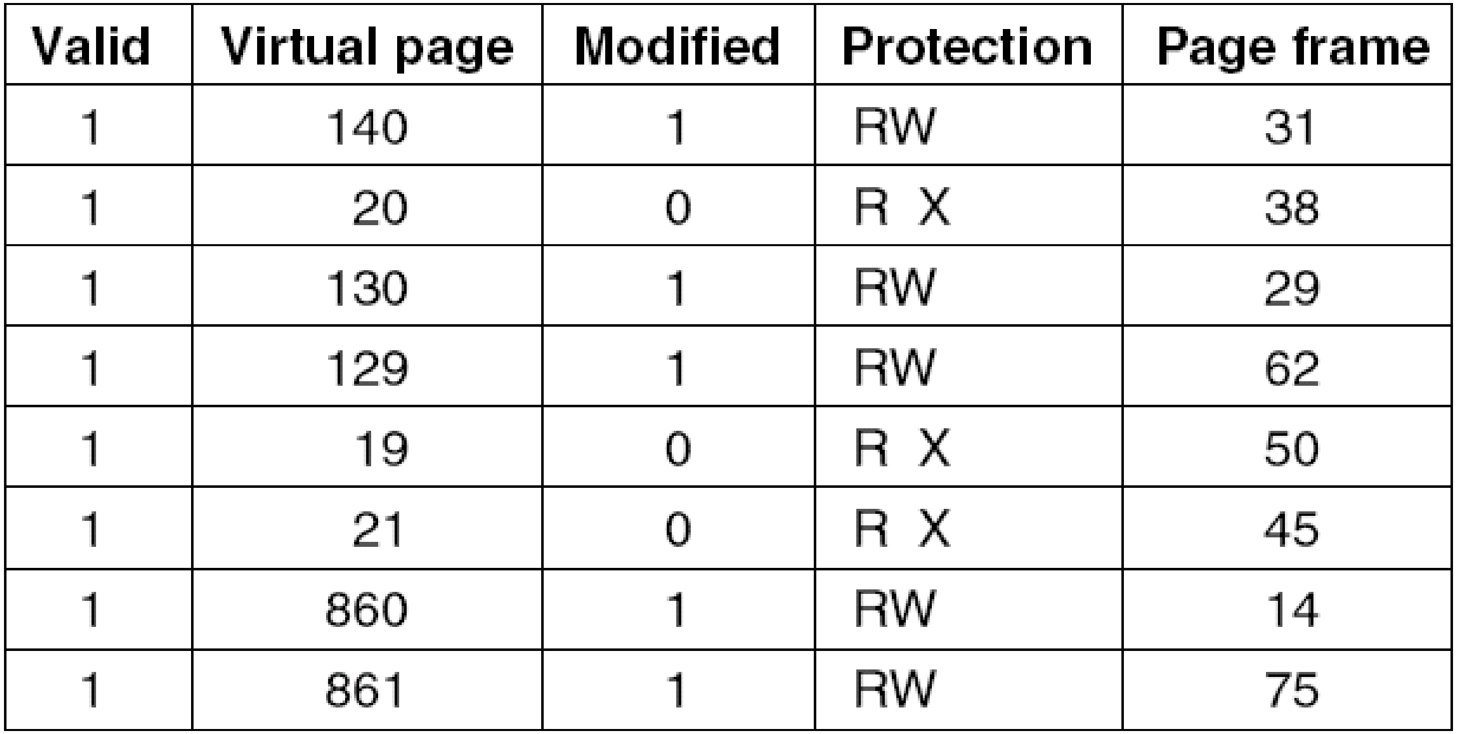


RW(X) for whom?

□Figure 4-11. A typical page table entry.

Tanenbaum & Woodhull, Operating Systems: Design and Implementation, (c) 2006

TLBs—Translation Lookaside Buffers



Valid	Virtual page	Modified	Protection	Page frame
1	140	1	RW	31
1	20	0	R X	38
1	130	1	RW	29
1	129	1	RW	62
1	19	0	R X	50
1	21	0	R X	45
1	860	1	RW	14
1	861	1	RW	75

Figure 4-12. A TLB to speed up paging.

Inverted Page Tables

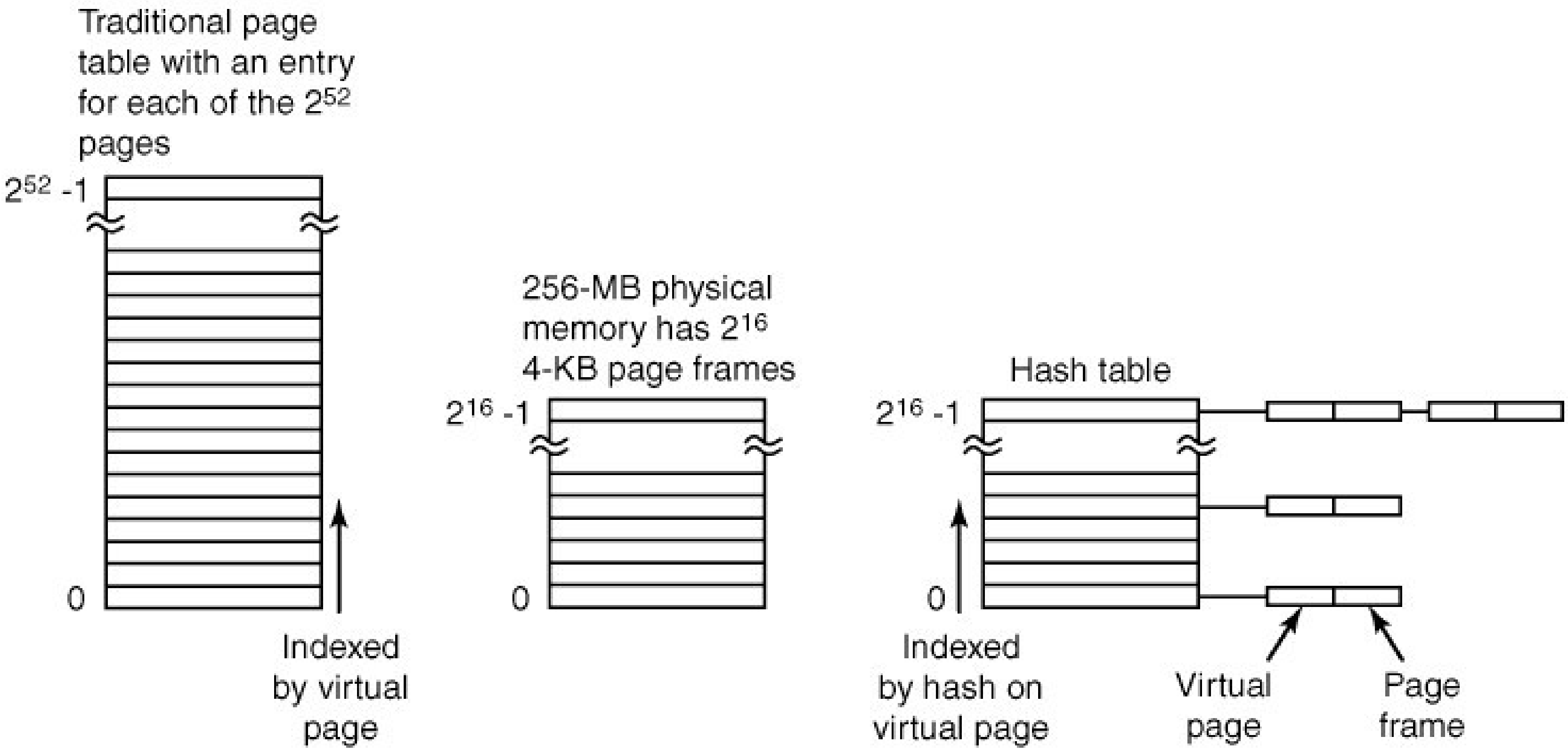

Traditional page table with an entry for each of the 2^{52} pages
2^{52} -1
0
Indexed by virtual page
256-MB physical memory has 2^{16} 4-KB page frames
2^{16} -1
0
Hash table
2^{16} -1
0
Indexed by hash on virtual page
Virtual page
Page frame

Figure 4-13. Comparison of a traditional page table with an inverted page table.

Tanenbaum & Woodhull, Operating Systems: Design and Implementation, (c) 2006