CSCI315 – Operating Systems Design

Department of Computer Science
Bucknell University

Paging: Other Issues and Examples

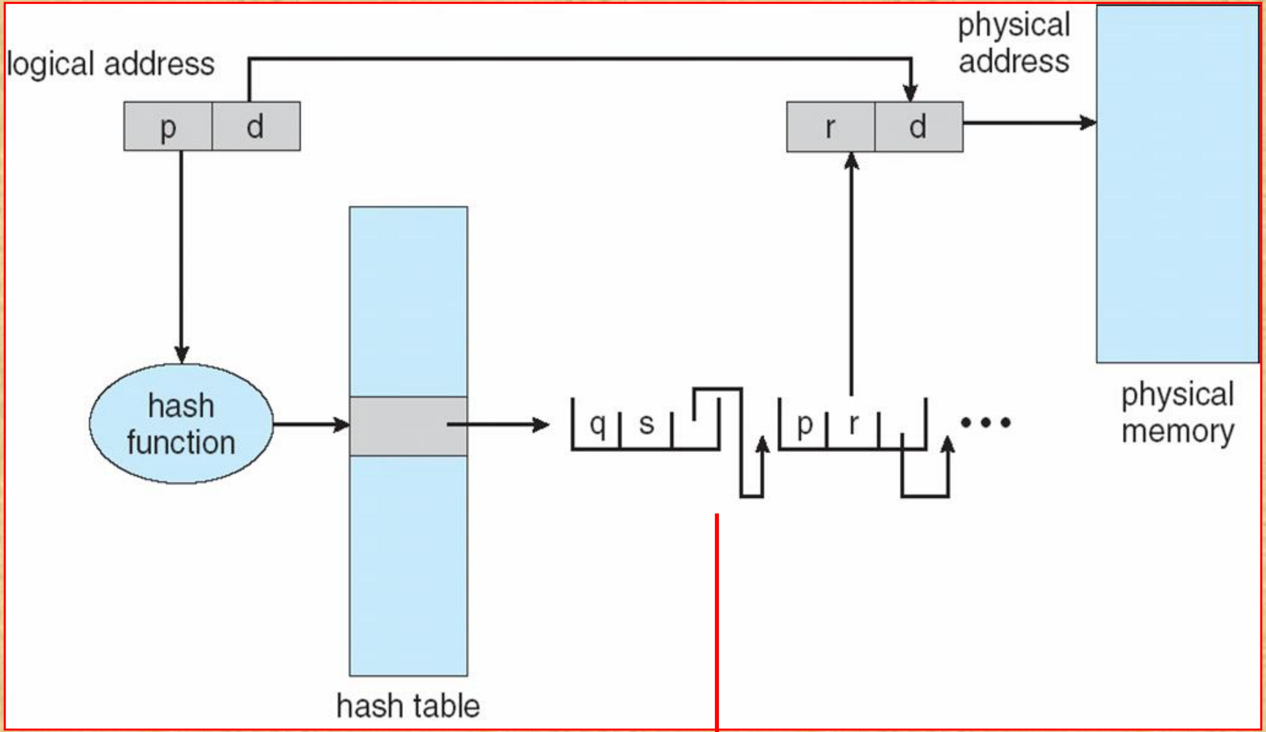
Ch 9.5-9.7

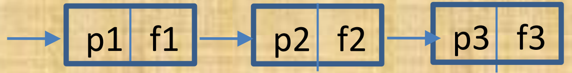
This set of notes is based on notes from the textbook authors, as well as L. Felipe Perrone, Joshua Stough, and other instructors.
Xiannong Meng, Fall 2021.

Hashed Page Tables

- Common in address spaces > 32 bits
- The virtual page number is hashed into a page table
  - This page table contains a chain of elements hashing to the same location
- Each element contains (1) the virtual page number (2) the value of the mapped page frame (3) a pointer to the next element
- Virtual page numbers are compared in this chain searching for a match
  - If a match is found, the corresponding physical frame is extracted

Hashed Page Table





Linked list of page/frame pair

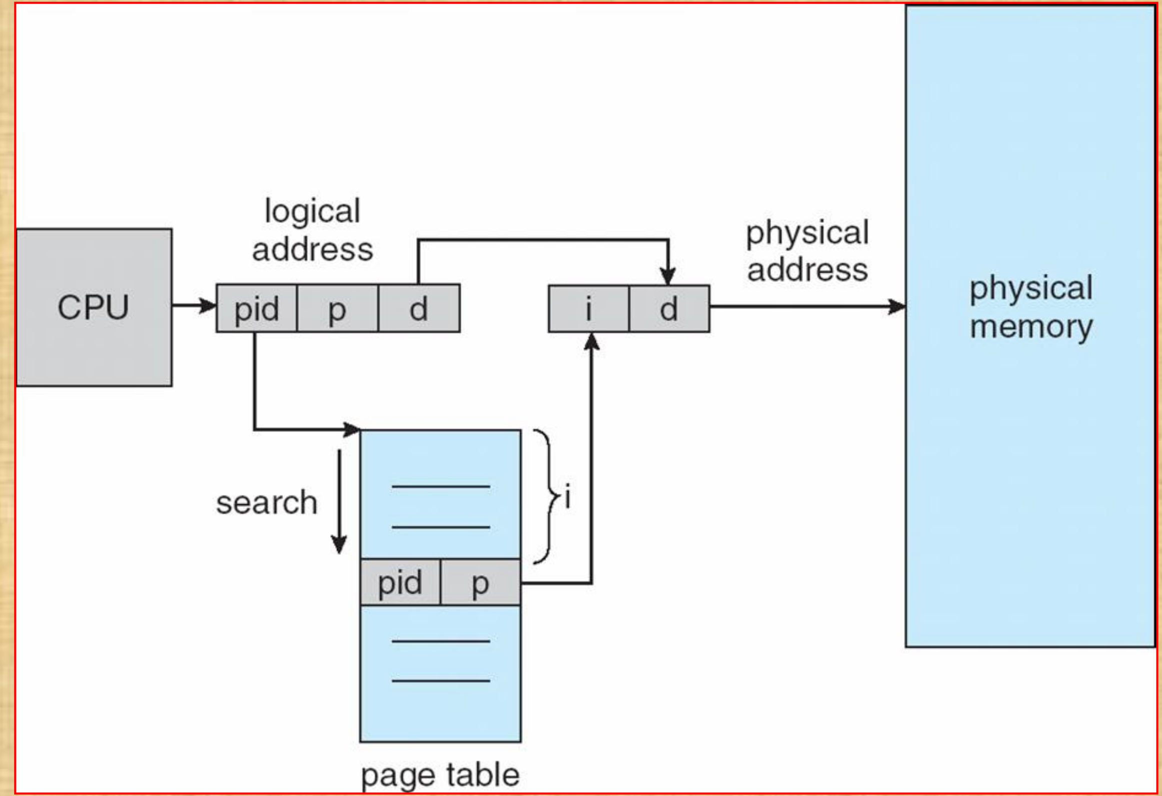
Clustered Page Tables

- Variation for 64-bit addresses is **clustered page tables**
  - Similar to hashed but each entry refers to several pages (such as 16) rather than 1
  - Especially useful for **sparse** address spaces (where memory references are non-contiguous and scattered)

Inverted Page Table

- Rather than each process having a page table and keeping track of all possible logical pages, track all physical pages (frames)
- One entry for each physical page (frame) of memory
- Entry consists of the virtual address of the page stored in that real memory location, with information about the process that owns that page
- Decreases memory needed to store each page table, but increases time needed to search the table when a page reference occurs
- Use hash table to limit the search to one — or at most a few — page-table entries
  - TLB can accelerate access
- But how to implement shared memory?
  - One mapping of a virtual address to the shared physical address

Inverted Page Table Architecture



Example: The Intel 32 and 64-bit Architectures

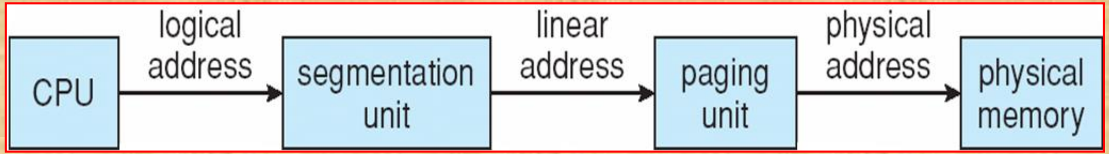
- Dominant industry chips
- Pentium CPUs are 32-bit and called IA-32 architecture
- Current Intel CPUs are 64-bit and called IA-64 architecture
- Many variations in the chips, cover the main ideas here
- The IA-32 defines a 48-bit segment address format, a 16-bit segment number and a 32-bit offset within the segment. The segmented addresses are mapped to 32-bit addresses.

https://en.wikipedia.org/wiki/IA-32

https://en.wikipedia.org/wiki/X86_memory_segmentation

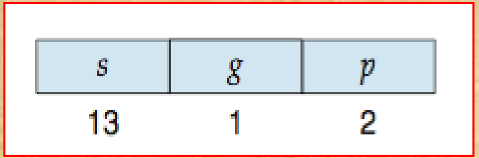
Example: The Intel IA-32 Architecture

- Supports both segmentation and segmentation with paging
  - Each segment can be 4 GB (max for 32 bits)
  - Up to 16 K segments per process
  - Divided into two partitions
    - First partition of up to 8 K segments are private to process (kept in **local descriptor table** (**LDT**))
    - Second partition of up to 8K segments shared among all processes (kept in **global descriptor table** (**GDT**))



Example: The Intel IA-32 Architecture (Cont.)

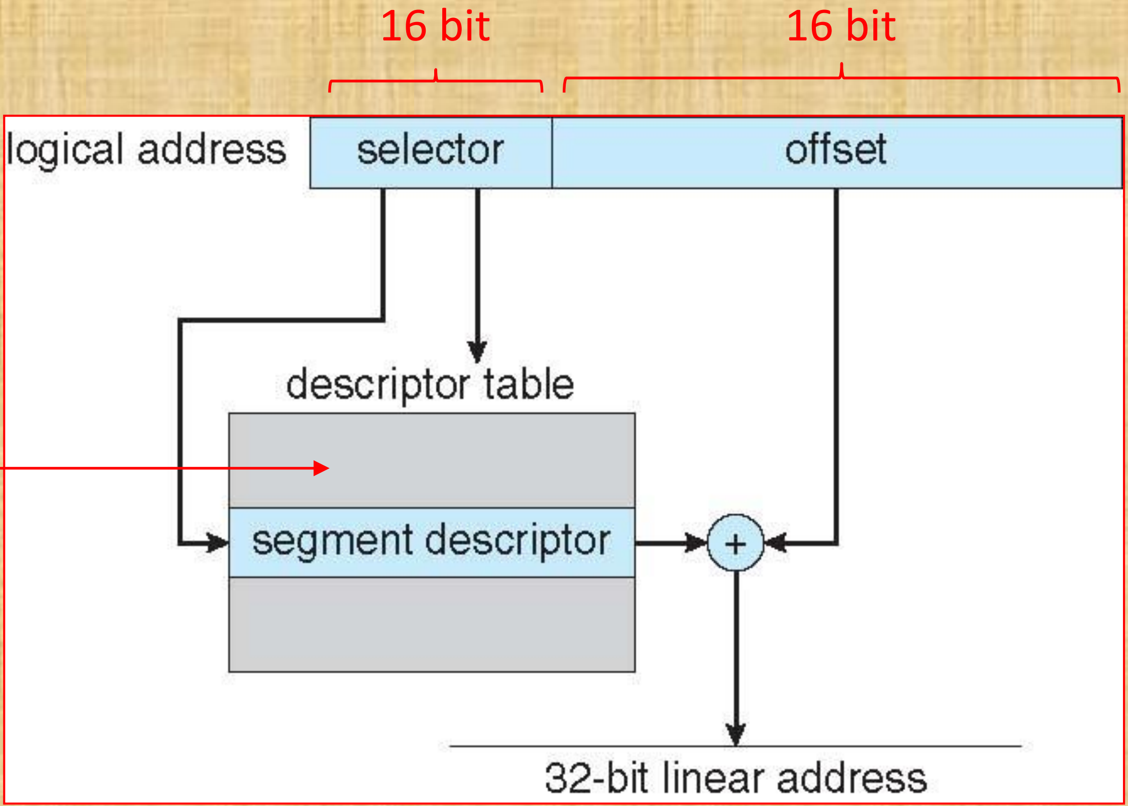
- CPU generates logical address, a **selector** and an **offset**.
  - Selector given to segmentation unit
    - Which produces linear addresses



16-bit segment number

- s: segment number
- g: whether the segment is in the GDT (Global Descriptor Table) or LDT (Local Descriptor Table)
- p: protection information

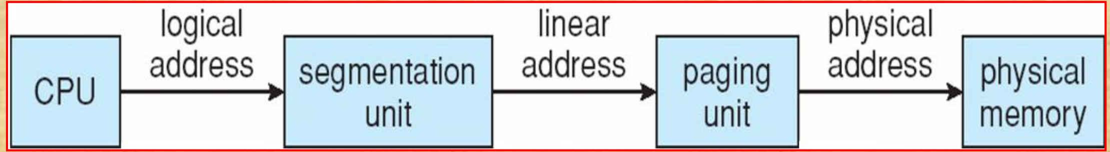
Intel IA-32 Segmentation



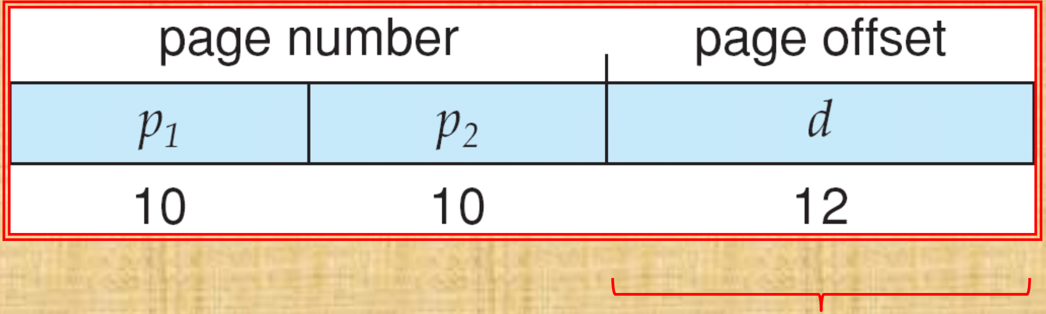
Global or local descriptor table

Fed to paging system

Logical to Physical Address Translation in IA-32



IA-32 allows a page size either 4 KB or 4 MB. For a 4 KB page, 2-level paging is used. Example shown on the right.

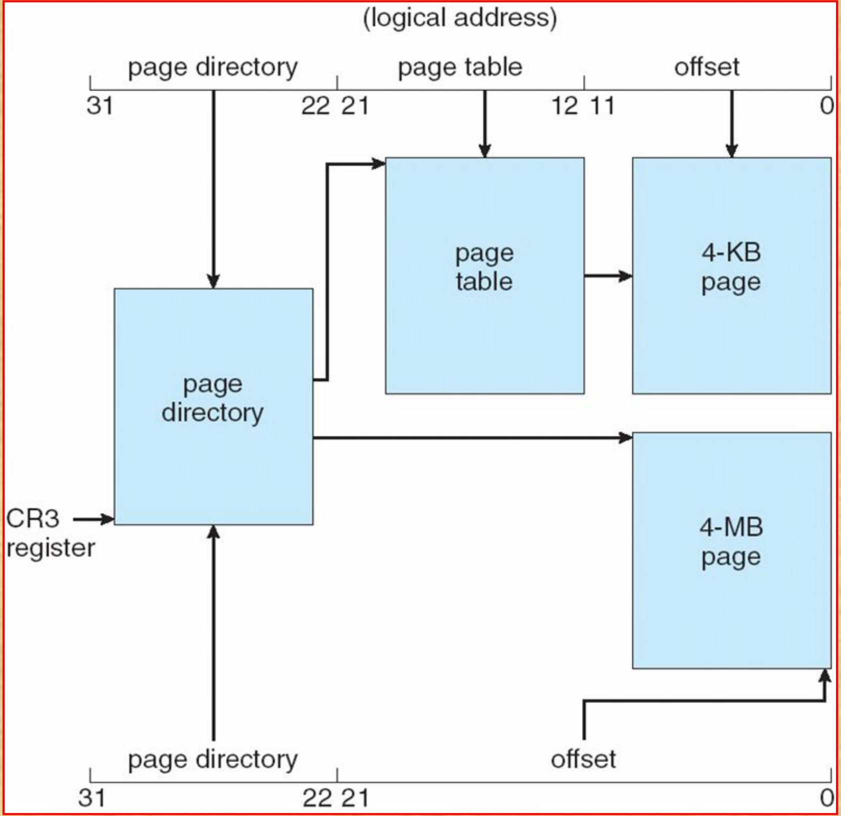


12 bit for 4 KB

Intel IA-32 Paging Architecture

Two-level paging system in IA-32.

- 10-bit page directory
- 10-bit page table
- 12-bit offset for 4K page

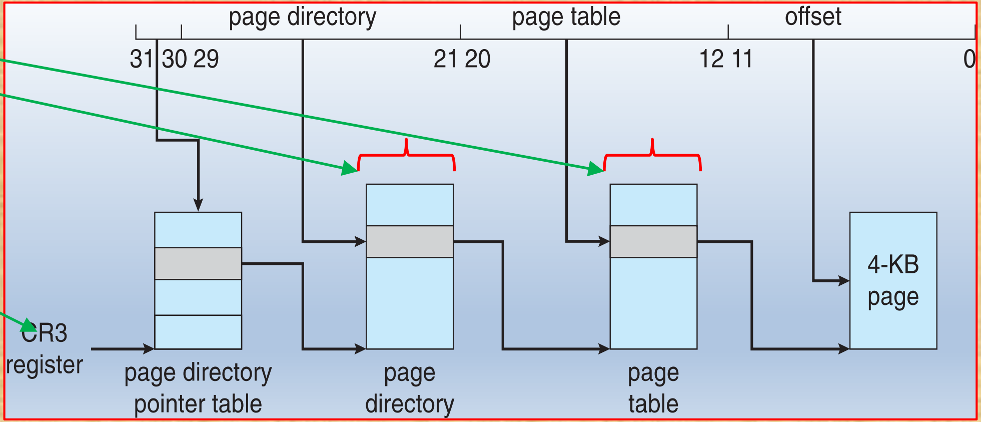


Intel IA-32 Page Address Extensions

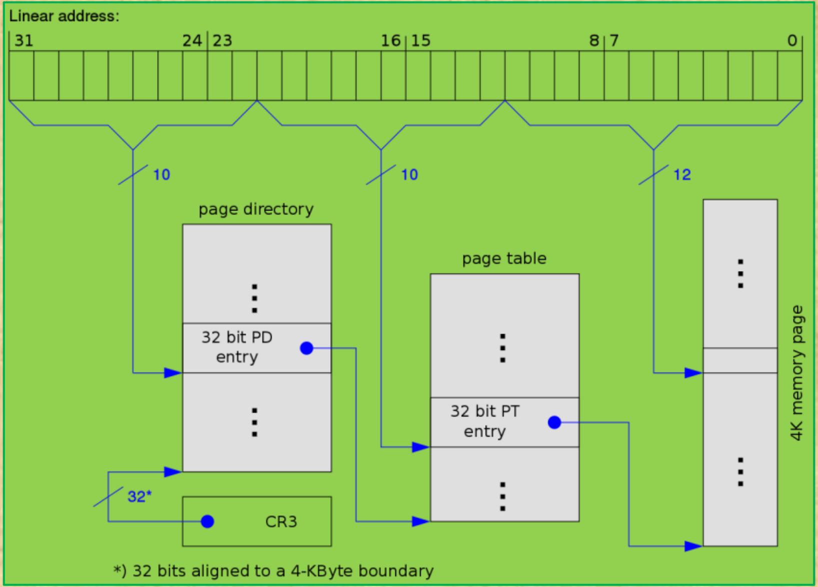
- 32-bit address limits led Intel to create **page address extension** (**PAE**), allowing 32-bit apps access to more than 4GB of physical memory space
  - Paging went to a 3-level scheme
  - Top two bits refer to a **page directory pointer table**
  - Page-directory and page-table entries moved to 64-bits in size
  - Net effect is increasing address space to 36 bits – 64GB of physical memory

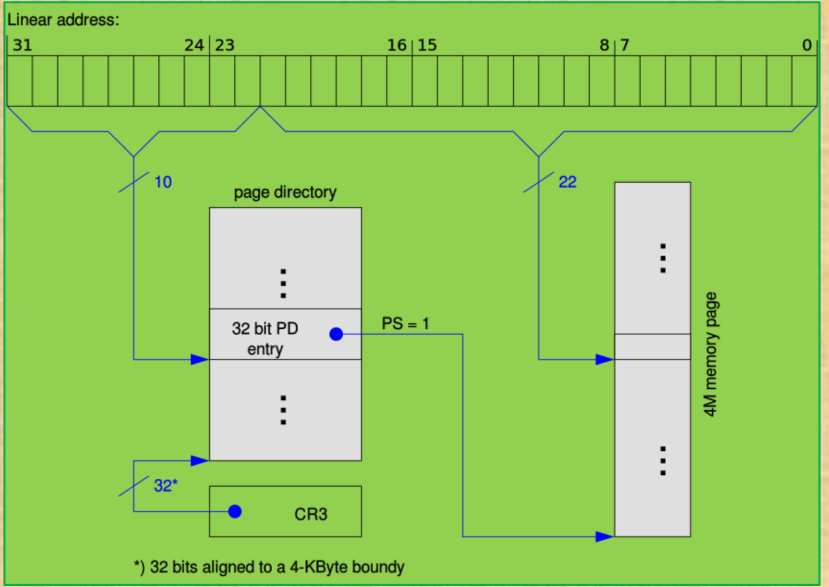
Could extend to 64 bits wide

The CR3 register holds the base address of a page table entry



Intel IA-32 Extension (PAE)



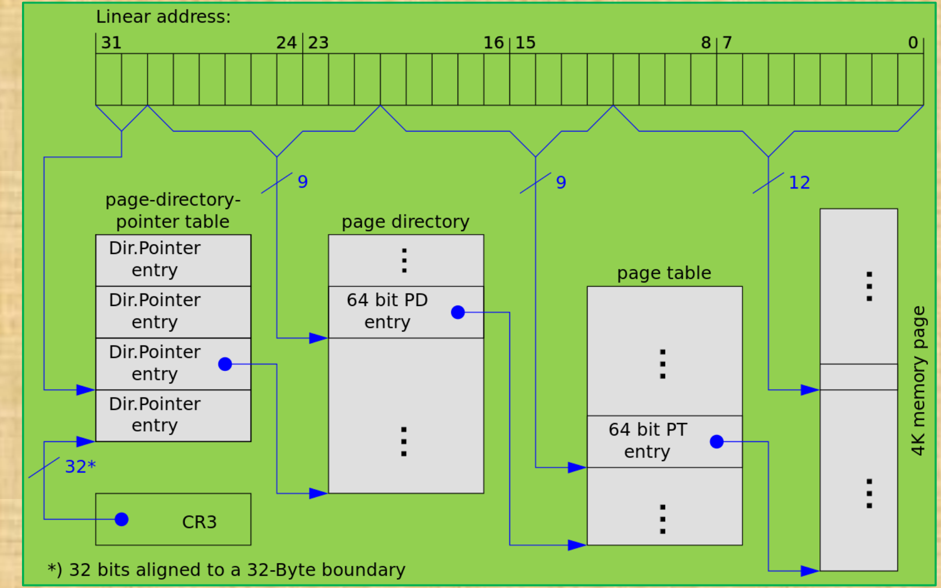


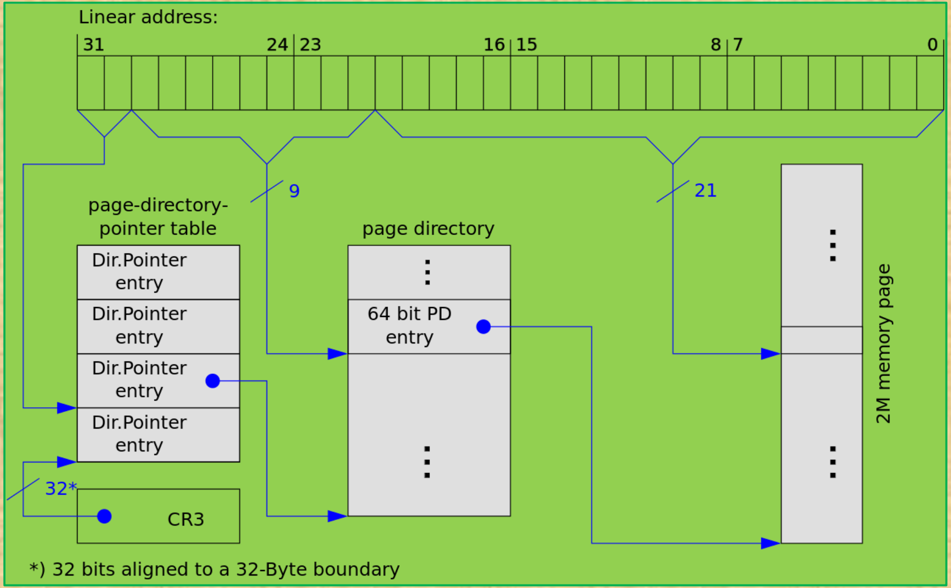
No PAE: 4 KB pages

No PAE: 2 MB pages

https://en.wikipedia.org/wiki/Physical_Address_Extension

Intel IA-32 Extension (PAE)





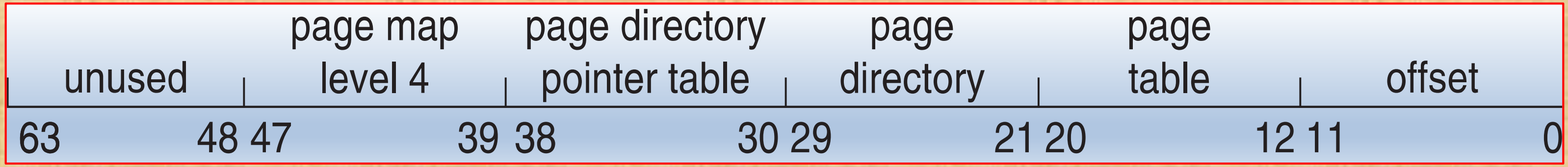
With PAE: 4 KB pages

With PAE: 2 MB pages

https://en.wikipedia.org/wiki/Physical_Address_Extension

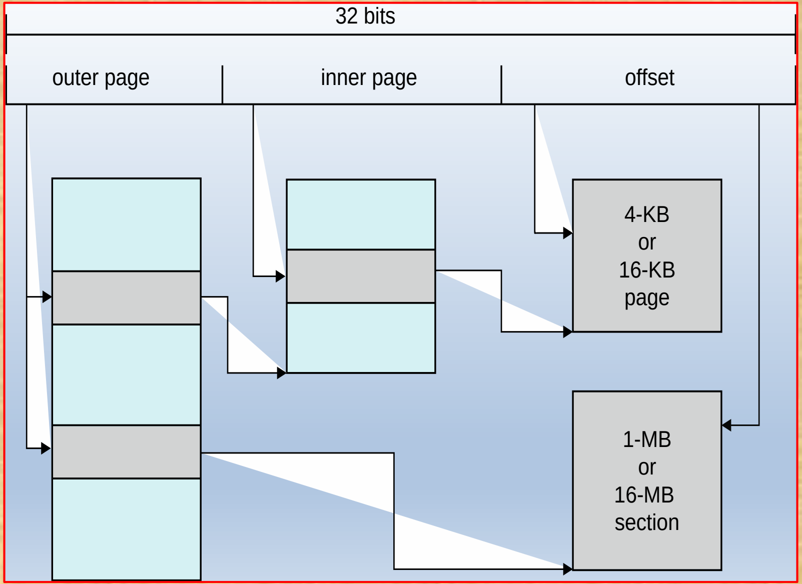
Intel x86-64

- Current generation Intel x86 architecture
- 64 bits is ginormous (> 16 exabytes)
- In practice only implement 48 bit addressing
  - Page sizes of 4 KB, 2 MB, 1 GB
  - Four levels of paging hierarchy
- Can also use PAE so virtual addresses are 48 bits and physical addresses are 52 bits



Example: ARM Architecture

- Dominant mobile platform chip (Apple iOS and Google Android devices for example)
- Modern, energy efficient, 32-bit CPU
- 4 KB and 16 KB pages
- 1 MB and 16 MB pages (termed **sections**)
- One-level paging for sections, two-level for smaller pages
- Two levels of TLBs
  - Outer level has two micro TLBs (one data, one instruction)
  - Inner is single main TLB
  - First inner is checked, on miss outers are checked, and on miss page table walk performed by CPU


32 bits
outer page
inner page
offset
4-KB
or
16-KB
page
1-MB
or
16-MB
section