

# 32 bits address space, 4K pages, 4 bytes/PTE

- How many bits in offset? 4K  
so the virtual address requires **12 bits for the offset**
  - We want a master page table to fit in one page  
 $4K/4 \text{ bytes} = 1K$  possible entries  
so the virtual address requires **10 bits for the master page index**
  - We also want a secondary page table to fit in one page  
so the virtual address requires **10 bits for the secondary page index**
- ➔  $10 + 10 + 12 = 32$  bits address  
This is why 4K page size is recommended

# x86 Paging

- Paging enabled by bits in a control register `%cr0`  
(only privileged OS code can manipulate control registers)
- Register `%cr3` points to 4KB page directory  
(for Pintos, see `pagedir_activate()` in `userprog/pagedir.c`)
- Page directory has 1024 PDEs (Page Directory Entries) (see pagination details)
  - Each contains physical address of a page table
  - Each page table has 1024 PTEs (page table entries) and covers 4 MB of virtual memory
  - Each contains physical address of virtual 4K page