MEMORY PAGING

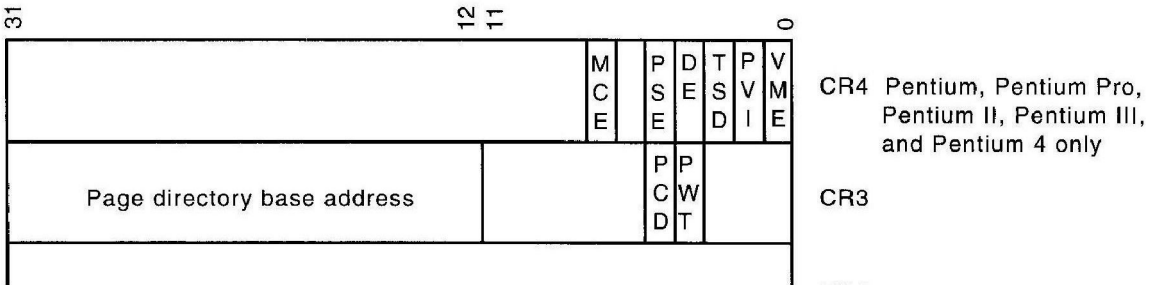
Paging is enabled when the PG bit in control register CR0 is set. The paging mechanism can function in both the real and protected modes.

When paging is enabled, physical memory is divided into small blocks (typically 4K bytes or 4M bytes) in size, and each block is assigned a page number. The operating system keeps a list of free pages in its memory. When a program makes a request for memory, the OS allocates a number of pages to the program.

A key advantage to memory paging is that memory allocated to a program does not have to be **contiguous**, and because of that, there is very little internal fragmentation - thus **little memory is wasted**.

THE PAGE DIRECTORY AND PAGE TABLE

To convert a 32-bit linear address into a 32-bit physical address, we need to understand that the most significant **20 bits** of the linear address indicate the **linear page number**, while the least significant **12 bits** of the linear address indicate the **offset within this page**. The offset should remain the same but the linear page number has to be converted into a physical page number.



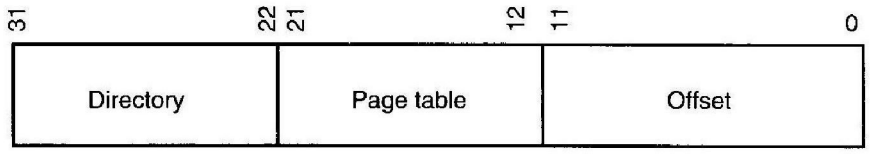
Each page directory entry is a physical address pointing to a **page table**, which contains **page table entries**. Each page table contains **1024** page table entries, each of which is 4 bytes (32 bits). This means that each page table is **4 K bytes** long.

Each page table entry points to the starting physical address of a page in memory (i.e., the **physical page number**).

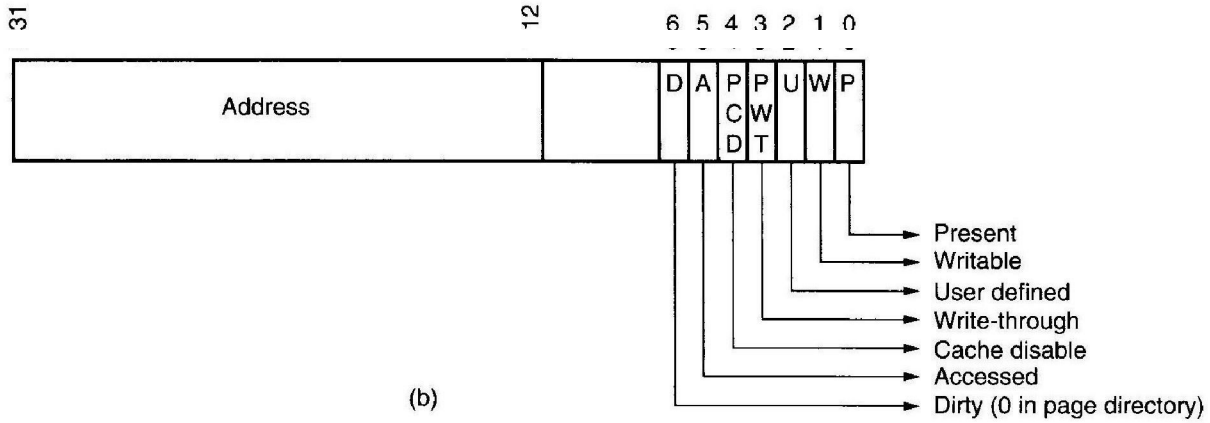
This means that if we have **one** page directory and **1024** page tables, then we have a total of 1M table entries or 1 M pages. Since each page is 4K bytes long, this will cover a total of 4G bytes of maximum physical memory.

The figure below Part (a) shows the linear address (generated by the software) and how it selects one of the 1024 page directory entries from the page directory (using the left most 10 bits) and then selects one of the 1024 page table entries (using the next 10 bits). Part (b) of the figure shows the page table entry, which contains the physical number that must be associated with the offset.

For example, the linear addresses 00000000h-00000FFFh access the first page directory entry, and the first page table entry. Notice that one page is a 4K-byte address range. So, if that page table entry contains 00100000h, then the physical address of this page is 00100000h-00100FFFh for linear address 00000000h-00000FFFh. This means that when the program accesses a location between 00100000h and 00100FFFh, the microprocessor physically addresses location 00100000h-00100FFFh.



(a)



(b)

For example, the linear addresses 00000000h-00000FFFh access the first page directory entry, and the first page table entry. Notice that one page is a 4K-byte address range. So, if that page table entry contains 00100000h, then the physical address of this page is 00100000h-00100FFFh for linear address 00000000h-00000FFFh. This means that when the program accesses a location between 00100000h and 00100FFFh, the microprocessor physically addresses location 00100000h-00100FFFh.

The procedure for converting linear addresses into physical addresses: