Memory Management : Paging

* 1 KB = 1024 Bytes = 2^{10} Bytes

  1 MB = 1024 KB = 2^{20} Bytes

  1 GB = 1024 MB = 2^{30} Bytes

* 32 bit Virtual Memory = 2^{32} Bytes = $2^2 \times 2^{30}$ Bytes
  = 4 GB

* 64 MB RAM = $\log_2(64 \times 2^{20})$ = 26 bits

* Size of virtual memory, physical memory, page are typically given.

* Number of virtual pages = $\frac{\text{Virtual Memory Size}}{\text{Page size}}$

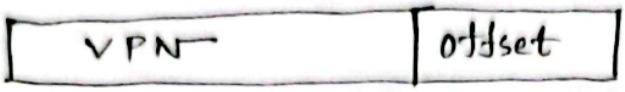
* Page size in virtual memory & physical memory is same.

* Page table keeps track of Virtual Page Number (VPN) to Physical Frame Number (PFN) mapping.

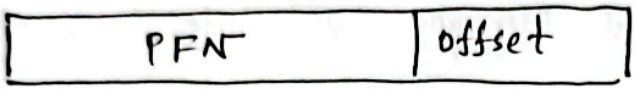
* Offset means number of bits required to address each byte location in a page.

  offset = $\log_2$ (page size in Bytes)

* virtual address has two parts:



* physical address has two parts:



*** offset is not changed while translating virtual address to physical address.

* To keep track of which VPN is mapped to which PFN, operating system creates a page table for each process.

* Number of PTE (Page Table Entries) = Number of Virtual pages.

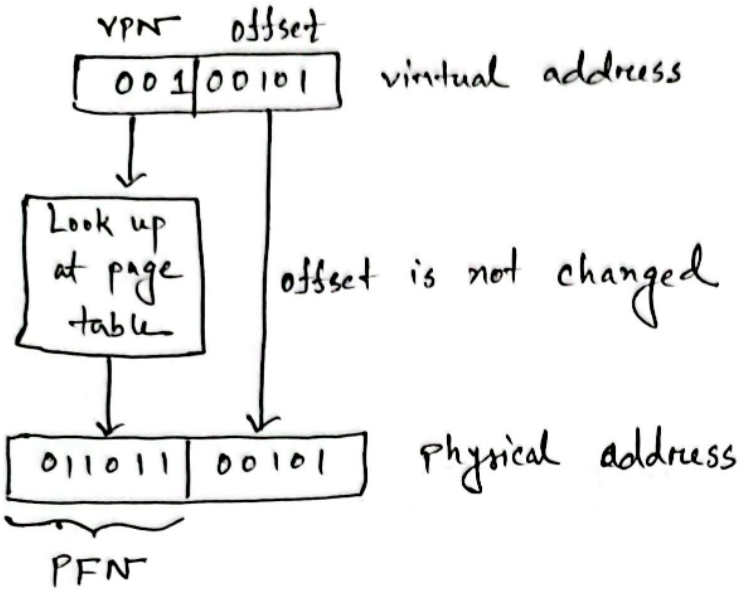
* Typically PTE size is given.

* Each PTE contains corresponding PFN and some flag bits(valid bit, write protection bit, dirty bit etc.)

* valid bit = 1 means that virtual page is used by the running process.

valid bit = 0 means that virtual page is not used.

A typical Address Translation



* Single page table requires contiguous space and this size can be larger. To solve this problem, we chop (divide) the page table into page size segments and these can be allocated in different Page Frames (Physical pages). [This concept is discussed in more details in the class].

* To keep track of those segments, another level of page table is created. Here comes the concept of Multi-level page table.

* Number of bits required for each level of page table = $\log_2\left(\frac{\text{Page size}}{\text{PTE size}}\right)$

Here, $\frac{\text{Page size}}{\text{PTE size}}$ means number of PTEs that can be placed in a single page.