Page Table structure

Page Table:

A page table is a data structure used by a virtual memory system in a computer system.

It is used to store the mapping between virtual addresses and physical addresses.

Techniques used for structuring the page table are:

① Hierarchical paging
② Harshed page table
③ Inverted page table

1) Hierarchical page Table: (Multilevel paging)

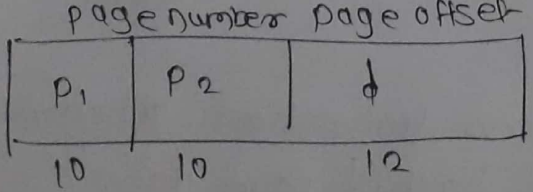
- Most modern computer systems support a large logical address space (2^{32} to 2^{64}).
- In such an environment, the page table itself becomes excessively large.
- To solve this problem many computers use a hierarchical paging technique.
- In this technique, the logical address space is broken into multiple page tables.

eg. Two level page technique. (32 bit logical address)

A page number consisting of 20 bits.
A page offset consisting of 12 bits.

Since the page table is paged, the page number is further divided into

A 10-bit page number
A 10-bit page offset



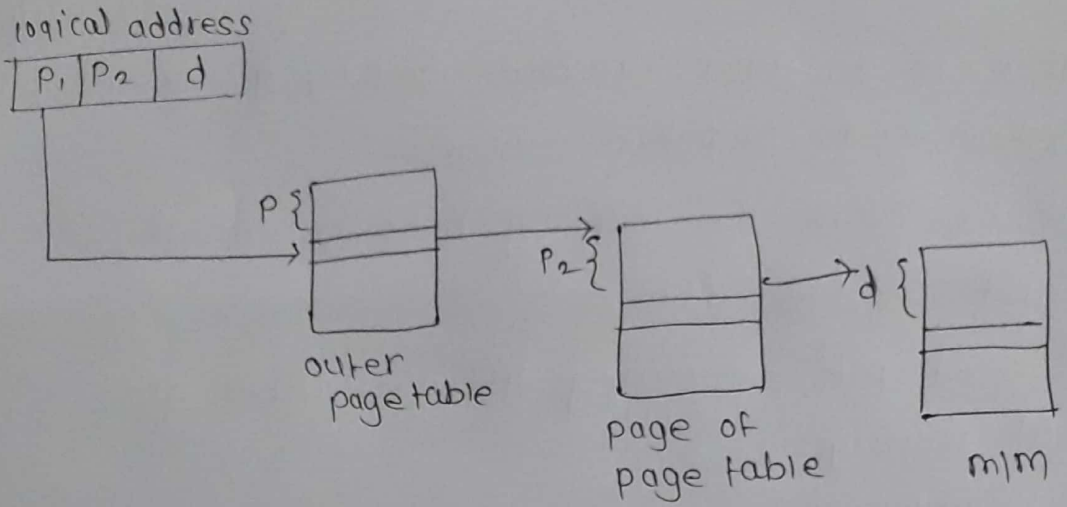


Fig: Address translation scheme for the two level paging.

② Hashed page Tables

- It is a common approach used when address space is greater than 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 consist of 3 fields:
  1. The virtual page number.
  2. The value of the mapped page frame
  3. A pointer to the next element in the linked list

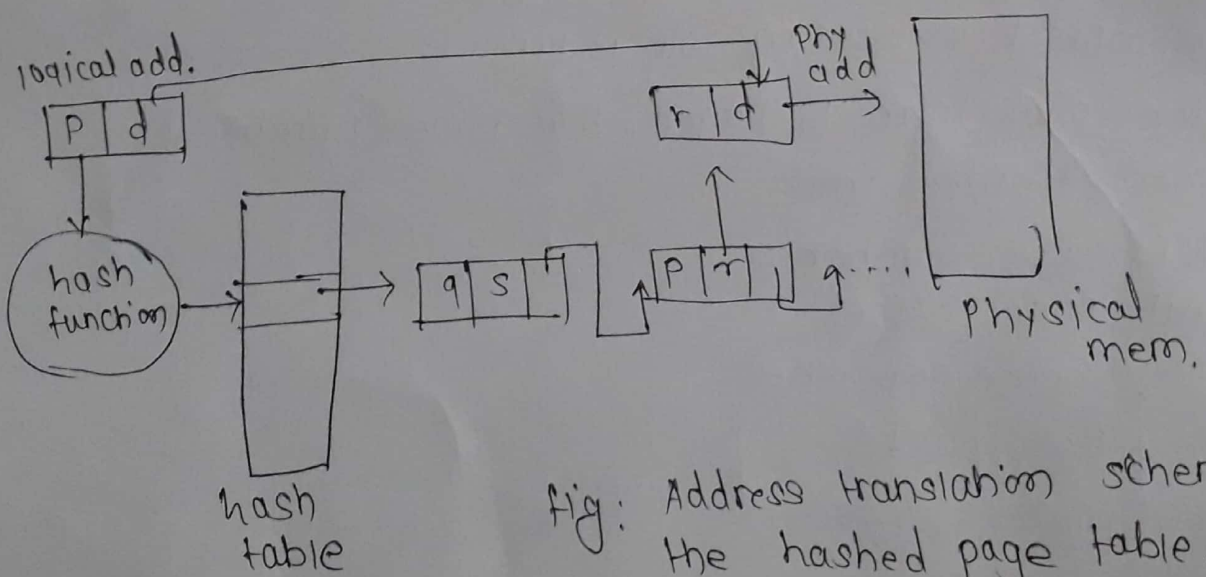


Fig: Address translation scheme for the hashed page table.

(3) Inverted page table:

- Inverted page table contains page table and frame table into one data structure.
- one entry for each virtual page number & real page of memory.
- Entry consist of virtual address of the page stored in that real memory location, with information about the process that owns the page table.
- It decreases the memory needed to store each page table, but increases time needed to search table when a page reference occurs.

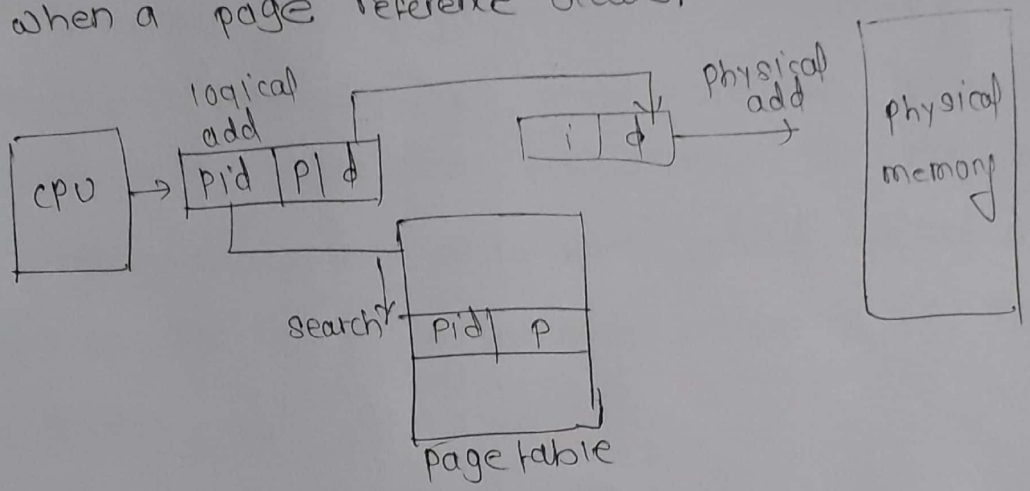


fig: Address translation scheme for the inverted page table