

OPERATING SYSTEM

Inverted Paging (Inverted Page Table)

Why Inverted Paging is Needed

In traditional paging:

- Each **process has its own page table**
- For large systems, this results in **huge memory overhead**

👉 Inverted paging solves this by using a single page table for the entire system.

Definition

Inverted paging is a memory management technique in which **one global page table** is maintained for **all processes**, and the table is indexed by **physical frame number** instead of page number.

Each entry stores **which process and which virtual page** is stored in that frame.

Key Idea (Very Simple Explanation)

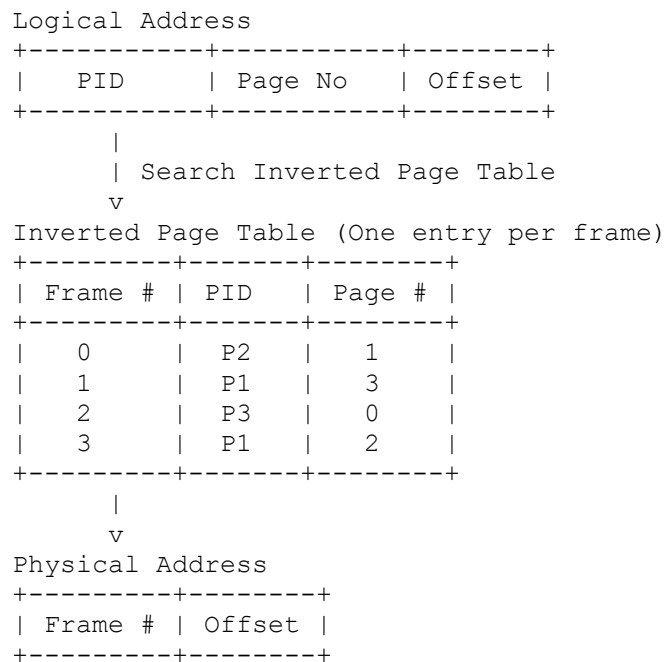
- Traditional page table:
👉 Page → Frame
- Inverted page table:
👉 Frame → (Process ID, Page Number)

Structure of an Inverted Page Table

Each entry contains:

- **Process ID (PID)**
- **Virtual Page Number**
- **Control bits** (valid, protection, etc.)

Inverted Paging Diagram



How Address Translation Works

1. CPU generates a **logical address** (PID, page number, offset)
2. OS searches the **inverted page table**
3. Finds a matching entry with the same **PID and page number**
4. Index of the matching entry gives the **frame number**
5. Frame number + offset → **physical address**

Advantages of Inverted Paging

- Very small page table (one entry per frame)
- Suitable for large address spaces (64-bit systems)
- Reduces memory overhead
- Scales well for many processes

Disadvantages of Inverted Paging

- Slow lookup (table must be searched)
- Requires **hashing** or TLB for speed
- Complex implementation

Role of Hashing & TLB

- **Hashing** is used to speed up page lookup
- **TLB** stores recent translations
- Together they reduce search time significantly

Comparison with Multilevel Paging

Feature	Multilevel Paging	Inverted Paging
Page Table	Per process	One global
Table Size	Large	Small
Lookup Speed	Faster	Slower
Memory Overhead	High	Low

Inverted paging uses a single page table for the entire system, indexed by physical frame number. Each entry stores the process ID and virtual page number. It significantly reduces memory overhead and is suitable for large address spaces.

One-Line Summary

👉 *Inverted paging reverses the page table mapping to save memory by using one global table for all processes.*