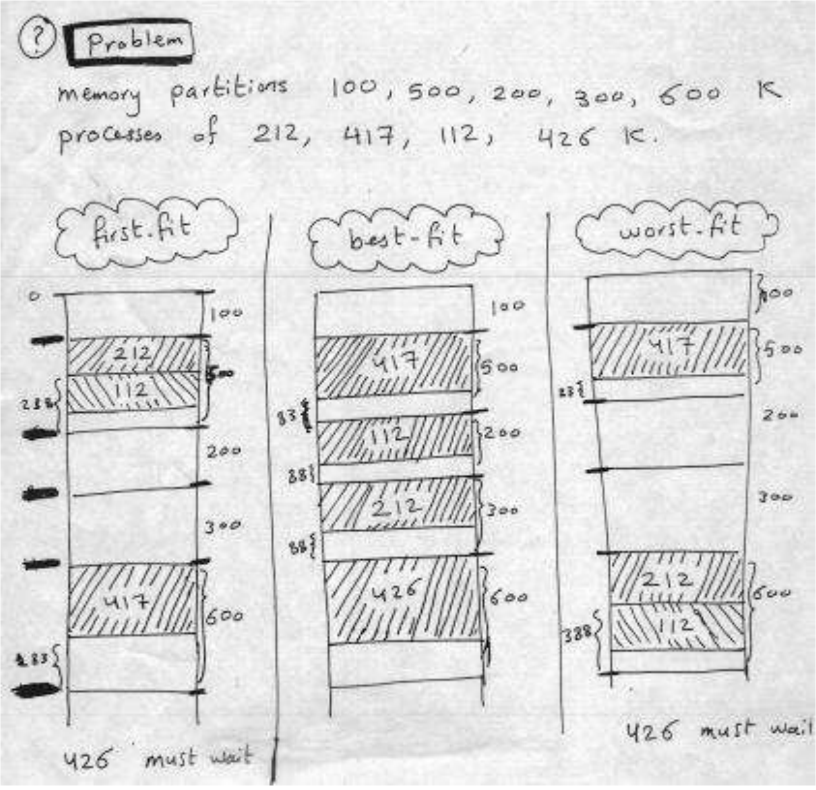
4.5.2 Storage-Allocation

Storage-Allocation

- **First-fit**: Allocate the *first* hole that is big enough.
- **Best-fit**: Allocate the *smallest* hole that is big enough; must search entire list, unless ordered by size. Produces the smallest leftover hole.
- **Worst-fit**: Allocate the *largest* hole; must also search entire list. Produces the largest leftover hole.



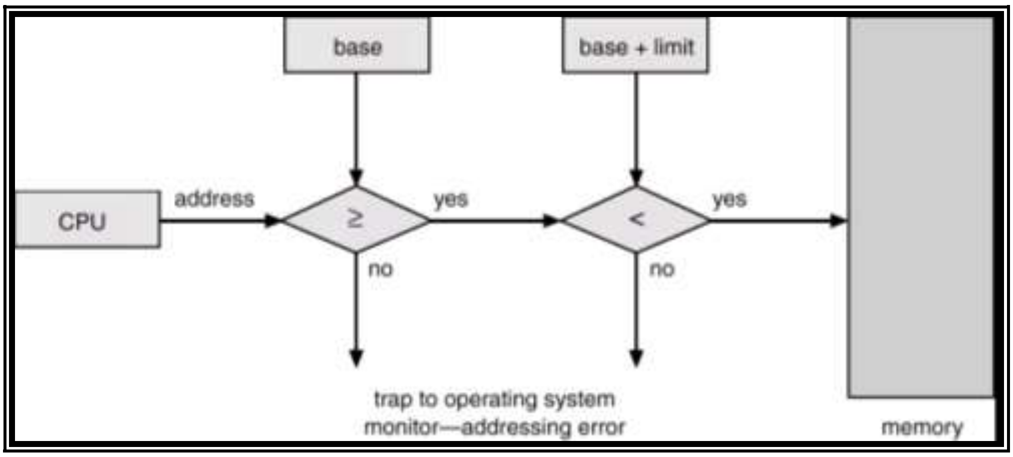
Fragmentation

- **External Fragmentation** – total memory space exists to satisfy a request, but it is not contiguous.
- **Internal Fragmentation** – allocated memory may be slightly larger than requested memory; this size difference is memory internal to a partition, but not being used.

Reduce external fragmentation by compaction

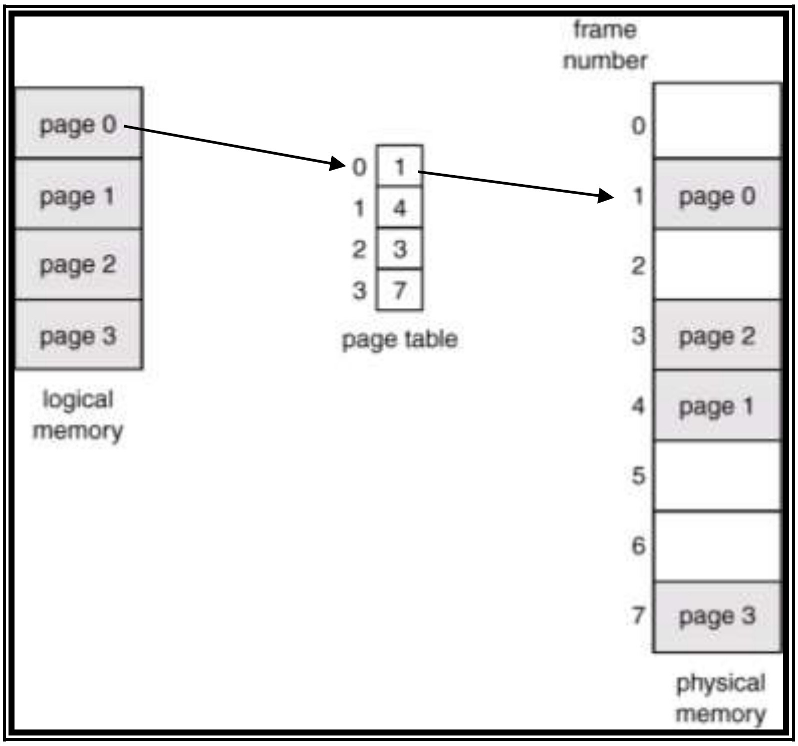
Logical address (virtual address) – generated by the CPU;.

Physical address – address seen by the memory unit.



Paging

- Divide **physical** memory into fixed-sized blocks called **frames** (size is power of 2, between 512 bytes and 8192 bytes).
- Divide **logical** memory into blocks of same size called **pages**.



Consider a **logical** address space of **8 pages** of **1024 words** each, mapped onto a **physical** memory of **32 frames**.

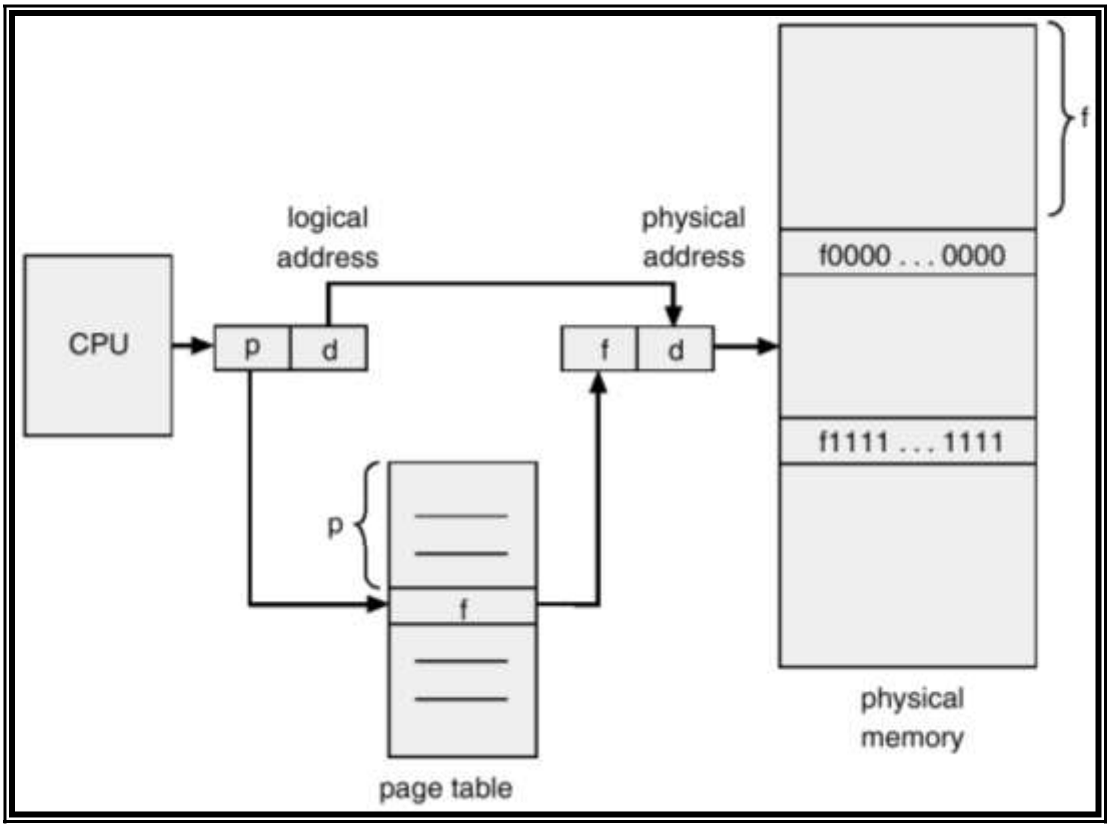
a. How many bits are there in the **logical** address?
b. How many bits are there in the **physical** address?

Answer
Addressing within a 1024-word page requires 10 bits because $1024 = 2^{10}$. Since the logical address space consists of $8 = 2^3$ pages, the logical addresses must be 10+3 = 13 bits. Since there are $32 = 2^5$ physical pages, phyiscal addresses are 5 + 10 = 15 bits.

---

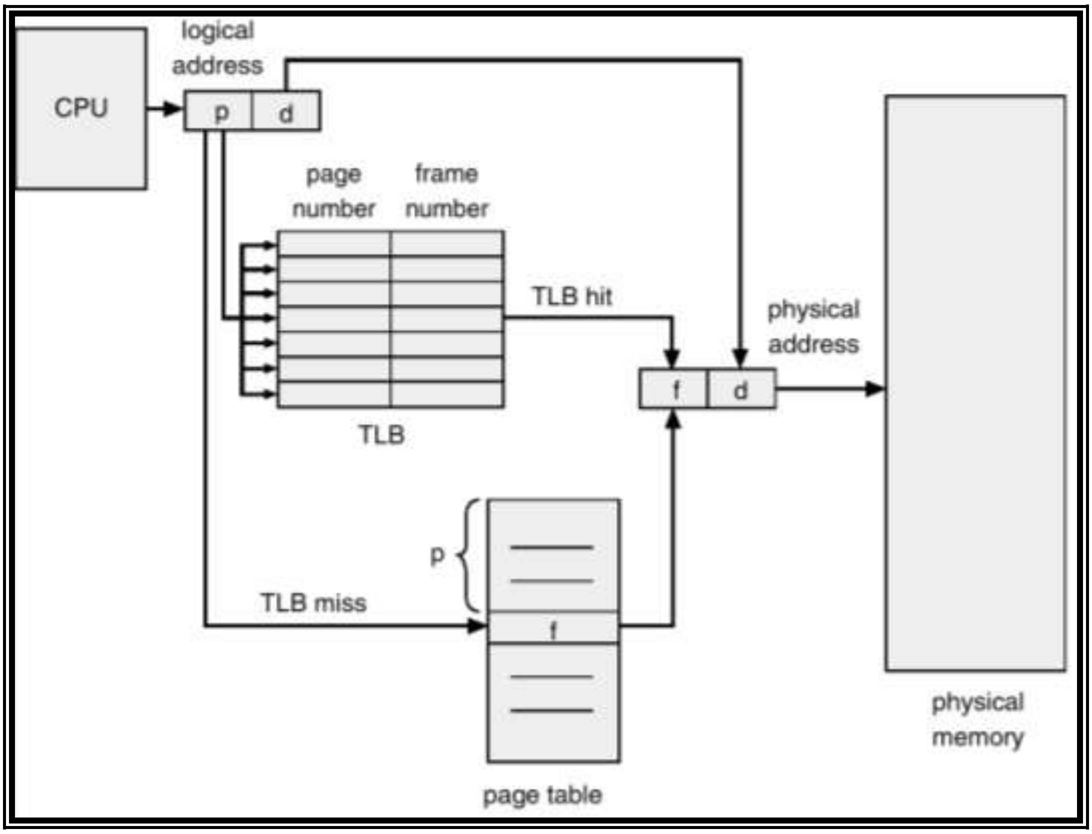
1. _____2
2. _____4
3. _____8
4. _____16
5. _____32
6. _____64
7. _____128
8. _____256
9. _____512
10. _____1024

Address Translation Scheme

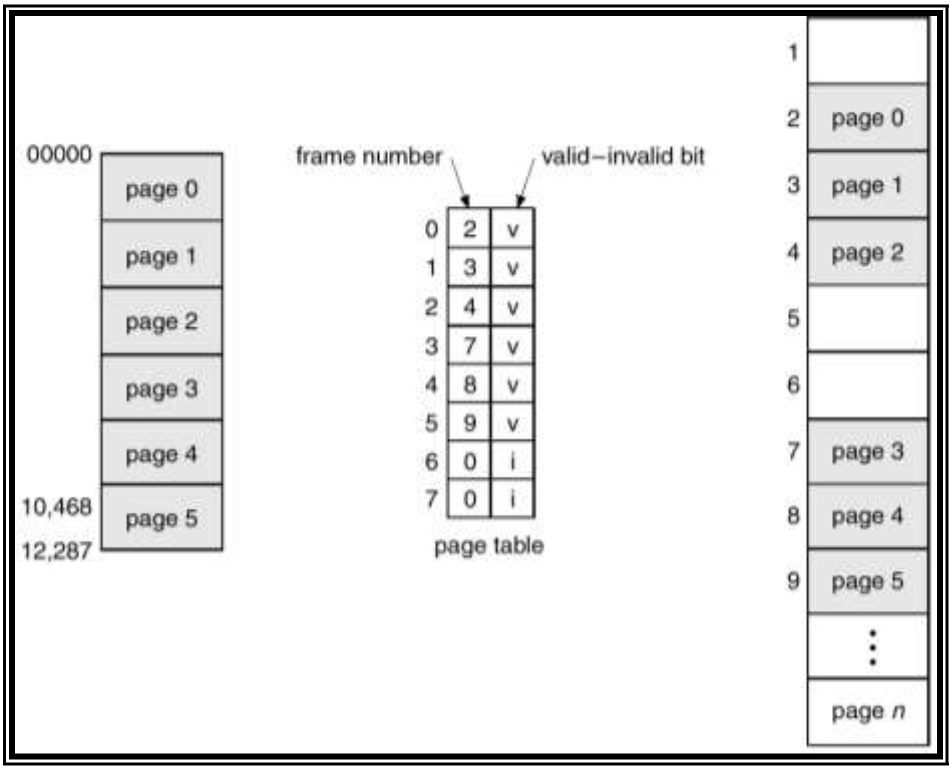


Address generated by CPU is divided into:Page number (p) and Page offset (d)

Paging With TLB hardware cache called *associative memory* or *translation look-aside buffers (TLBs)*



Memory Protection : Valid (v) or Invalid (i) Bit



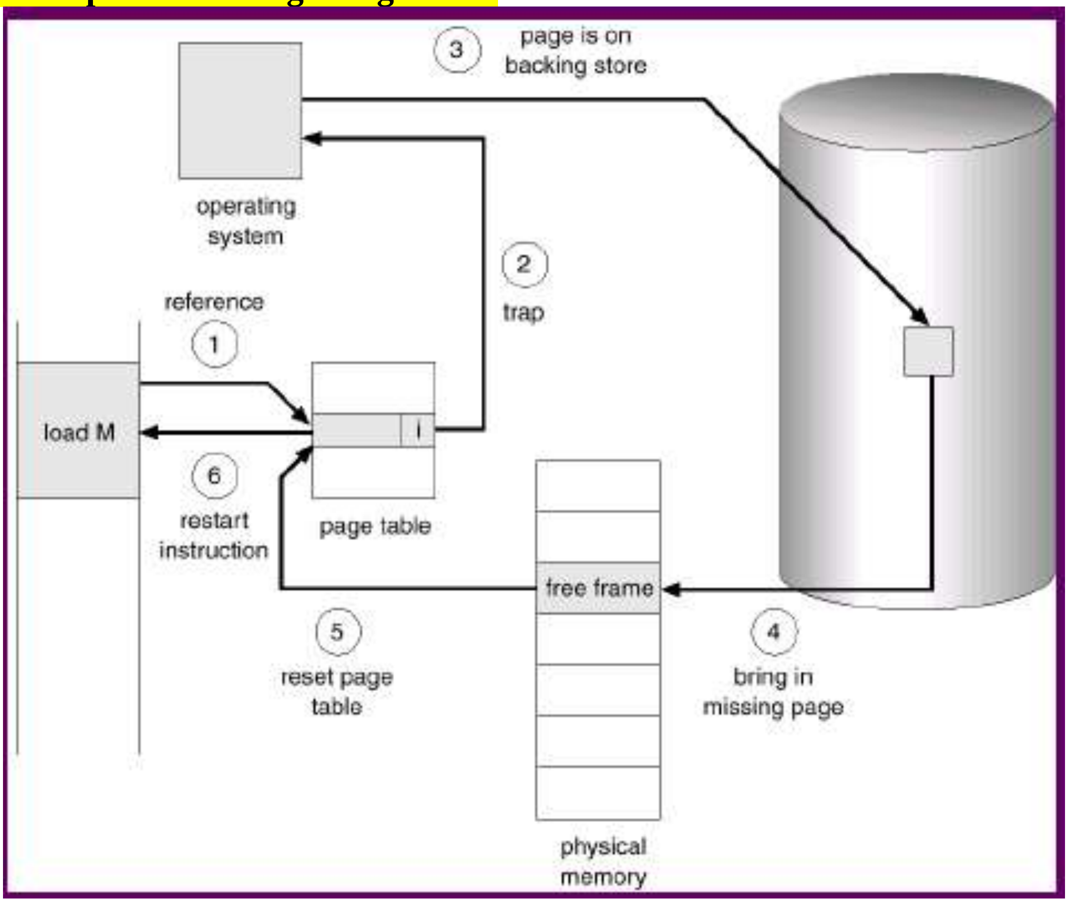
4.5.3 Virtual memory : separation of user logical memory from physical memory.

Virtual memory can be implemented via: ✦ Demand paging ✦ Demand segmentation

➔ **Demand Paging:** Bring a page into memory only when it is needed.

☺ Less I/O needed ☺ Less memory needed ☺ Faster response ☺ More users

Page Fault: Steps in Handling a Page Fault



Process Creation

1-Copy-on-Write(COW)

Allows both parent and child processes to initially share the same pages in memory. If either process modifies a shared page, only then is the page copied. COW allows more efficient process creation as only modified pages are copied. Free pages are allocated from a pool of zeroed-out pages.

2- Memory-Mapped Files

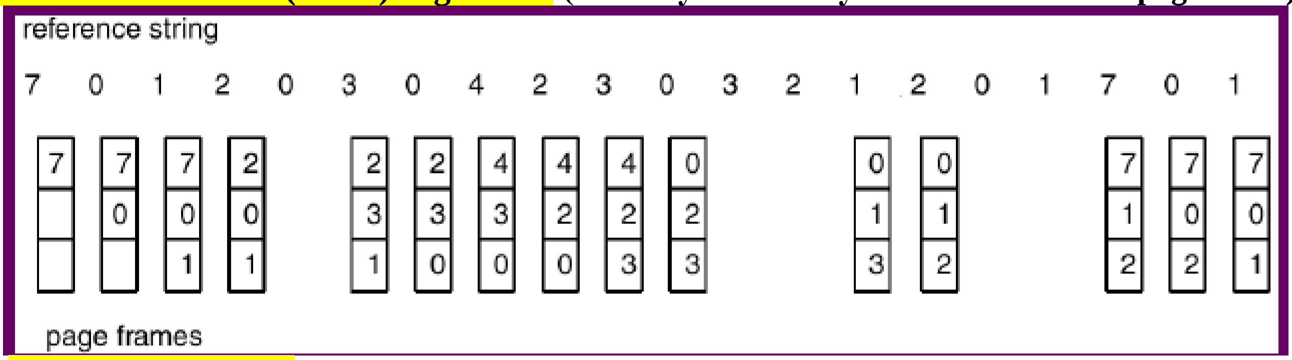
Memory-mapped file I/O allows file I/O to be treated as routine memory access by mapping a disk block to a page in memory. A file is initially read using demand paging. A page-sized portion of the file is read from the file system into a physical page. Subsequent reads/writes to/from the file are treated as ordinary memory accesses.

Page Replacement

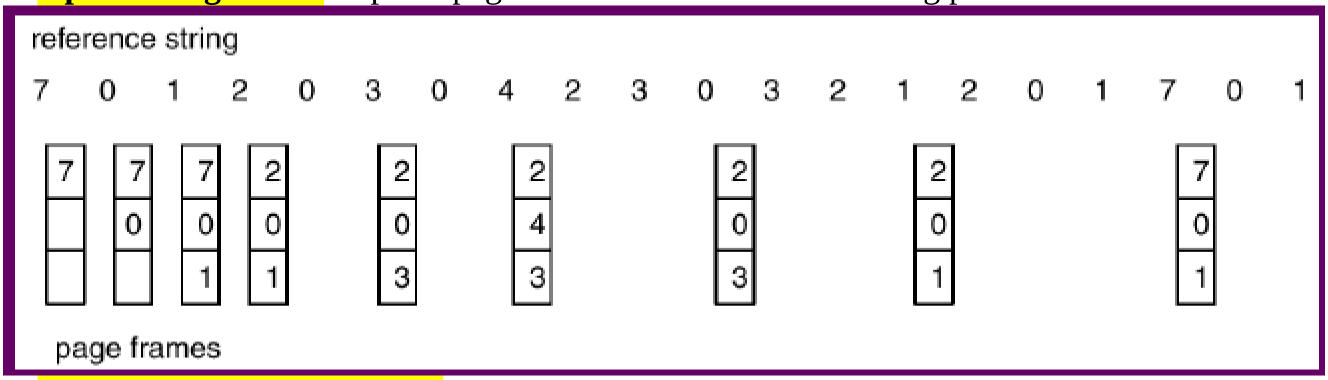
- Use modify (**dirty**) bit to reduce overhead of page transfers
  - only modified pages are written to disk.
- Page replacement completes separation between logical memory and physical memory
  - large virtual memory can be provided on a smaller physical memory.

In all our examples, the reference string is: 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5.

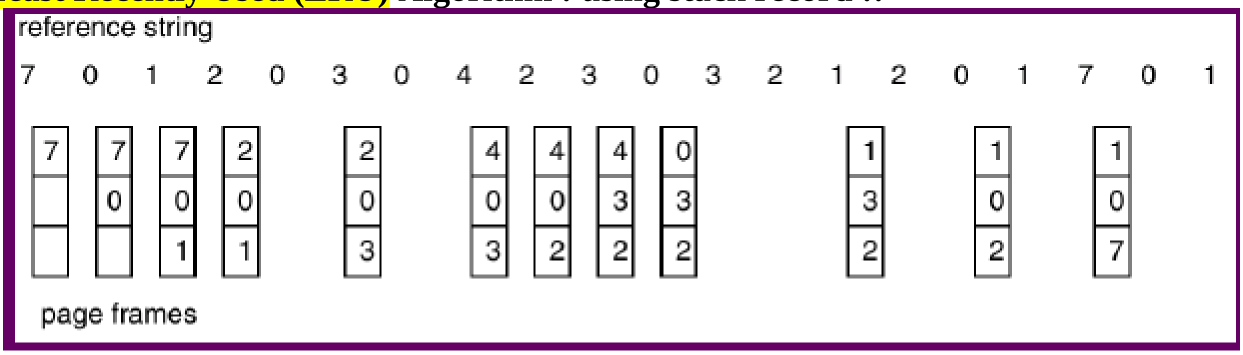
1. First-In-First-Out (FIFO) Algorithm (Beladdy's anomaly: more frames→less page faults)



2. Optimal Algorithm: replace pages that will not be used for long period of time



3. Least Recently Used (LRU) Algorithm : using stack record !!



4. LRU Approximation Algorithms

➔ **Reference bit**

- ✦ With each page associate a bit, initially = 0, ✦ When page is referenced bit set to 1.
- ✦ Replace the one which is 0 (if one exists). But, we do not know the order.

➔ ➔ **Second chance**

- ✦ Need reference bit., ✦ Clock replacement.
- ✦ If page to be replaced (in clock order) has reference bit = 1. then:
  - ✔ set reference bit 0.
  - ✔ leave page in memory.
  - ✔ replace next page (in clock order), subject to same rules.

Counting Algorithms: keep counter of number of references that have been made to each page.

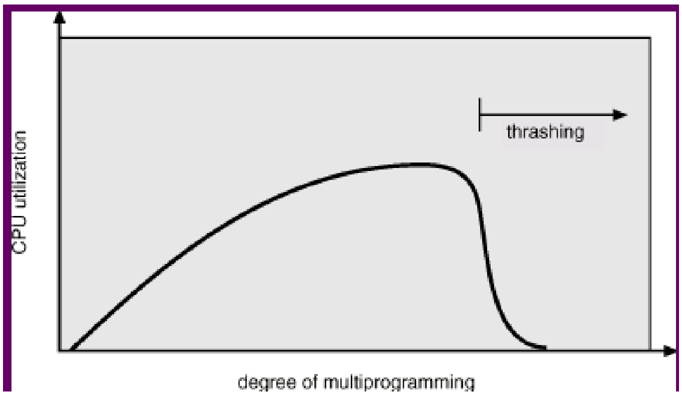
- **LFU**: replaces page with smallest count.
- **MFU**: based on that page with smallest count was just brought in and has yet to be used.

Global vs. Local Allocation

_ **Global** replacement – process selects a replacement frame from the set of all frames; one process can take a frame from another.

_ **Local** replacement – each process selects from only its own set of allocated frames.

Thrashing ≡ a process is busy swapping pages in and out.



References

- Edward L. Bosworth, Ph.D., Design and Architecture of Digital Computers:An Introduction, TSYS School of Computer Science
- Avi Silberschatz , Operating System Concepts - 8th edition