Memory Management Lecture 3

Paging

Minakshi R.

Operating System Concepts 8th edition silberschatz Galvin

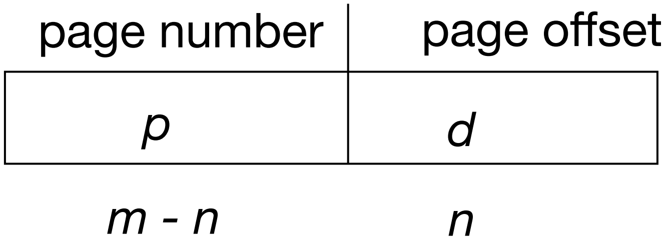
Paging

- Physical address space of a process can be noncontiguous; process is allocated physical memory whenever the latter is available
- Divide physical memory into fixed-sized blocks called **frames**
  - Size is power of 2, between 512 bytes and 16 Mbytes
- Divide logical memory into blocks of same size called **pages**
- Keep track of all free frames
- To run a program of size *N* pages, need to find *N* free frames and load program
- Set up a **page table** to translate logical to physical addresses
- Backing store likewise split into pages
- Still have Internal fragmentation

Address Translation Scheme

Address generated by CPU is divided into:

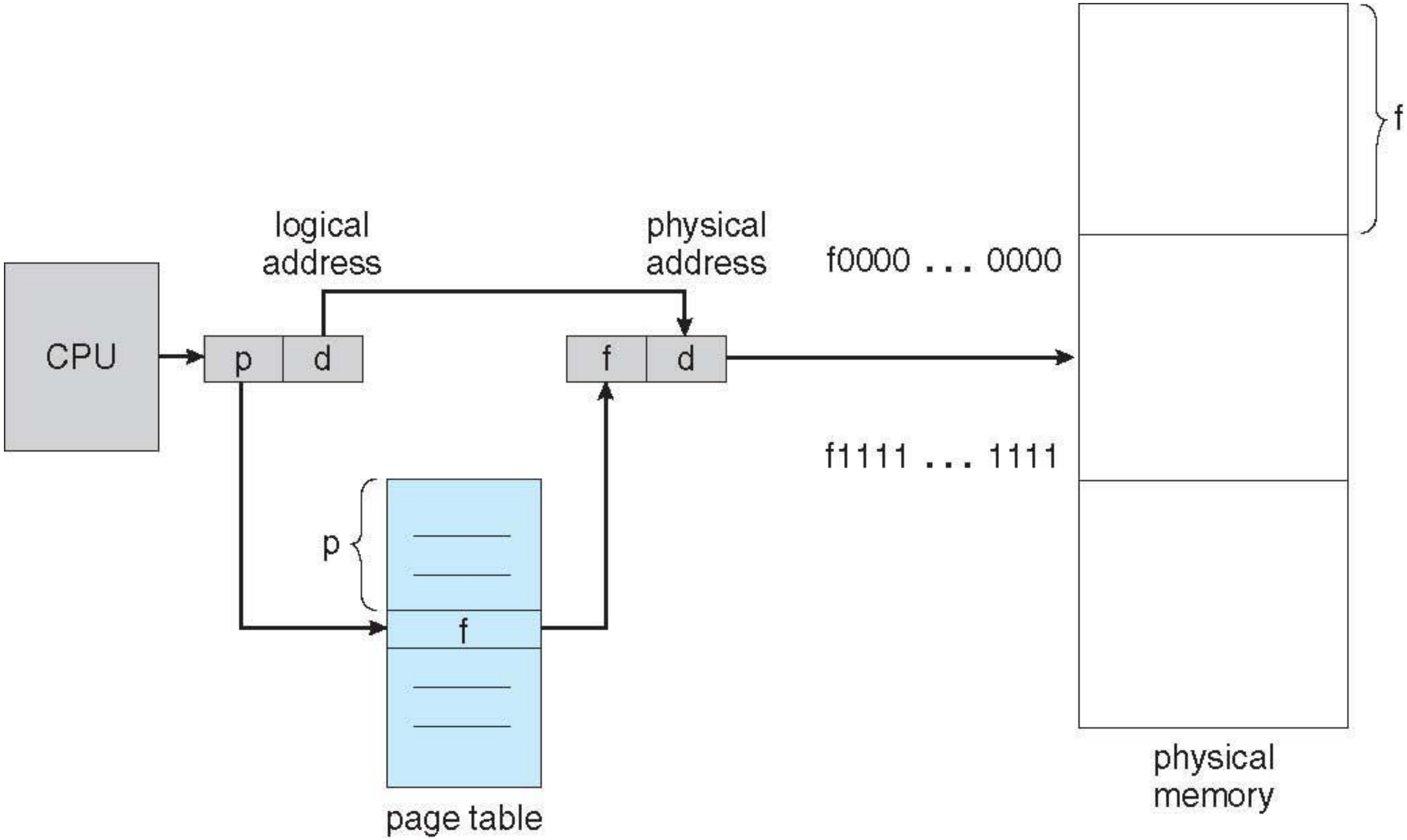
- **Page number (*p*)** – used as an index into a **page table** which contains base address of each page in physical memory
- **Page offset (*d*)** – combined with base address to define the physical memory address that is sent to the memory unit



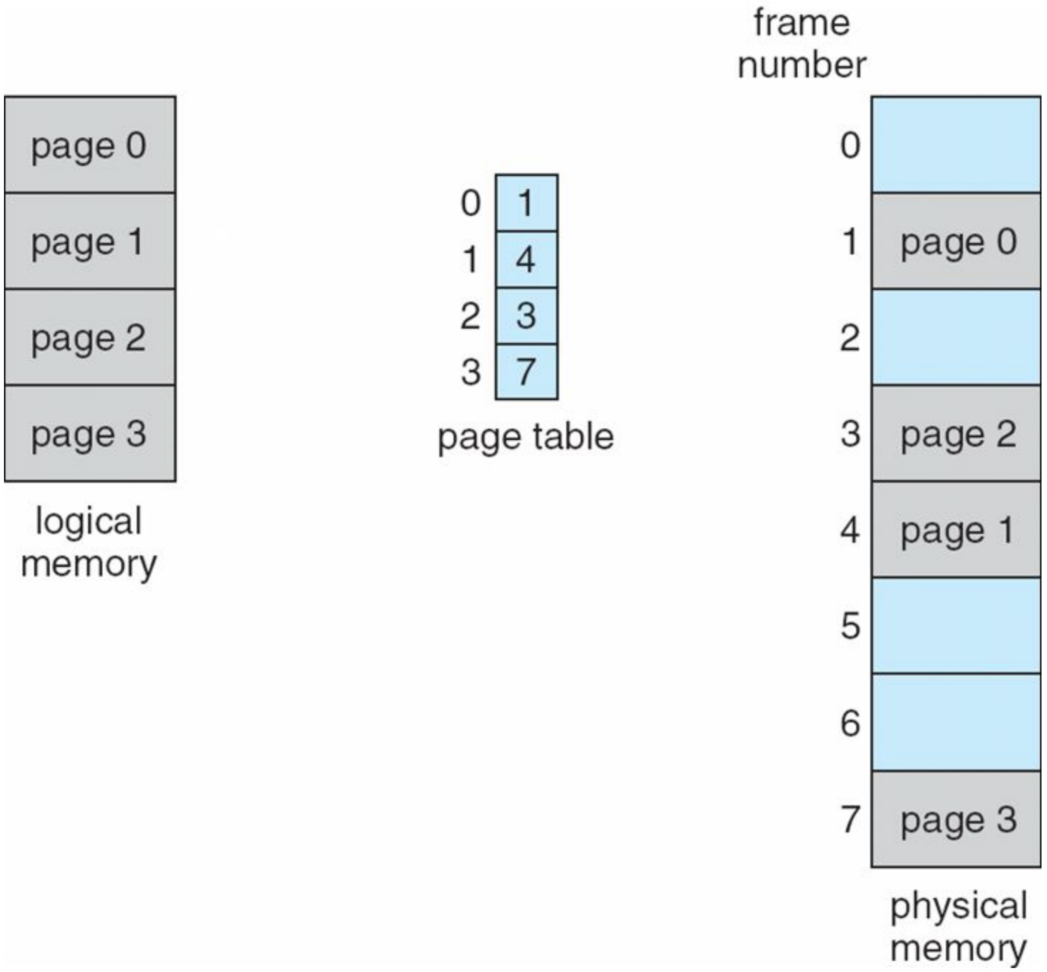
page number	page offset
p	*d*
m - n	*n*

- For given logical address space 2^m and page size 2^n

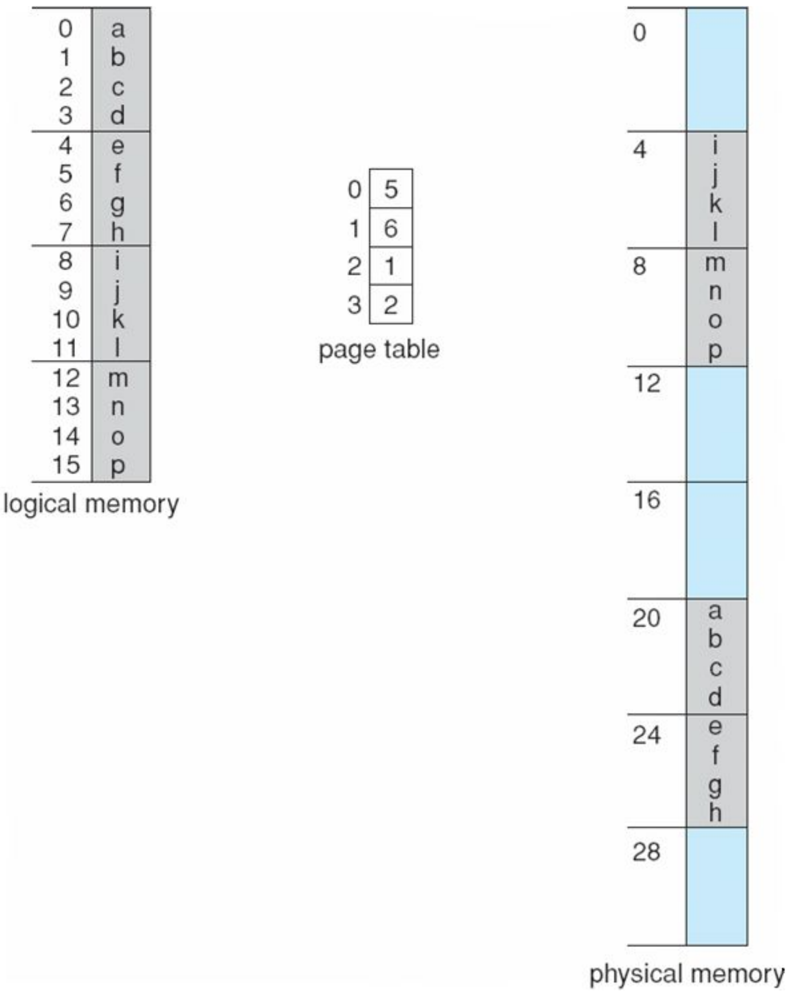
Paging Hardware



Paging Model of Logical and Physical Memory



Paging Example

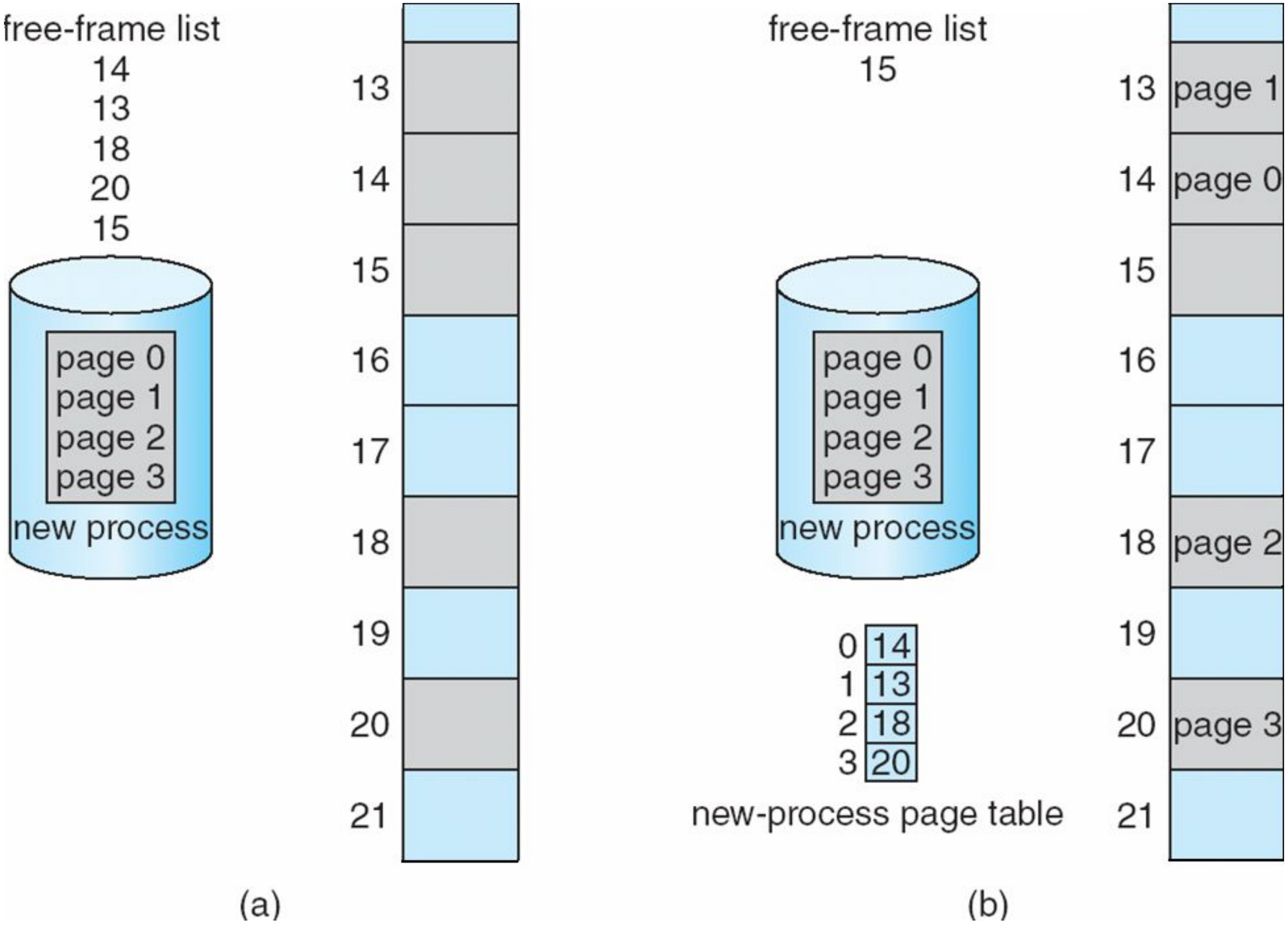


n=2 and *m*=4   32-byte memory and 4-byte pages

Paging (Cont.)

- No external fragmentation
- Calculating internal fragmentation
  - example:
  - If pages are of 2048 bytes, a process of 72,766. How much frames will be needed?
    - Page size = 2,048 bytes
    - Process size = 72,766 bytes
    - 35 pages + 1,086 bytes
    - Internal fragmentation of 2,048 - 1,086 = 962 bytes
    - Worst case fragmentation = 1 frame – 1 byte
    - On average fragmentation = 1 / 2 frame size
    - So small frame sizes desirable?
    - But each page table entry takes memory to track
- Process view and physical memory now very different

Free Frames



Before allocation

After allocation

THANK YOU