操作系统

Operating system

孔维强

大连理工大学

内容纲要

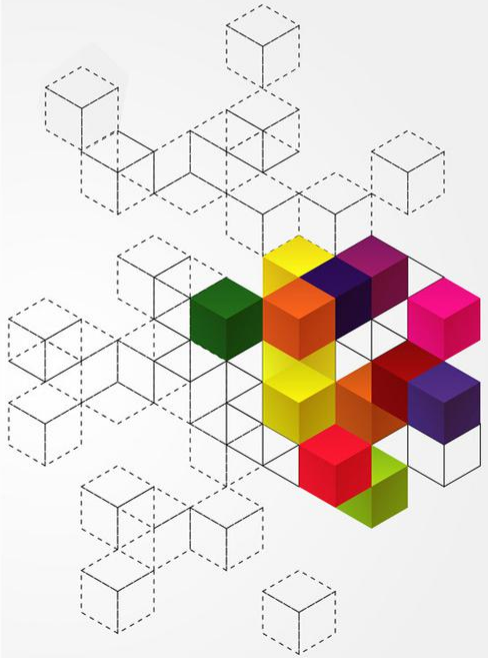
8.3 分页机制

一、 引入分页的意义

二、 分页原理

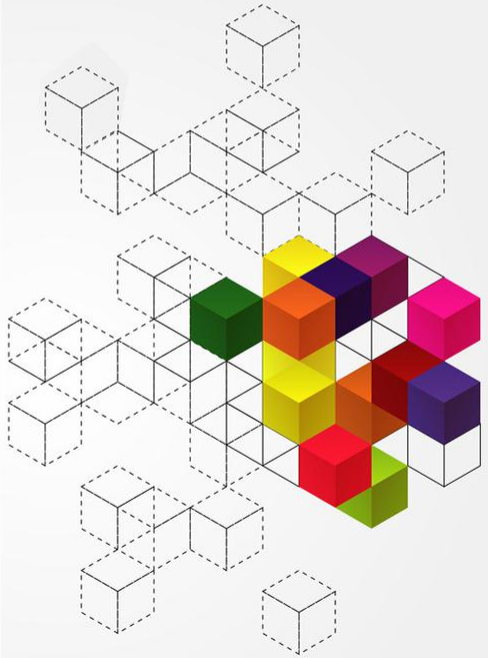
三、 分页机制下的内存保护

四、 分页机制下的内存共享



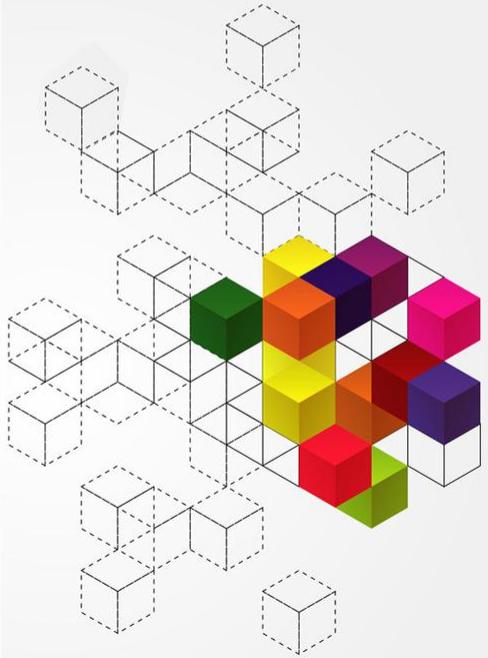
一、引入分页的意义

- **为什么要引入分页?**
  - 这是在计算机硬件性能提升，内存容量增大，多任务概念得以实现的情况下，分析连续内存分配的弊端，而进行的技术创新
  - 连续分区分配的问题
    - 容易形成数量较多的较大内存碎片，导致内存使用效率降低
    - 碎片大小不可预估，不好控制
  - 分页机制可以有效解决连续内存分配的问题，提升内存使用效率



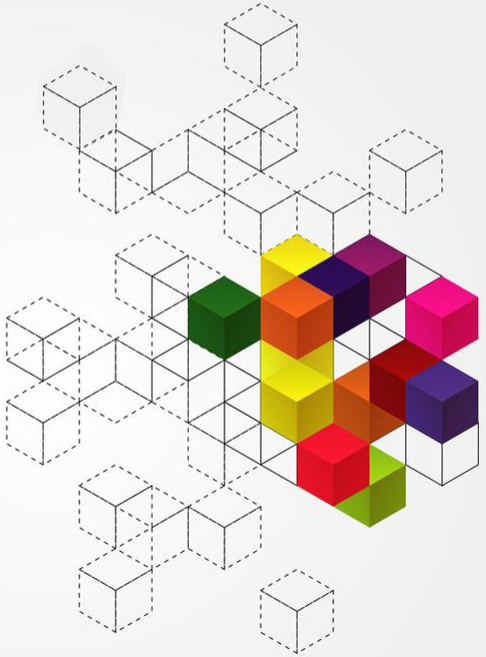
二、分页原理

- **分页基本思想**
  - 将进程的逻辑地址空间等分为同样的大小的块，每个这样的块称为逻辑页（Page，页）
  - 将物理内存等分为同样大小的块，每个这样的块称为物理页（frame，帧）
  - 如果进程的逻辑空间划分为N个页，那么在进程执行时，操作系统为其分配N个物理页，用以存放逻辑页面内容
    - 逻辑页与物理页形成1：1映射关系



二、分页原理

- **分页机制下，操作系统的内存管理任务变为**
  - 跟踪进程的页面使用情况。为每个进程维护一个页面与物理页框之间的映射表格（页表）
  - 跟踪系统内的物理页框情况。在进程需要新的物理页框时，从现有空闲页框中进行分配；在进程退出或进程释放内存时，回收物理页框资源



二、分页原理

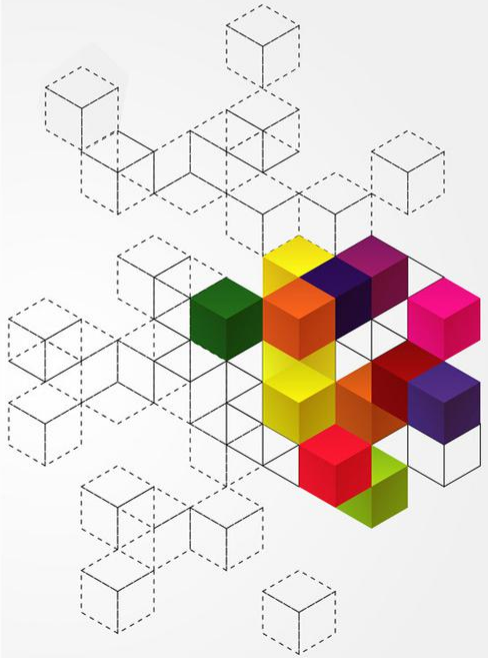
- **分页的三个重要环节**

地址划分 → 建立地址映射（页表） → 地址翻译

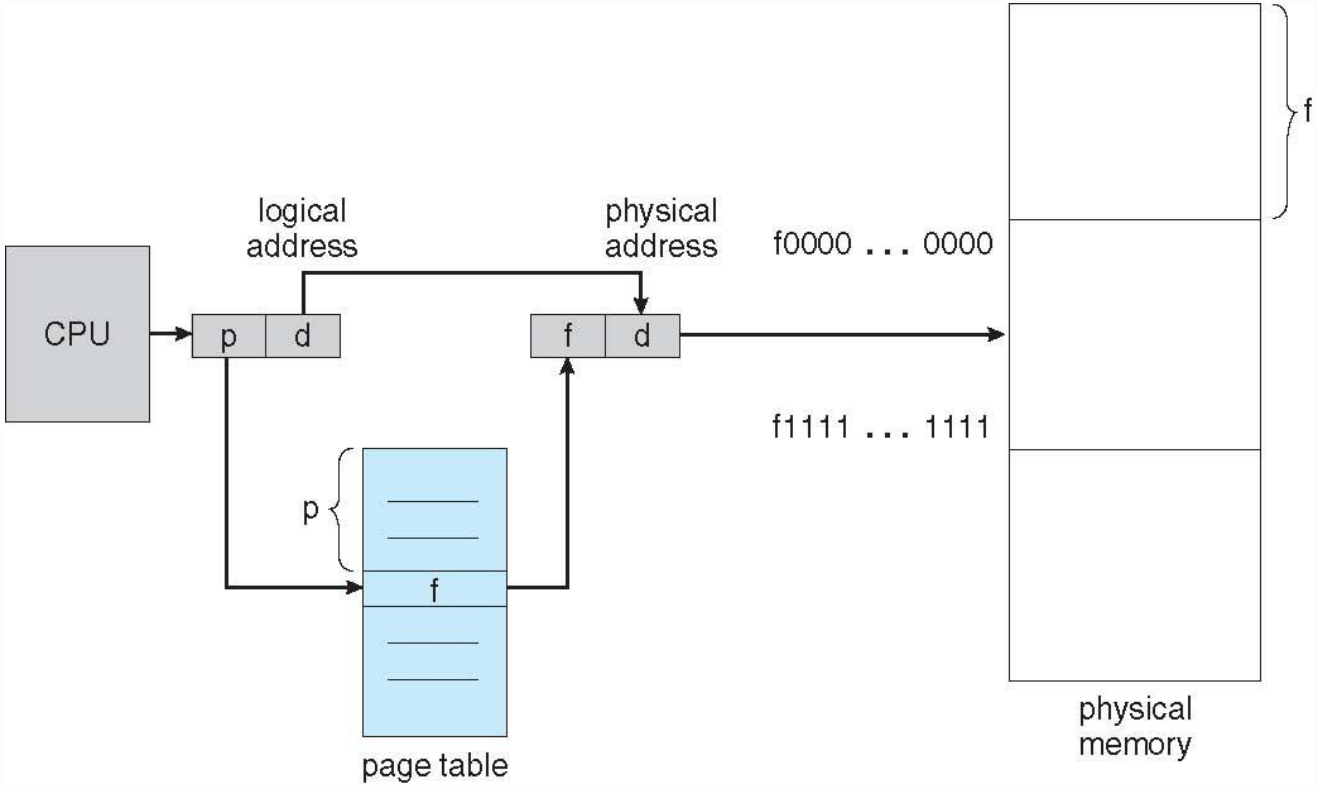
地址被按位拆分成页号和页内偏移这2部分

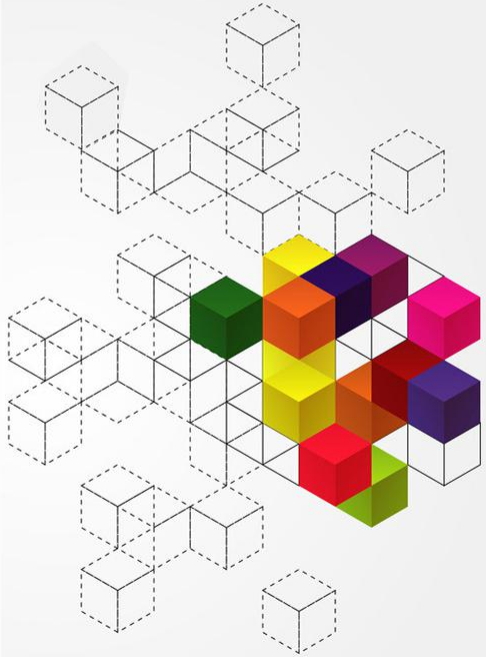
page number	page offset
p	*d*
m - n	*n*

假定逻辑地址空间为2^m，页大小为2^n



二、分页原理

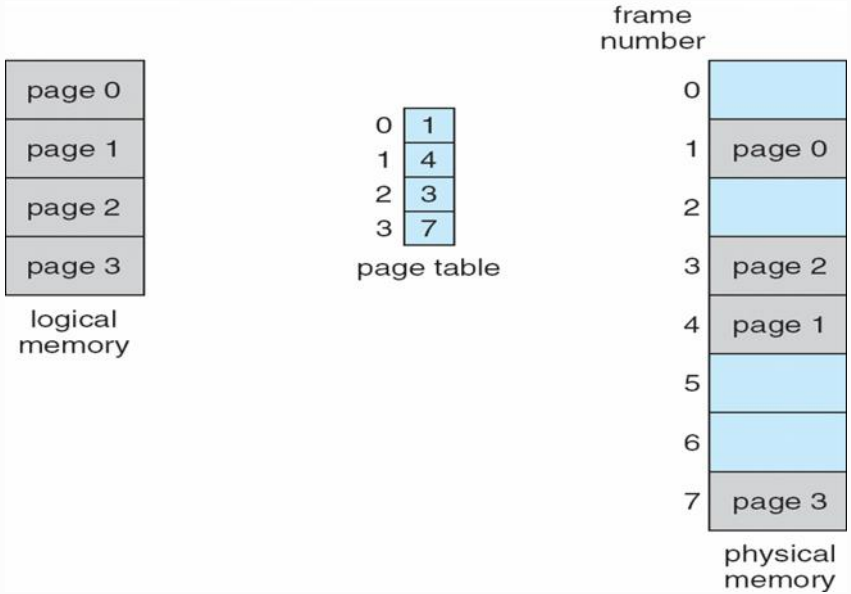

logical
address
physical
address
CPU
p
d
f
d
f0000 . . . 0000
f1111 . . . 1111
f
p
f
page table
physical
memory

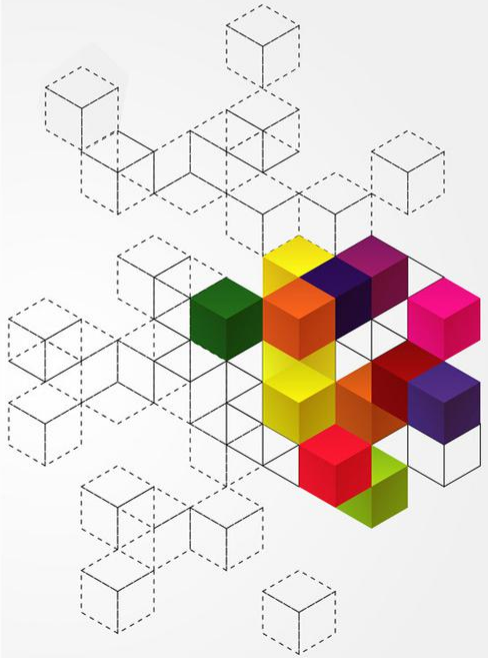


二、分页原理

- **分页的三个重要环节**

地址划分 → 建立地址映射（页表） → 地址翻译

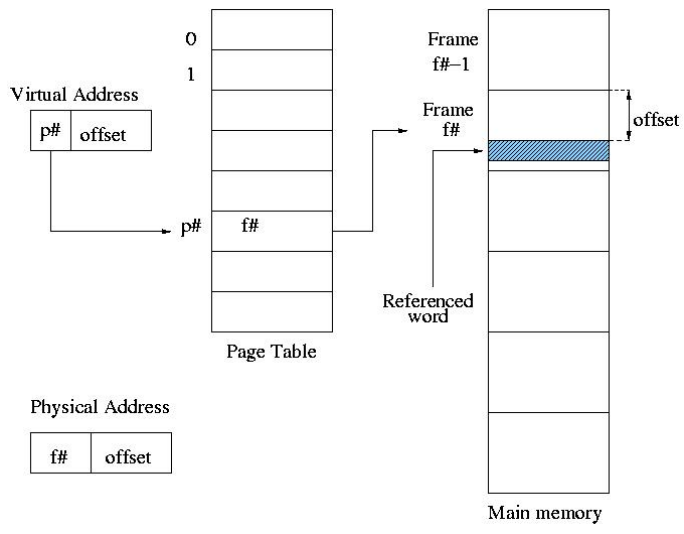


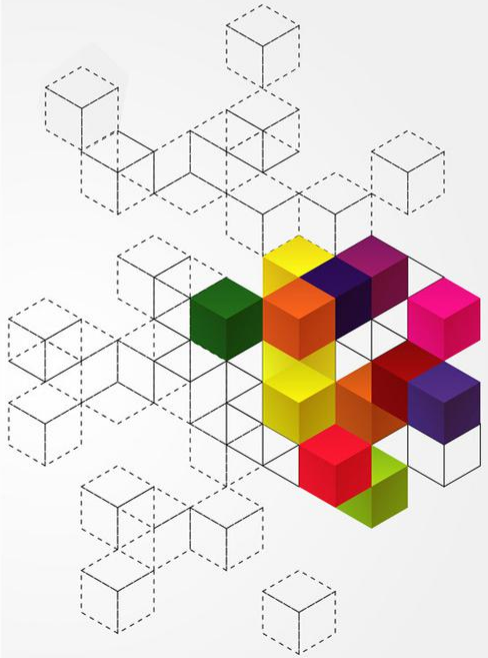


二、分页原理

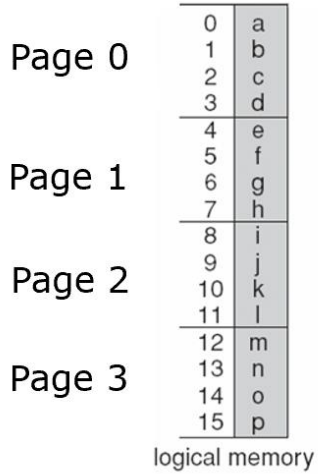
- **分页的三个重要环节**

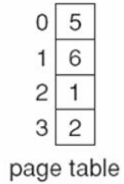
地址划分 → 建立地址映射（页表） → 地址翻译

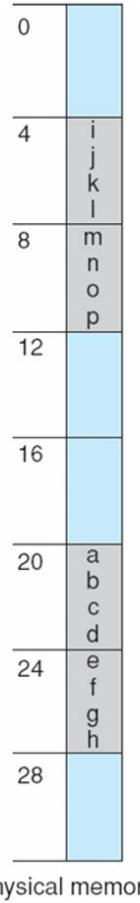




二、分页原理

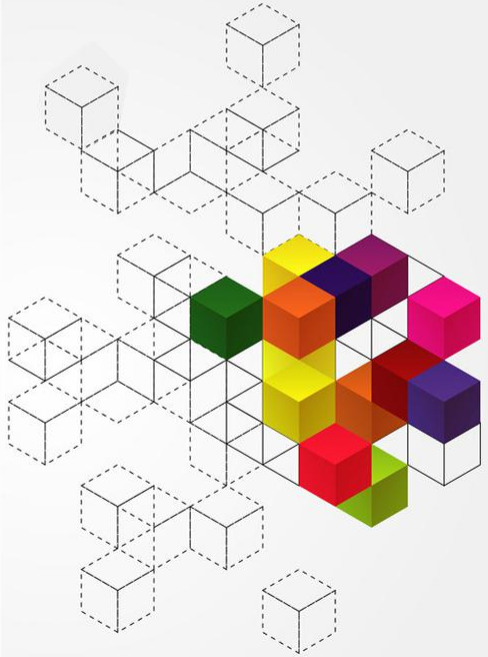






Address 0: page 0, offset 0
Physical address:
5*4 + 0 = 20

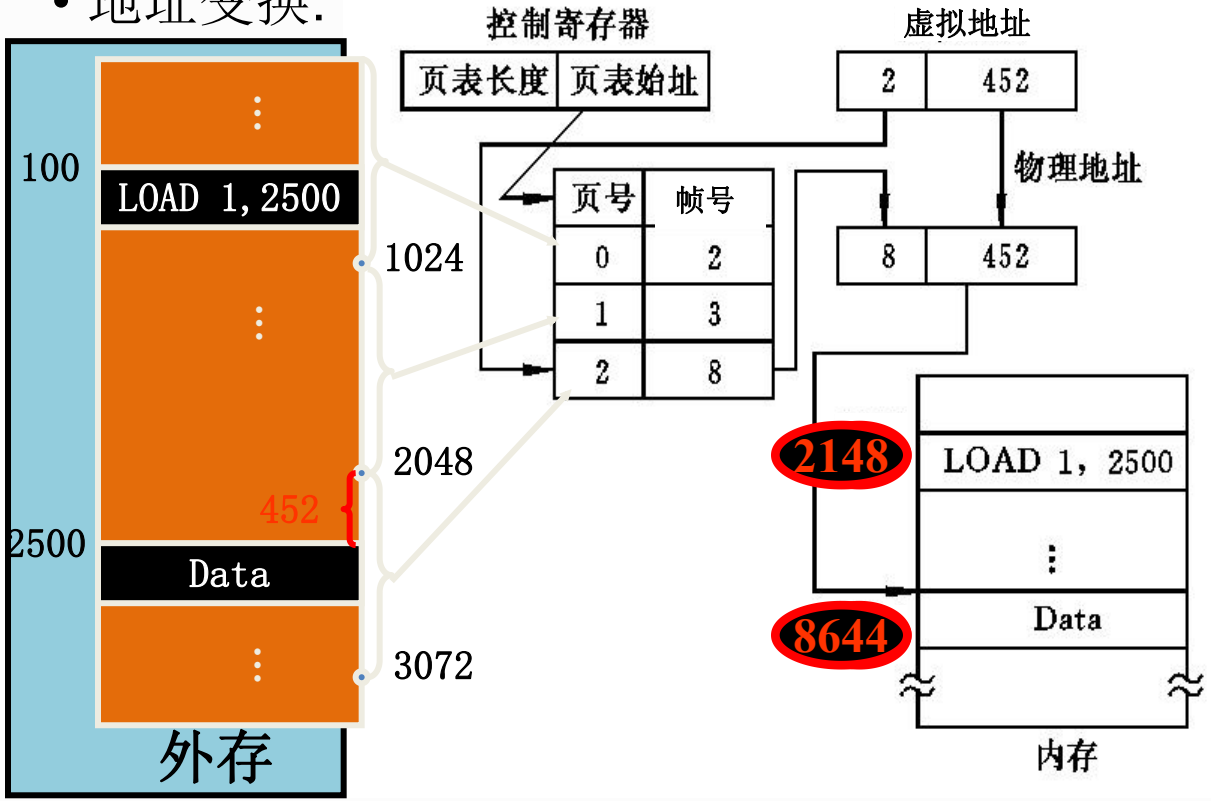
Address 3: page 0, offset 3
Physical address:
5*4 + 3 = 23

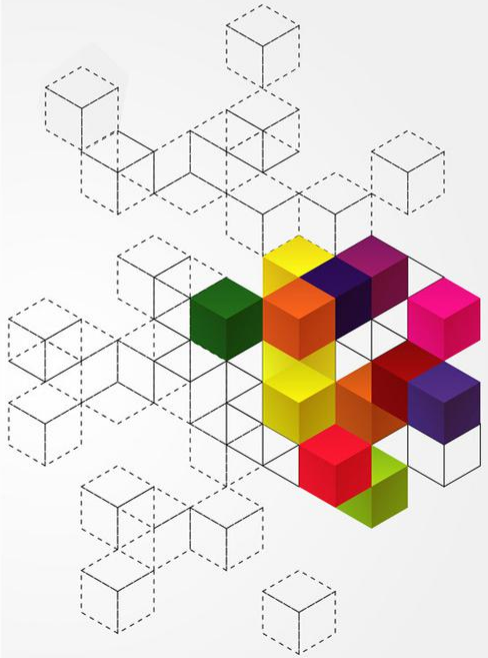


二、分页原理

设每个页面长度为1K

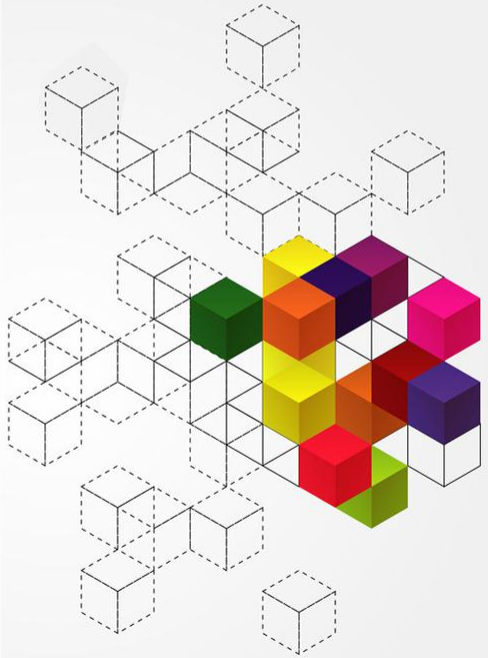
- 地址变换:





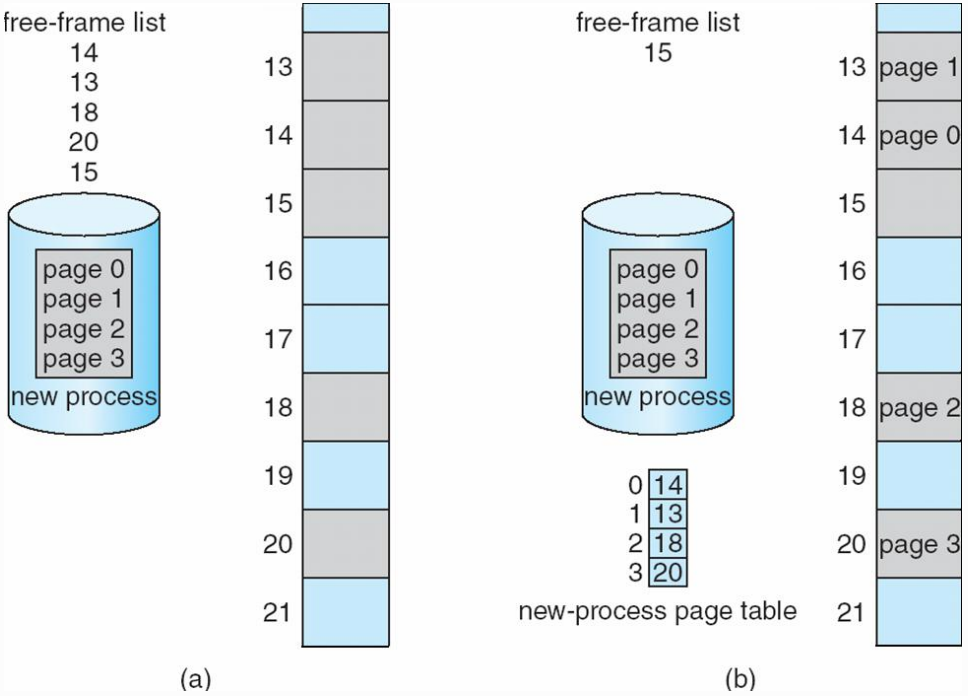
二、分页原理

- 计算内部碎片（分页系统中无外部碎片）
  - 页大小 = 2,048 bytes
  - 进程大小 = 72,766 bytes
  - 35 页 + 1,086 bytes
  - 内部碎片 2,048 - 1,086 = 962 bytes
  - 最差情况碎片 = 1 frame – 1 byte
  - 平均情况碎片 = 1 / 2 frame size
  - 因此，页/帧越小越好？（页表项也占用内存）
  - 页大小随技术发展逐渐变大
    - Solaris 支持2种页大小 – 8 KB and 4 MB



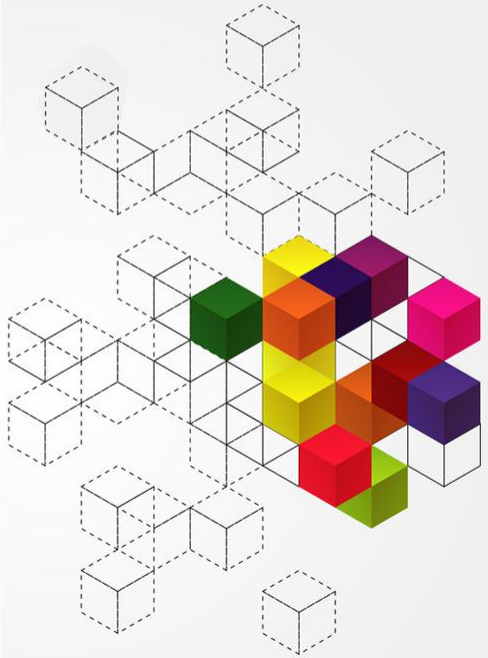
二、分页原理

空闲帧



分配前

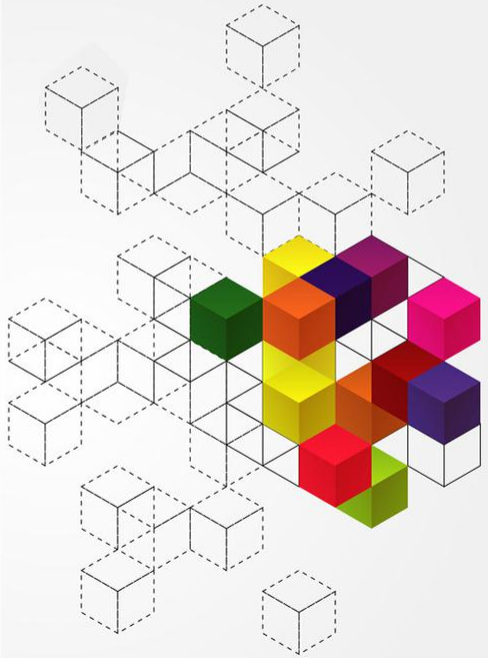
分配后



三、分页机制下的内存保护

- **通过页表项中的低位，存储页保护信息**
  - 有效位
  - 读写位
  - 禁止执行位

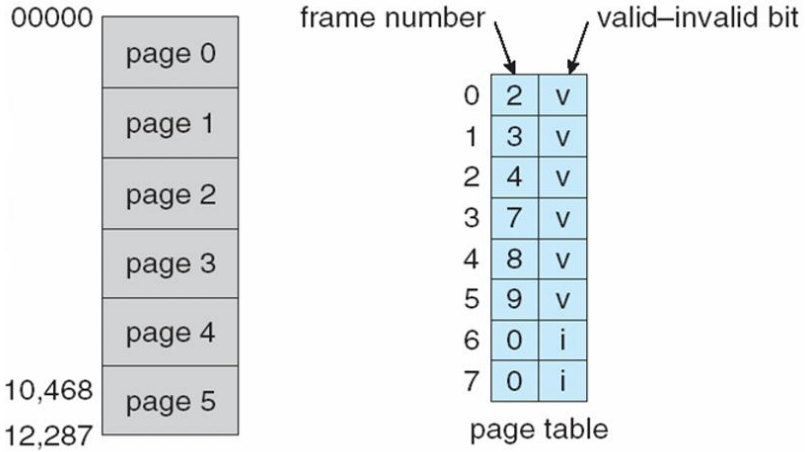
Page Number	Frame Number	Valid	Read Only	NX
0x40a1	0x0100	1	0	1
0x40a2	0x0200	0	0	1
0x40a3	0x00a0	0	0	1
0x40a4	0x0a00	1	1	0
0x40a5	0x0300	1	0	1

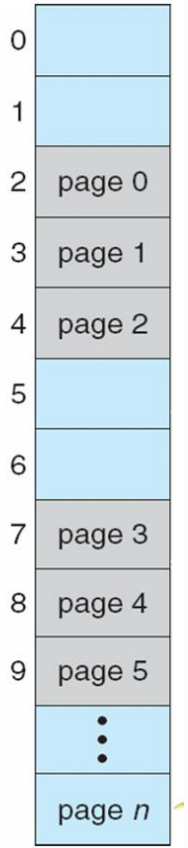


- 可以为每个页提供足够保护信息
- 违反规定的操作将导致trap

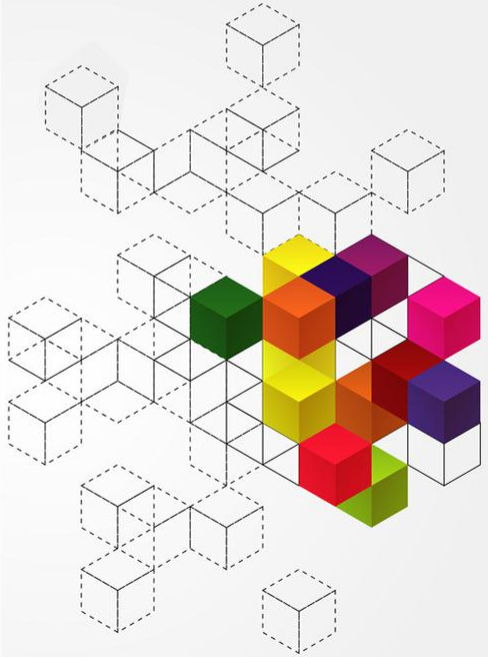
三、分页机制下的内存保护

A process uses addresses 0 - 10468



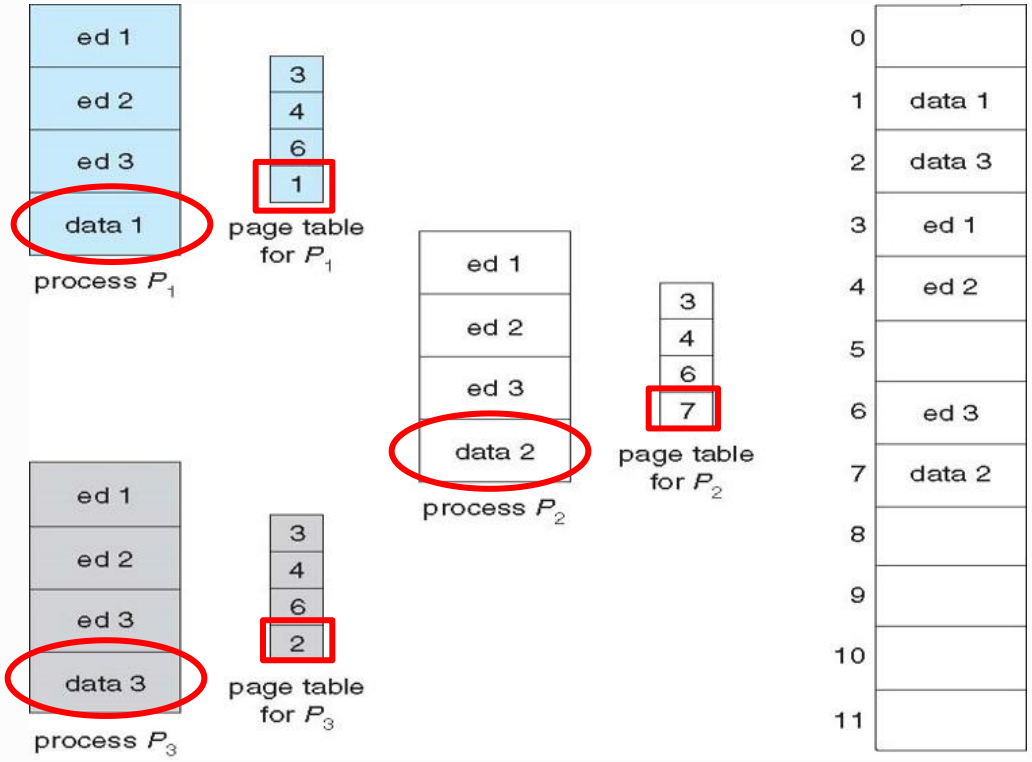


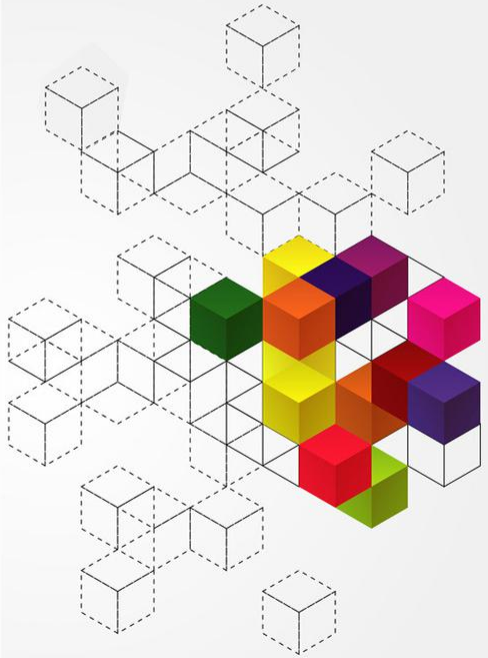
Problem: addresses 10468 – 12287 is illegal but access page 5 is legal. Some system uses additional hardware to handle this



四、分页机制下的内存共享

- **分页机制下内存共享示意图**



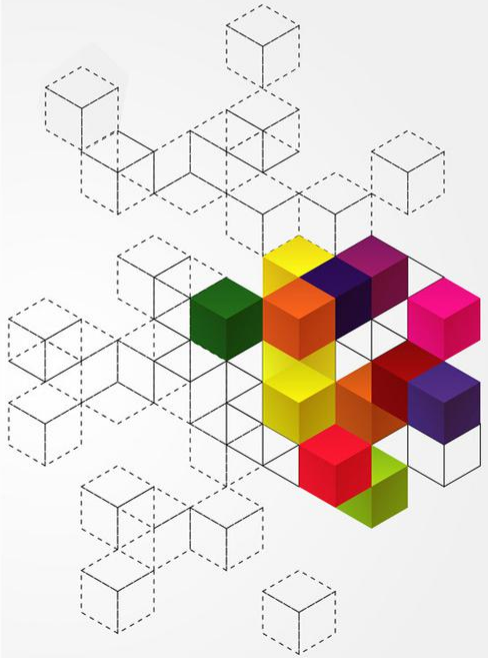


四、分页机制下的内存共享

- **共享代码**

进程间共享一份只读（可重入）代码，如文本编辑器、编译器等；与多线程共享进程空间类似

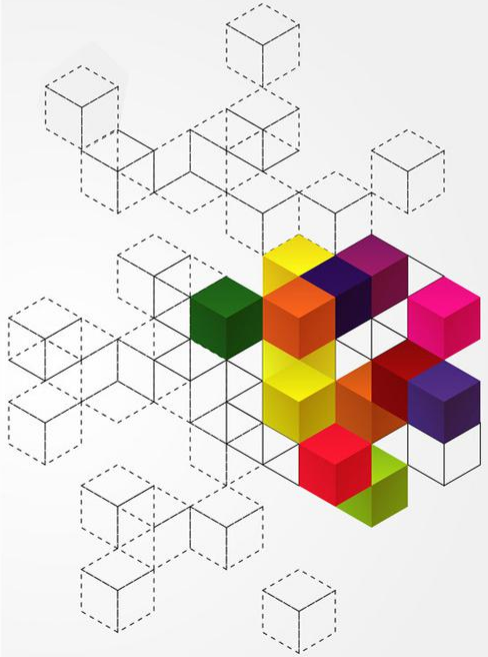
可重入代码（reentrant code）：不自我修改的代码，即在运行中用不更改



练习2

- 设有8页的逻辑空间，每页有1024字节，它们被映射到32个页面的物理存储区中，那么，逻辑地址的有效位为（）位，物理地址至少是（）位？

- $8=2^3$页逻辑空间，表示页号位数是3，每页大小$1024=2^{10}$，说明页内偏移量为10位，故逻辑地址的有效位为3+10＝13
- 内存页面为$32=2^5$，而每个内存块大小与页相同，即需10位表示，因此至少需要15位。



本讲小结

- 引入分页的意义
- 分页原理
- 分页机制下的内存保护
- 分页机制下的内存共享