

|| 高速缓冲存储器

- Cache基本原理
- 相联存储器
- 主存与Cache的地址映射
- **替换策略与写操作策略**
- Cache实际应用

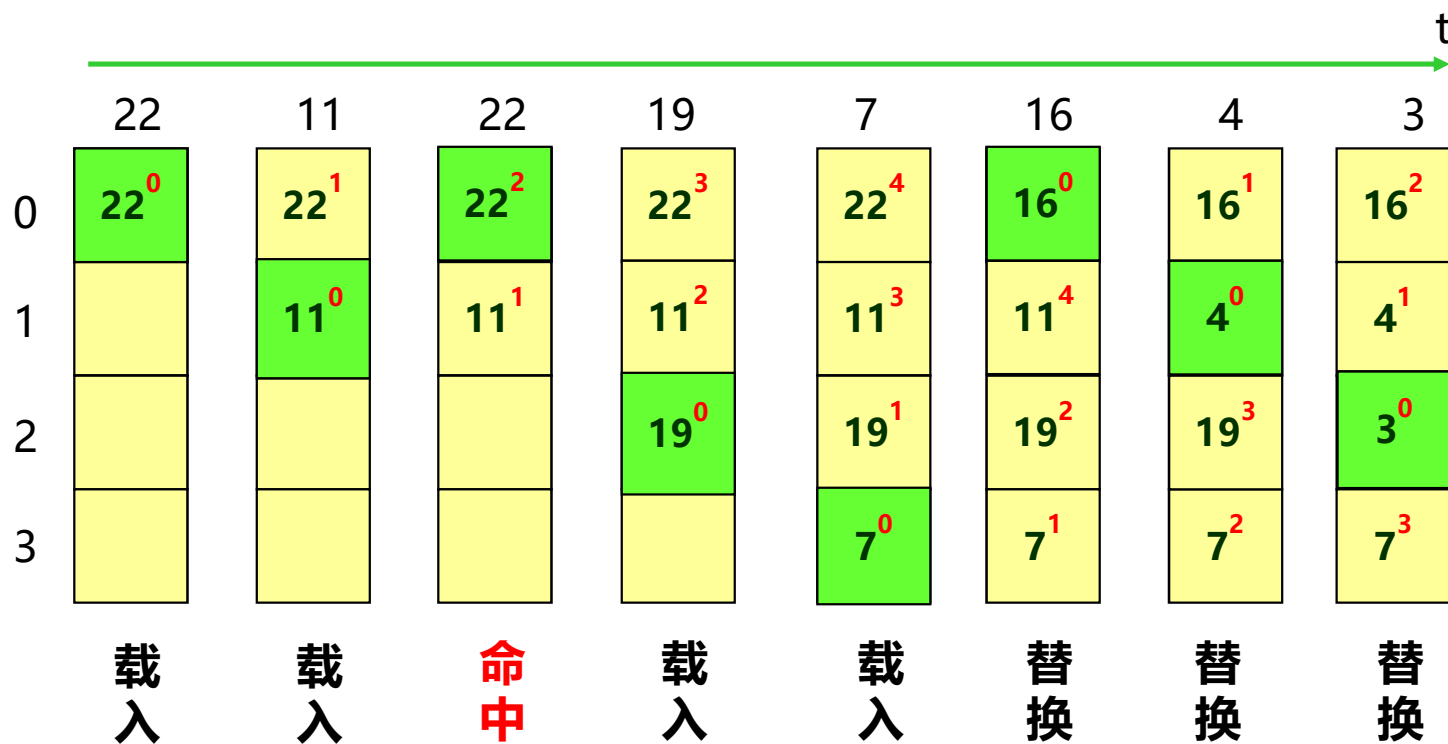
替换策略与写操作策略

■ 替换策略

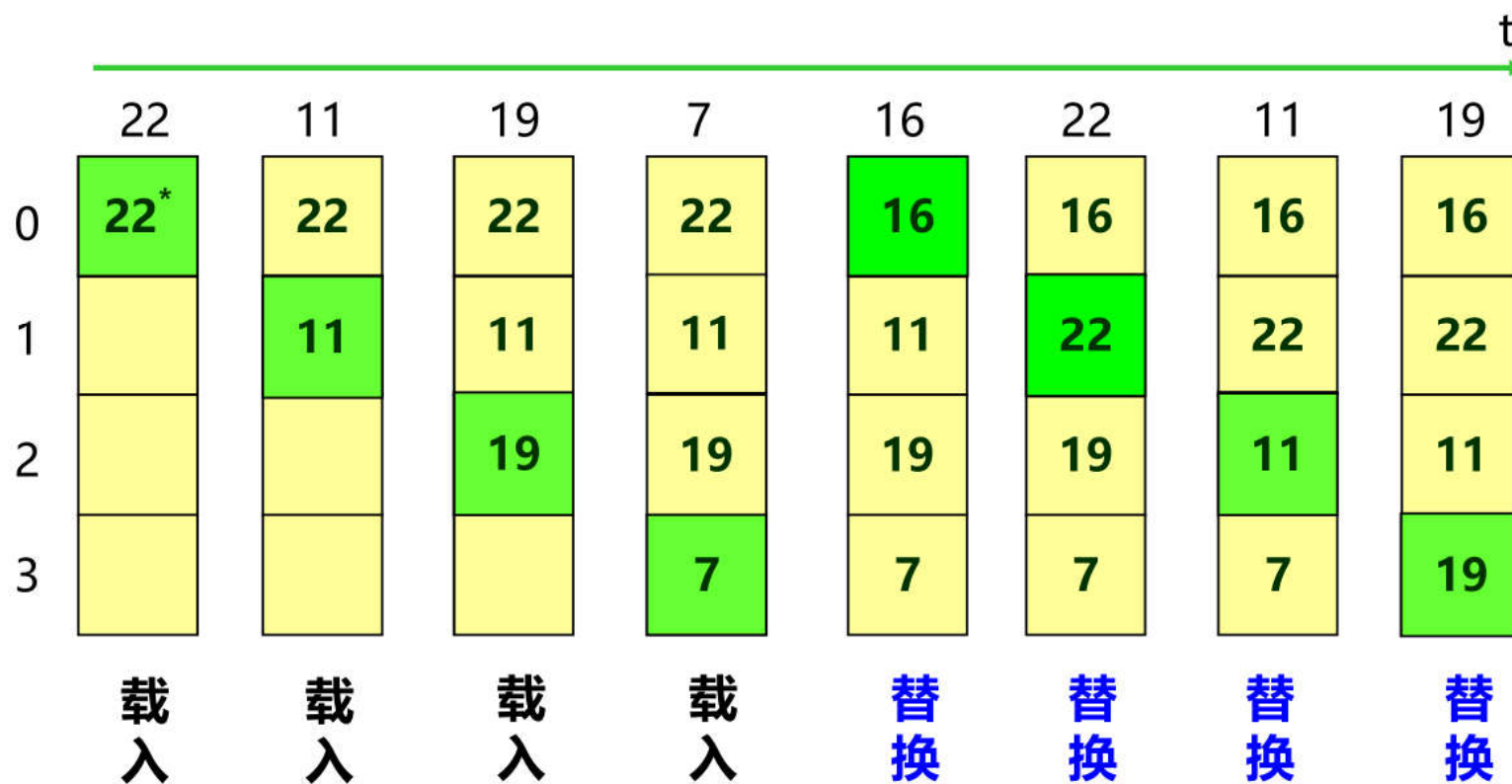
- 先进先出法
- 最近最不经常使用方法---LFU
- 近期最少使用法--- LRU
- 随机替换法

Cache先进先出替换策略(FIFO)

记录载入时间计数

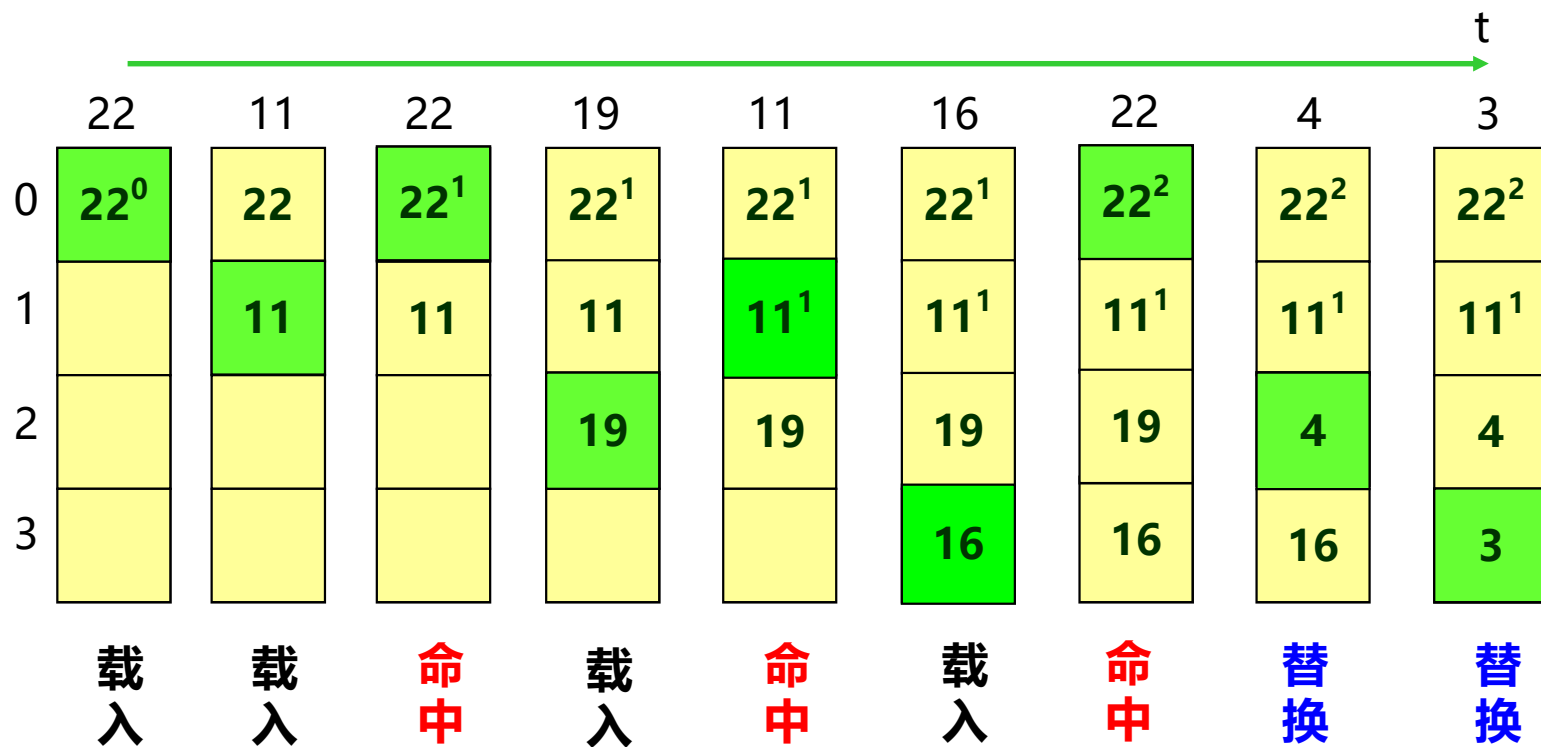


先进先出策略下的替换颠簸



Cache最不经常使用算法(LFU)

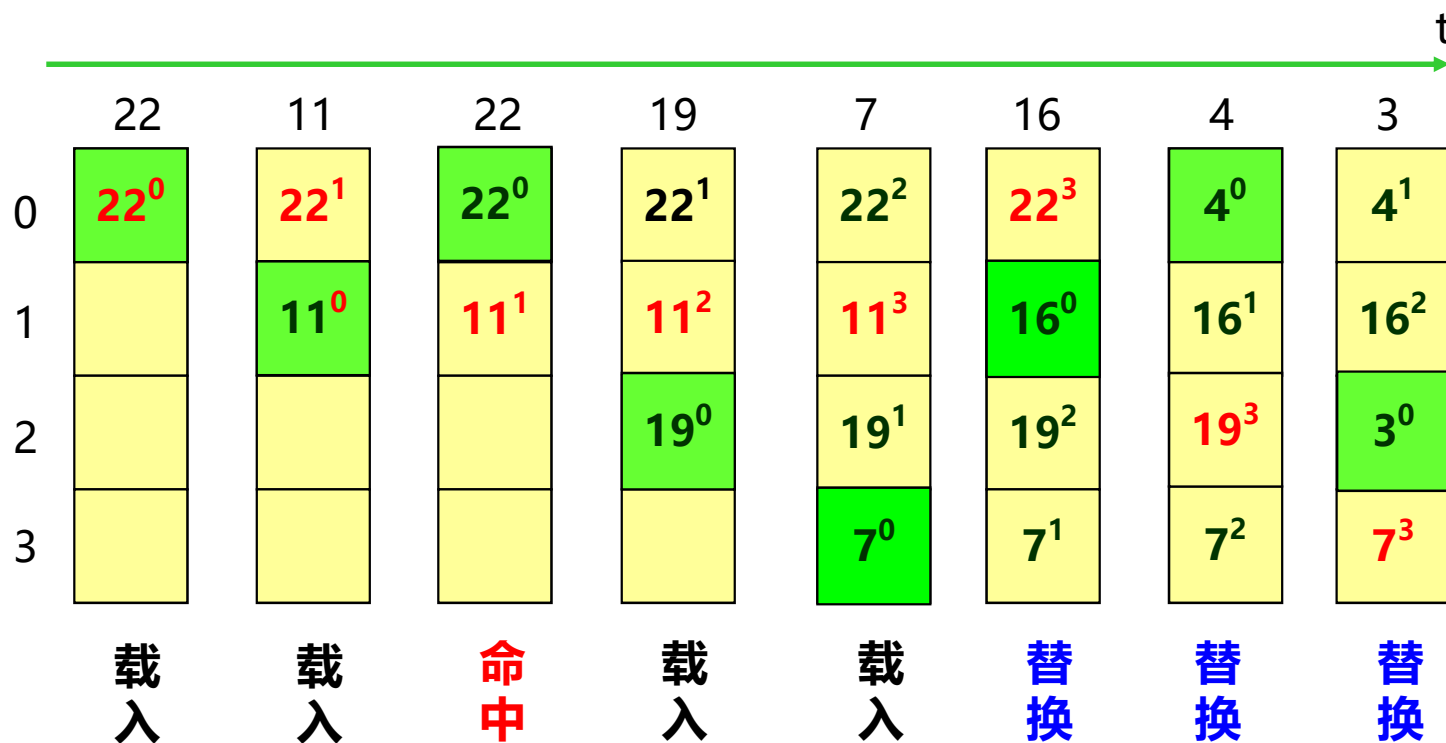
记录使用的次数



可能导致早期频繁使用数据被保留，而近期要用数据被替换

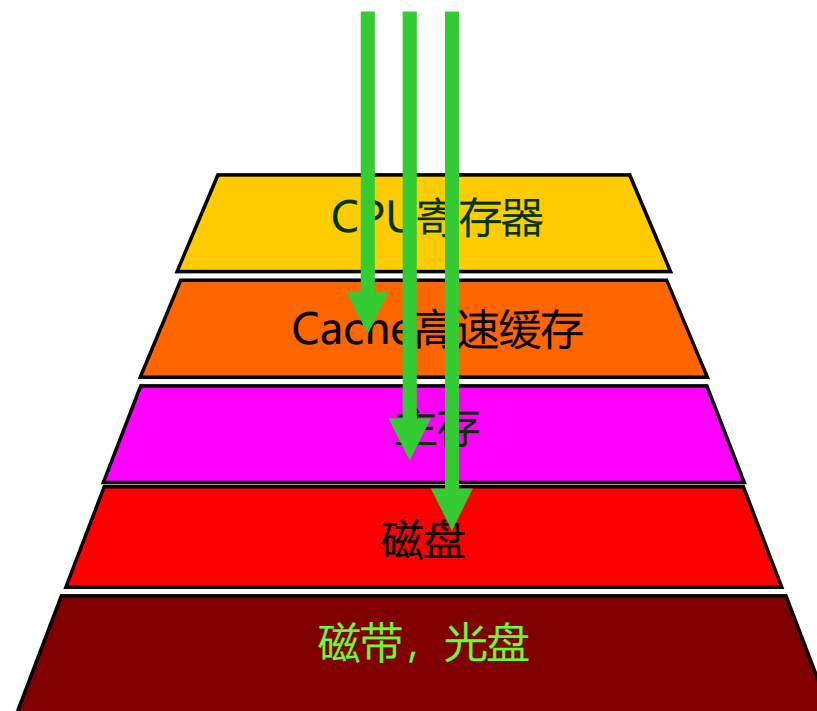
Cache近期最久未使用算法(LRU)

记录未使用的次数



写入策略

- 写回法(write back)
- 写穿法 (write through)
- 写分配 (write-allocate)
- 写不分配 (not-write-allocate)



cache对存储系统性能的影响

■ 读优化

□ 时间局部性

◆ 将刚访问的数据调度到cache中、利用淘汰算法将不经常使用的数据淘汰

□ 空间局部性

◆ 大块预读，相邻的数据被调度到cache中

■ 写优化

□ 写回策略提升突发写性能

■ 负面影响

□ 写回策略引起不一致性

□ 缓冲区满后，写性能降低

cache命中率

■ 命中率

- N_c 表示cache完成存取访问的总次数
- N_m 表示主存完成存取访问的总次数

$$h = \frac{N_c}{N_c + N_m}$$

■ t_a 平均访问时间

- t_c 表示命中cache时的访问时间
- t_m 表示命中主存时的访问时间

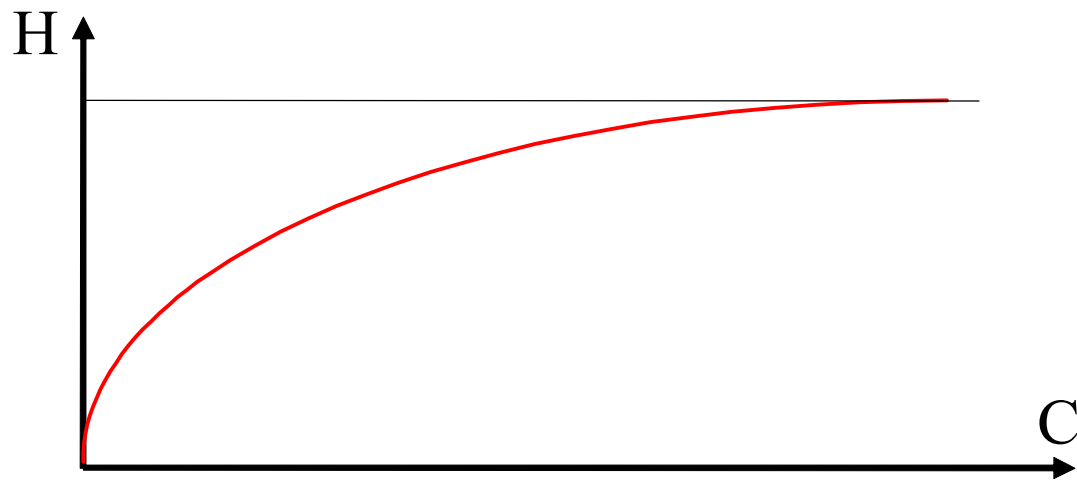
$$t_a = ht_c + (1 - h)t_m$$

■ 访问效率= t_c / t_a

■ 影响命中率的几个因素

- 程序行为（局部性） cache容量
- 组织方式 块大小

cache的命中率与cache容量的关系



块容量与命中率 Is bigger always better?

- 块越大，块数量越少

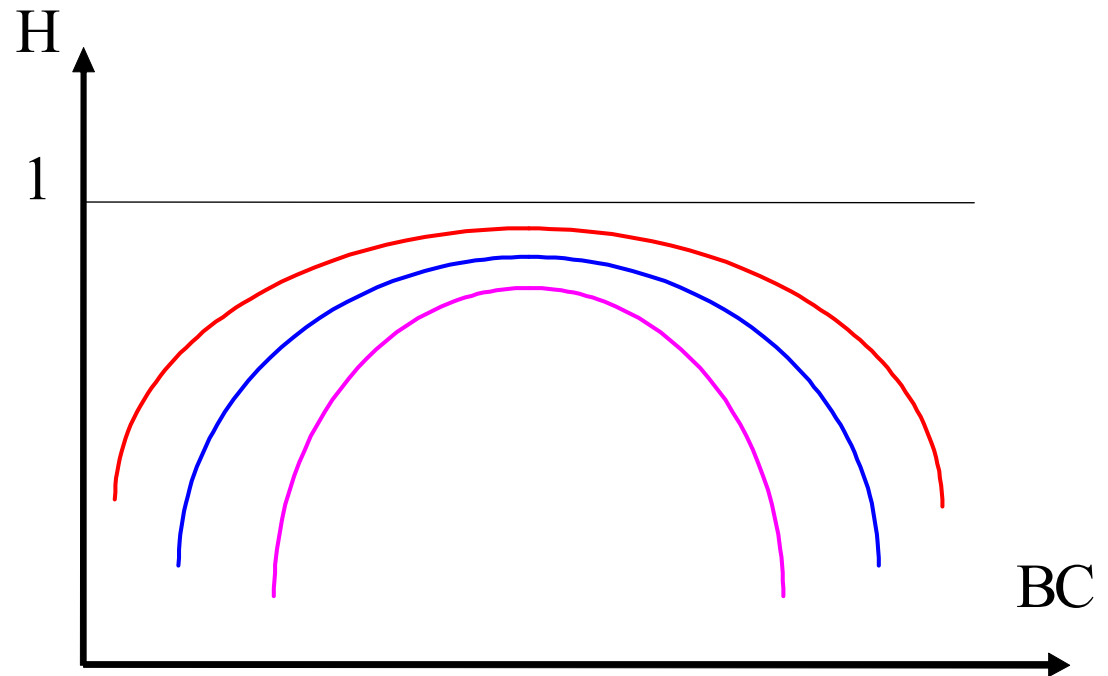
- 空间局部性越好

- 时间局部性不佳

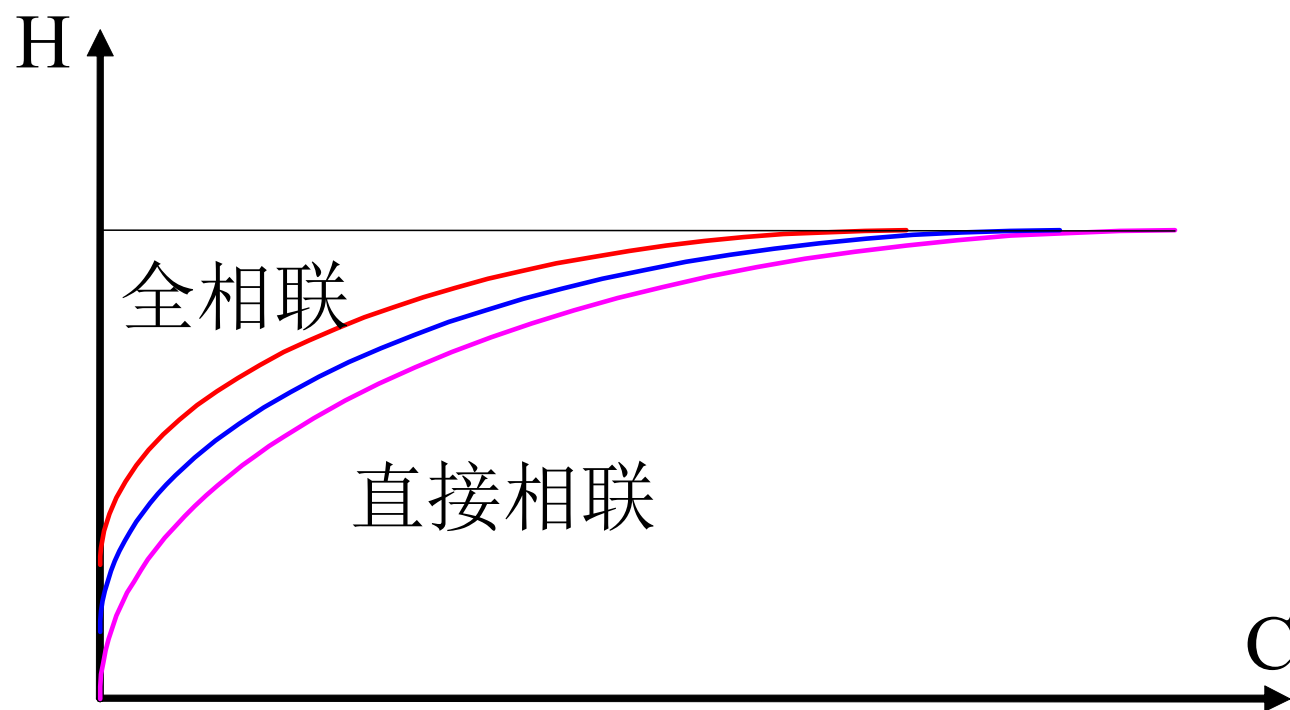
- ◆ 缺失率提升

- ◆ 极端例子

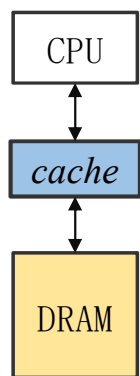
- 一个cache只有一块



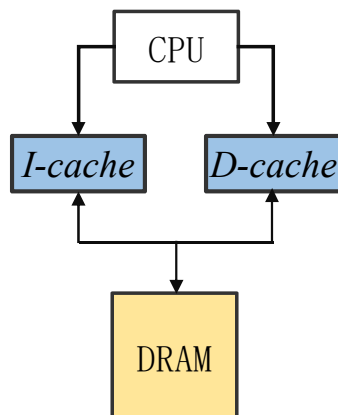
地址映射与命中率



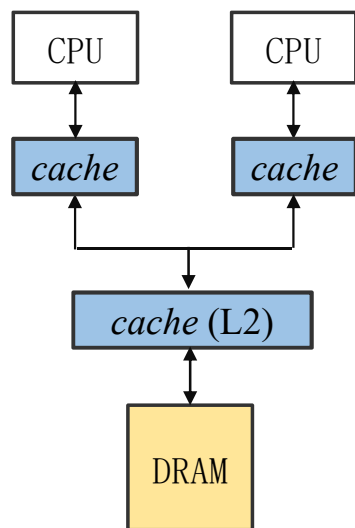
cache实际应用



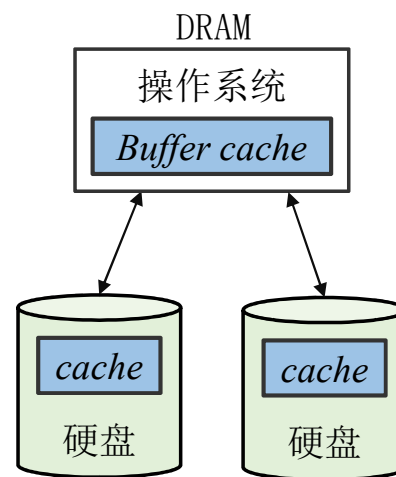
(a) 统一cache



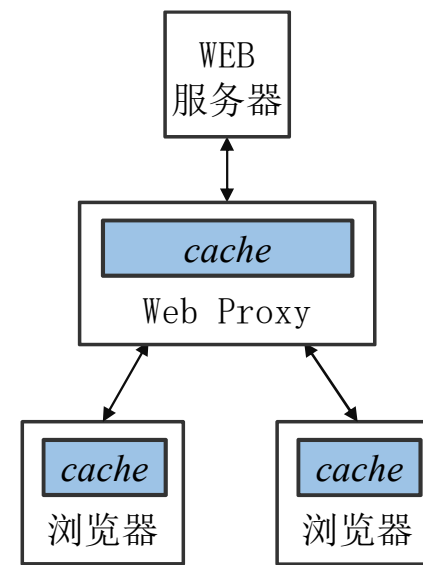
(b) 分离cache



(c) 多级cache



(d) Buffer cache和硬盘缓存芯片



(e) WEB cache

块设备缓存

- OS为优化磁盘等慢速块设备在内存中设置缓存
- 将常用数据存放在高速缓存中，提高磁盘访问速度
- 通过预读等提高命中率的方式可减少读访问时间
- 通过写回策略减少写访问时间
 - CPU将数据写入cache即完成写操作
 - 提升突发写性能
 - 仅写入cache的数据称为脏数据
 - 操作系统定时将脏数据写入磁盘保证数据一致性
 - 当系统中存在脏数据，磁盘断电???

硬盘缓存

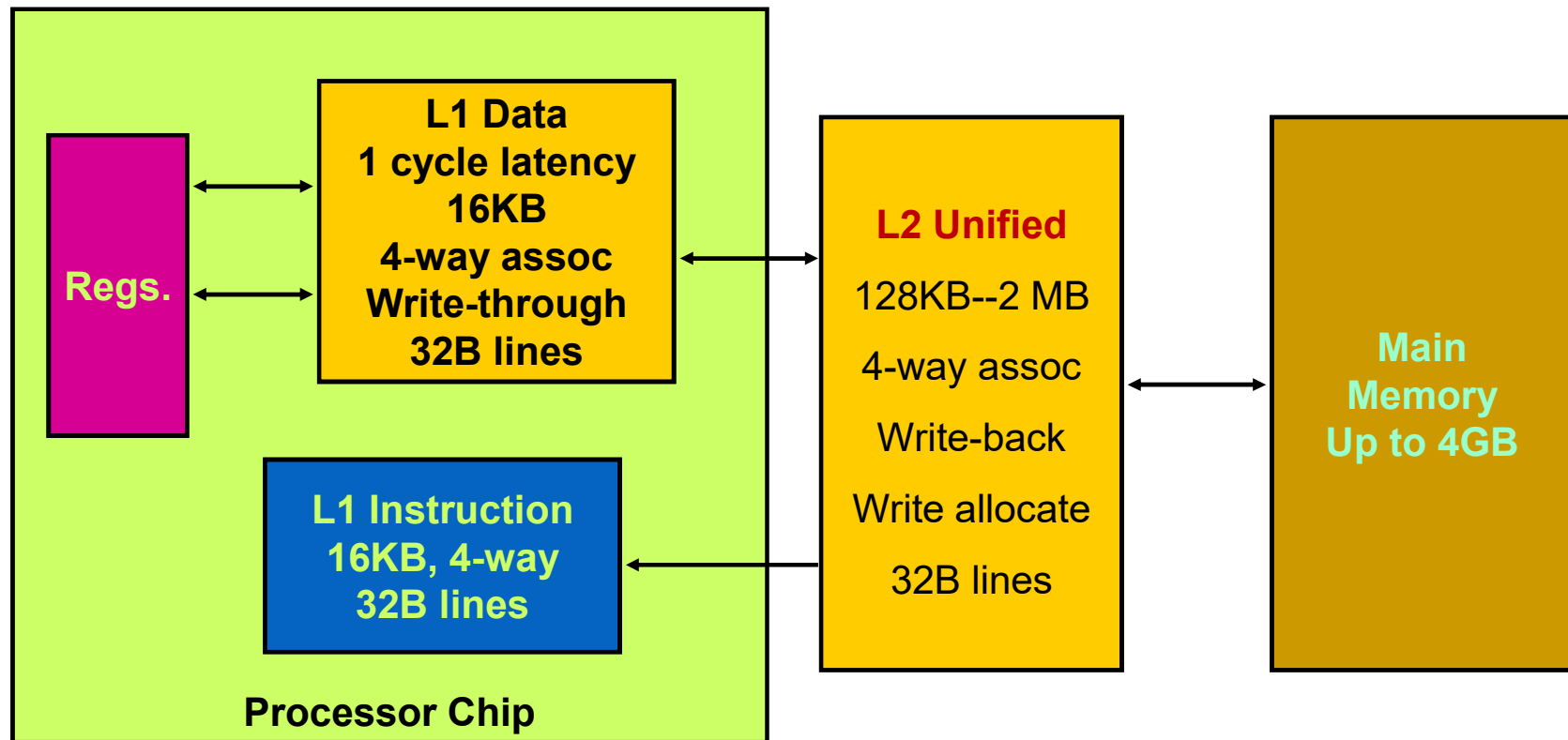
- 硬盘缓存是硬盘上内存芯片
 - 预读硬盘数据，写缓冲CPU数据
 - 临时存储最近访问的硬盘数据
- 缓解硬盘与CPU、内存的速度不匹配
- 缓存容量越大，硬盘访问速度越快
 - 副作用？



WEB cache

- **www.cache.edu.cn proxy.edu.cn :3128**
- 将用户常访问的web内容暂存在离用户近的地方
- 其他用户再次访问这些内容时可以快速获得信息，缩短了响应时间，从而提高访问速度和降低国际通信费用。
- 应有机制保障用户每次访问的数据不会过时

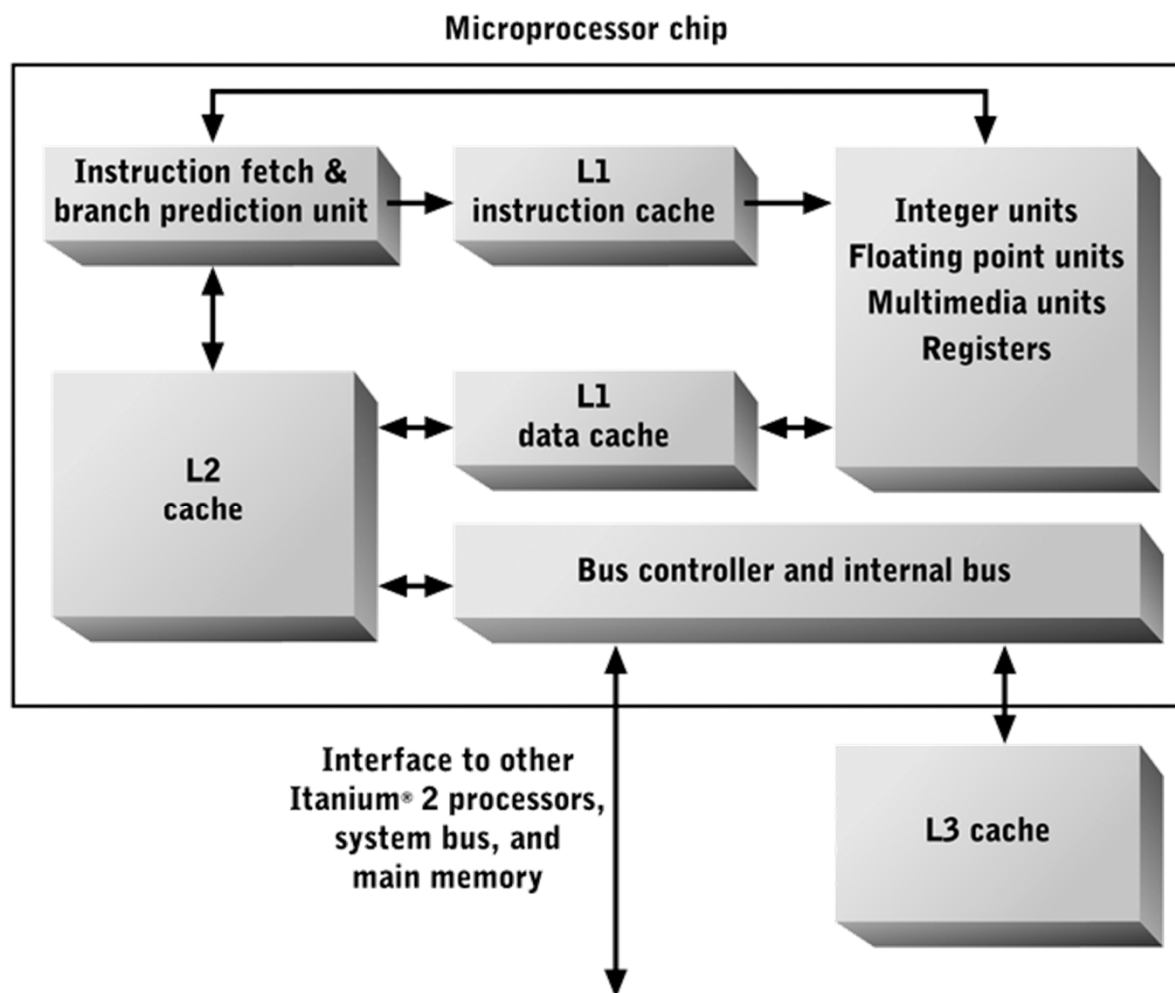
Intel Pentium cache hierarchy



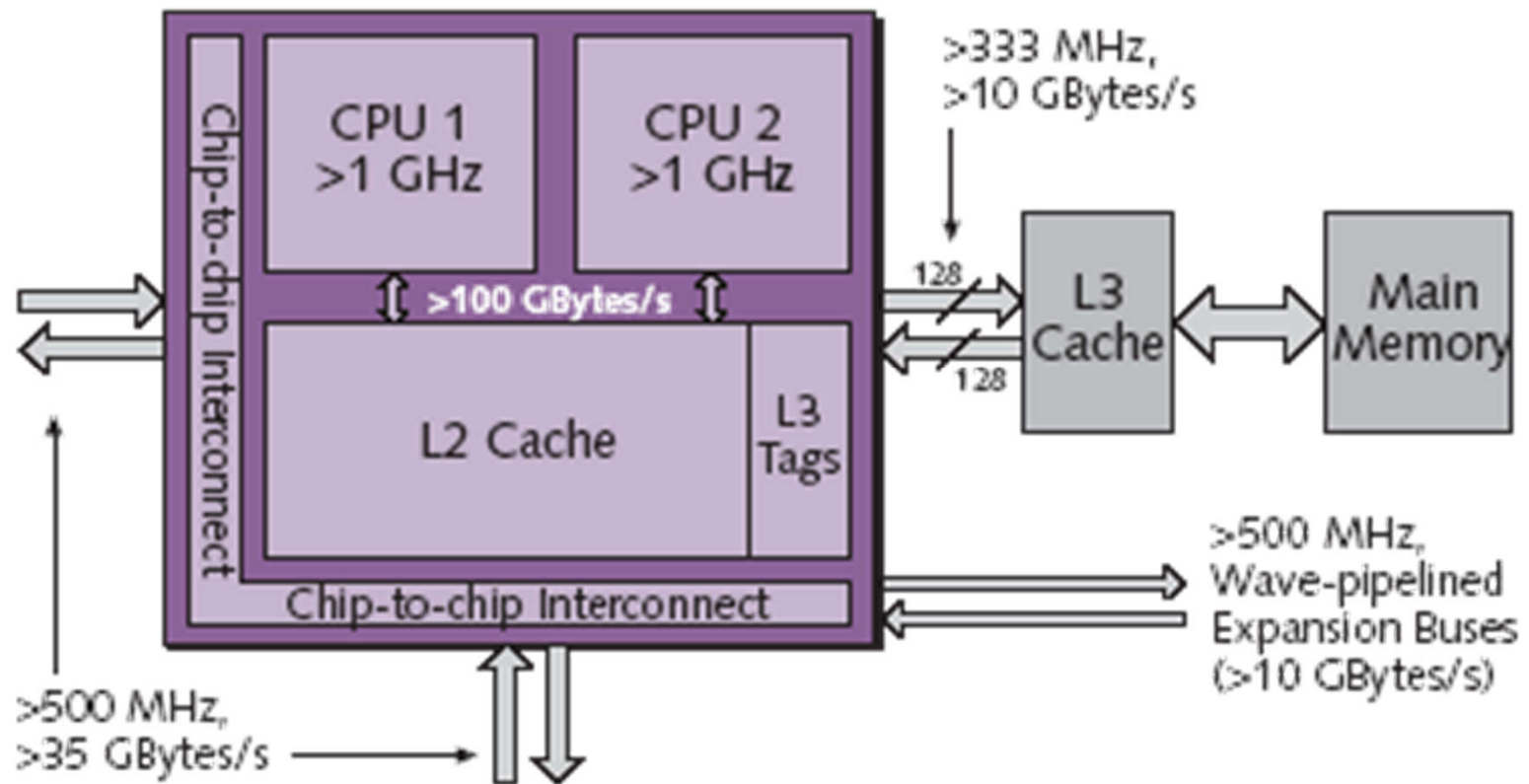
Intel Itanium® 3级Cache

Figure 6-11 ►

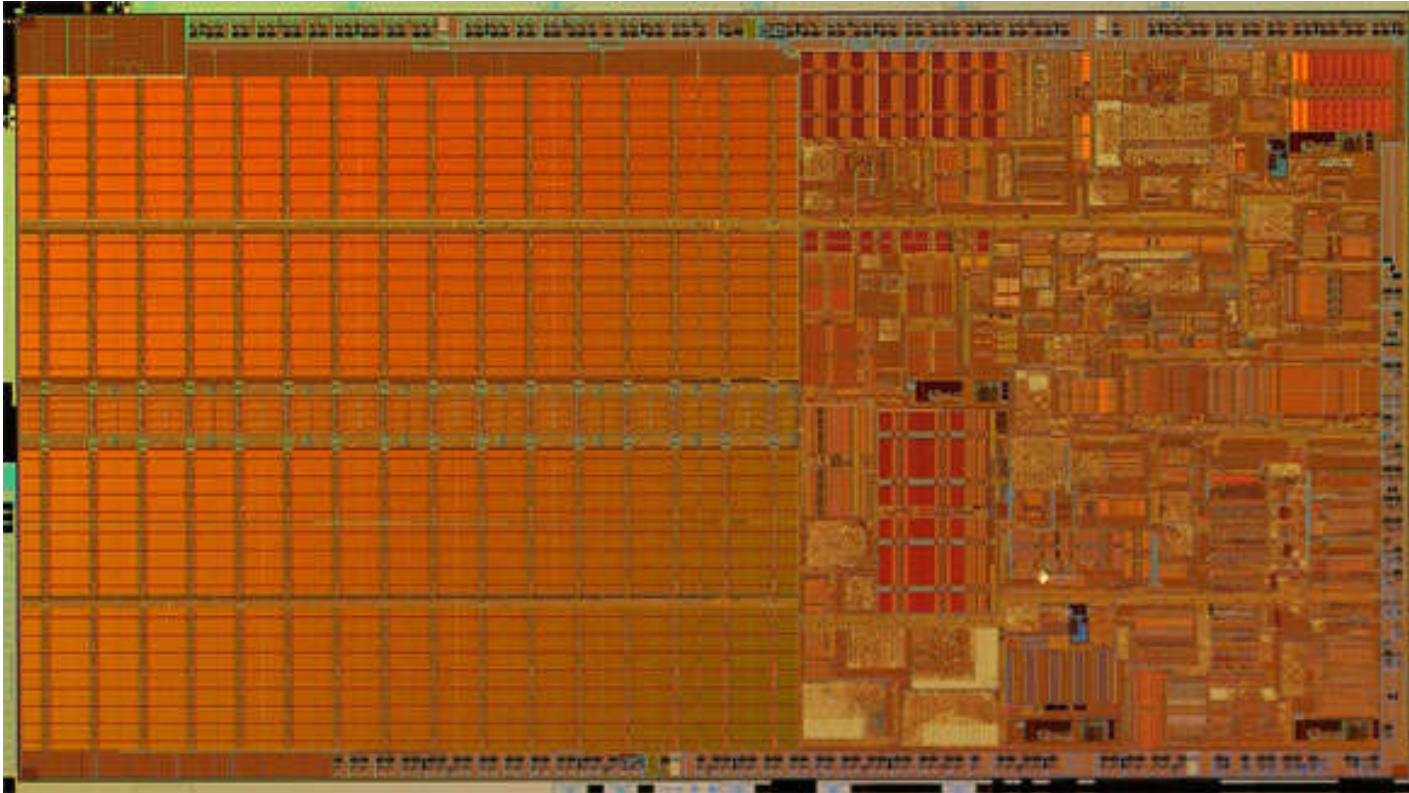
Memory caches
in the Intel
Itanium® 2
processor



IBM Power4 , 1999

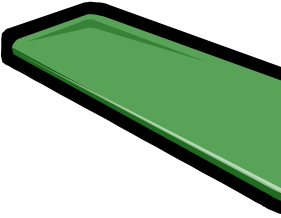


|| But What About More “Typical” Processors



Intel 90 nm – Pentium M Processor (2 MB cache) – Tom' s Hardware

Intel Pentium 4, 2.2 GHz Processor



Component	Access Speed	Size of Component
Registers	1 cycle = 0.5 ns	32 寄存器
L1 Cache	3 cycles = 1.5 ns	指令数据cache分离, 各 8 KB
L2 Cache	20 cycles = 10 ns	256 Kbytes, 8路 组相联
L3 Cache	30 cycles = 15 ns	512 Kbytes 8路 组相联
Memory	400 cycles = 200 ns	16 Gigabytes

Examples of caching in the hierarchy

Cache Type	What Cached	Where Cached	Latency (cycles)	Managed By
Registers	4-byte word	CPU registers	0	Compiler
TLB	Address translations	On-Chip TLB	0	Hardware
L1 cache	32-byte block	On-Chip L1	1	Hardware
L2 cache	32-byte block	Off-Chip L2	10	Hardware
Virtual Memory	4-KB page	Main memory	100	Hardware+OS
Buffer cache	Parts of files	Main memory	100	OS
Network buffer cache	Parts of files	Local disk	10,000,000	AFS/NFS client
Browser cache	Web pages	Local disk	10,000,000	Web browser
Web cache	Web pages	Remote server disks	1,000,000,000	Web proxy server

|| 本章主要内容

- 存储器概述
- 主存储器
- 高速缓冲存储器
- **虚拟存储器**
- 外存储器



虚拟存储器

- 比尔·盖茨(1981) “无论对谁来说,640K内存都足够了”
- 如何在有限的主存空间运行较多的较大的用户程序?
 - 初衷: 采用虚存来扩大寻址空间
 - 现在: 主要用于存储保护, 程序共享
- 主存、辅存在OS和硬件的管理之下, 提供比实际大得多的存储空间
- 虚存空间是靠辅存(磁盘)来支持的
- 虚拟存储采用与Cache类似原理, 提高速度, 降低成本
- 用户感觉到的是一个速度接近主存而容量极大的存储器

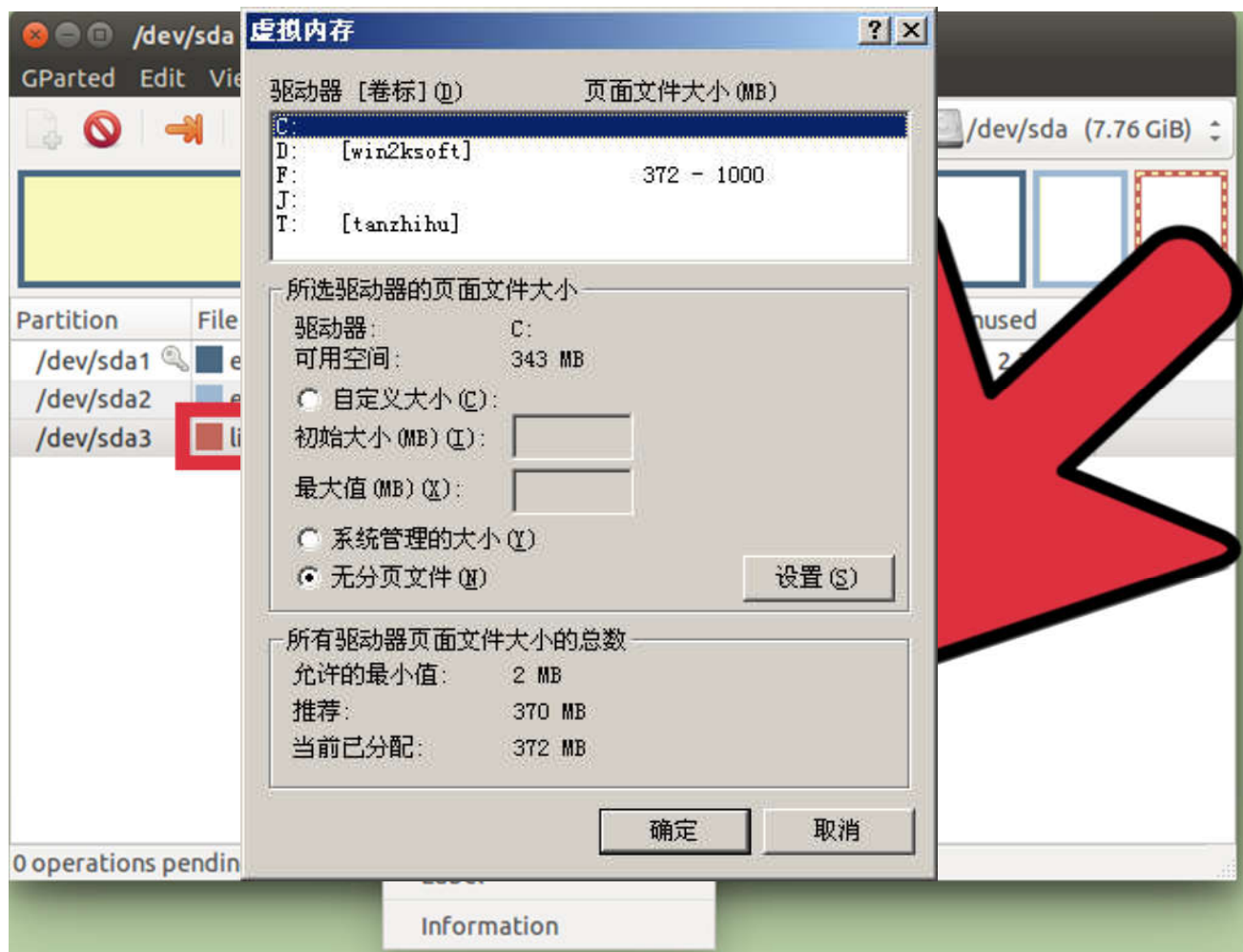
虚拟内存实例

■ Linux

- 交换分区swap

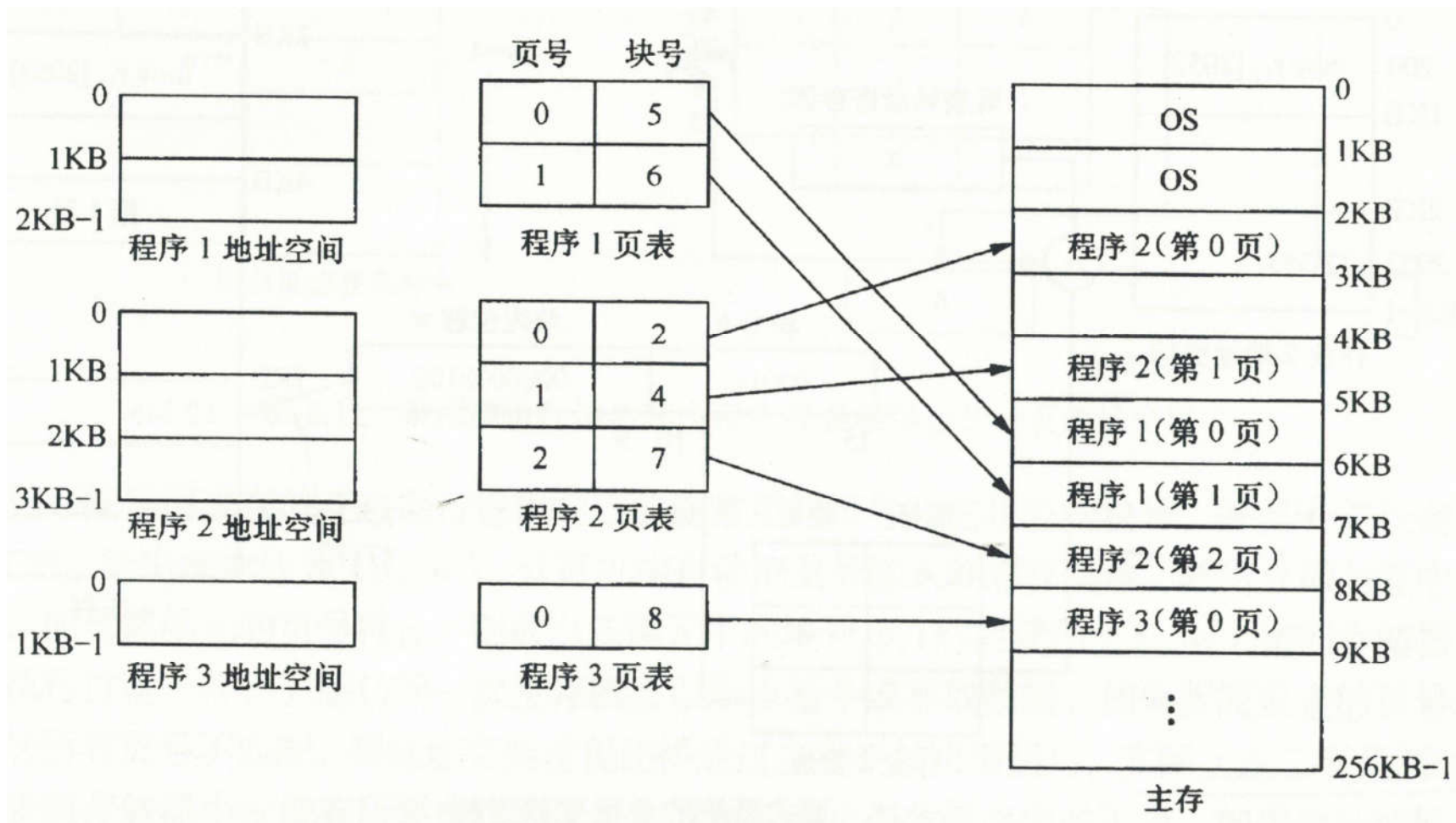
■ Windows

- 页面文件

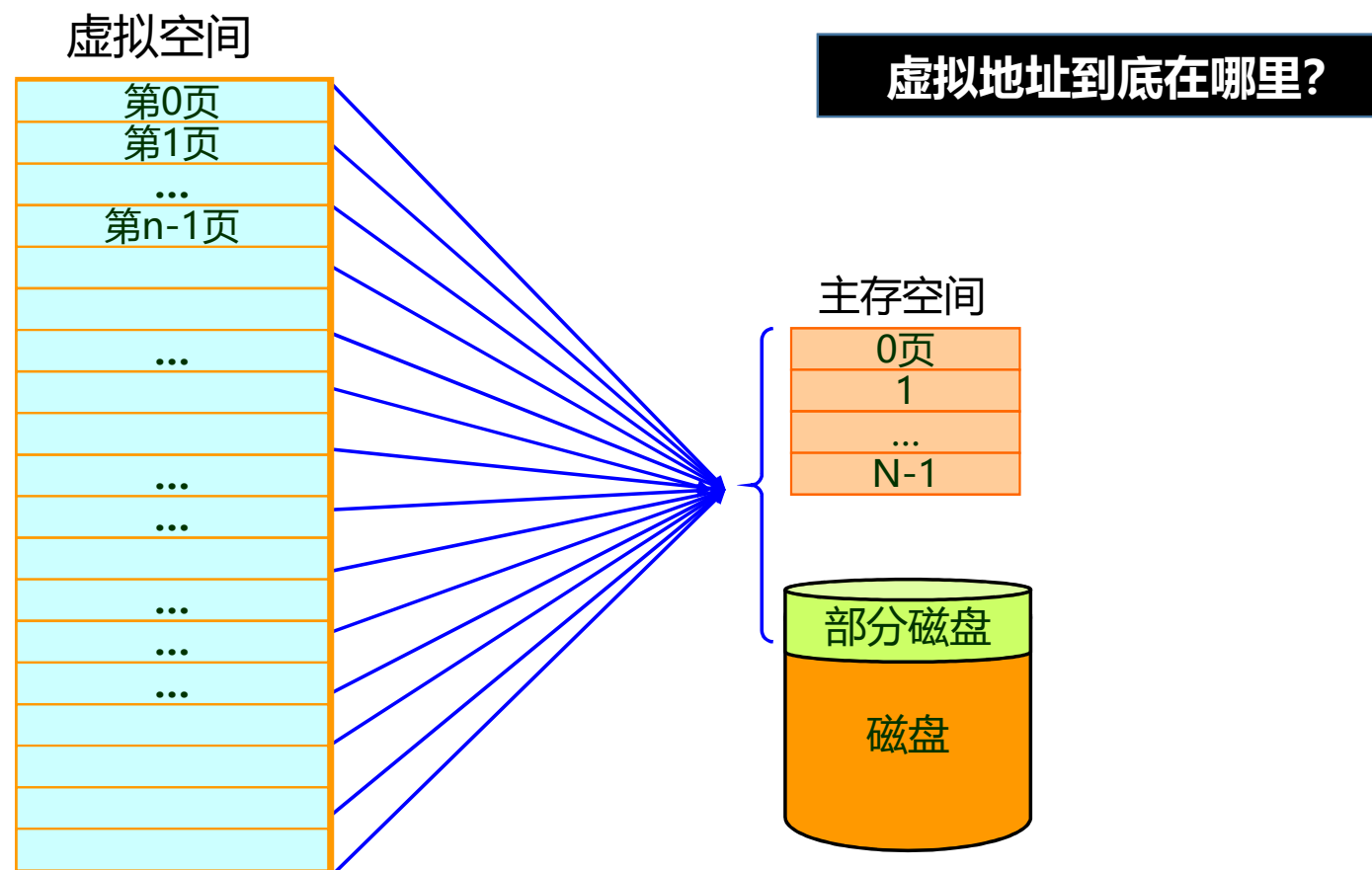


分页映像存储的概念

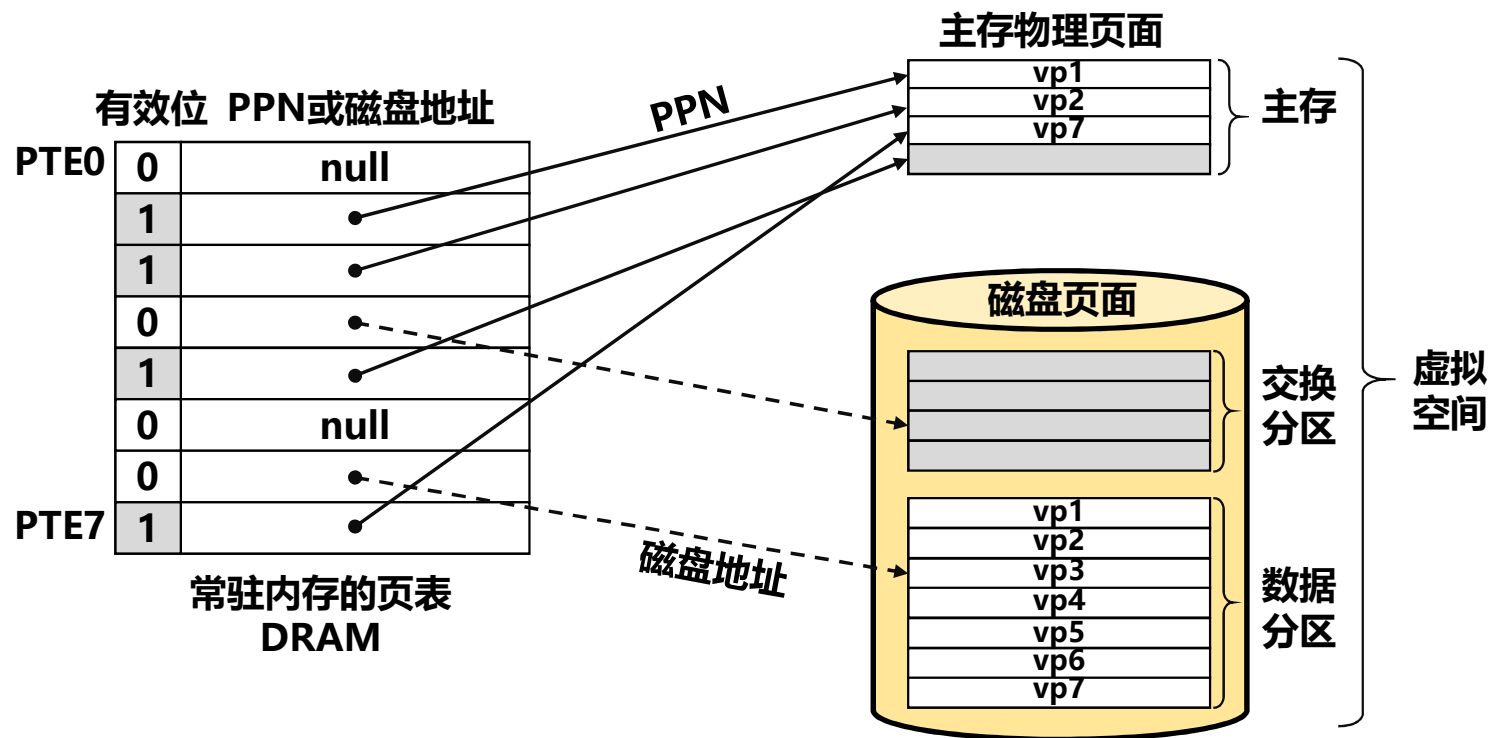
操作系统调度内存的使用，并且可以保护系统内核，避免某个进程破坏其它进程数据



|| 页式虚拟地址空间



Page table (页表, 查找表)

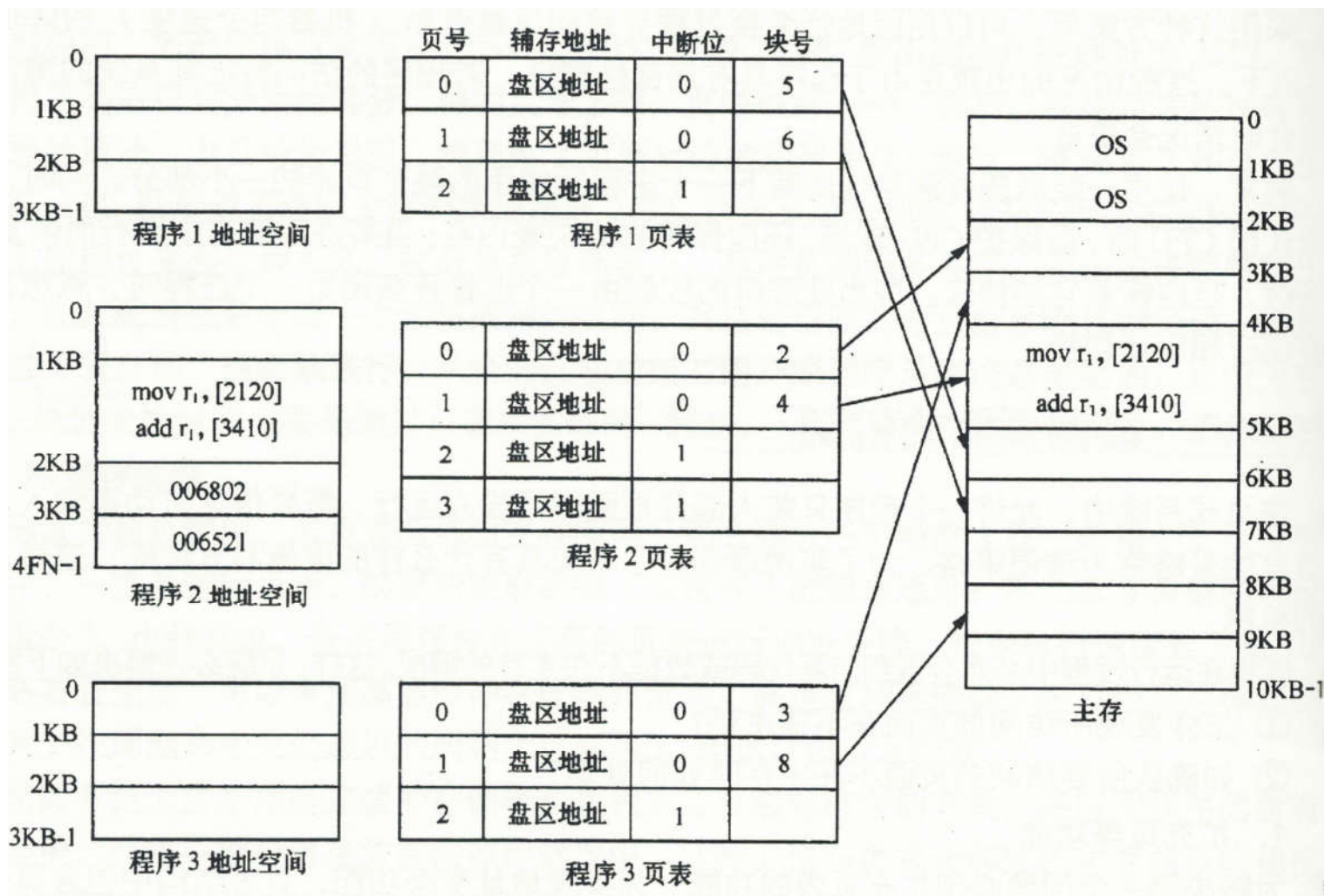


注意：虚线表示缺页

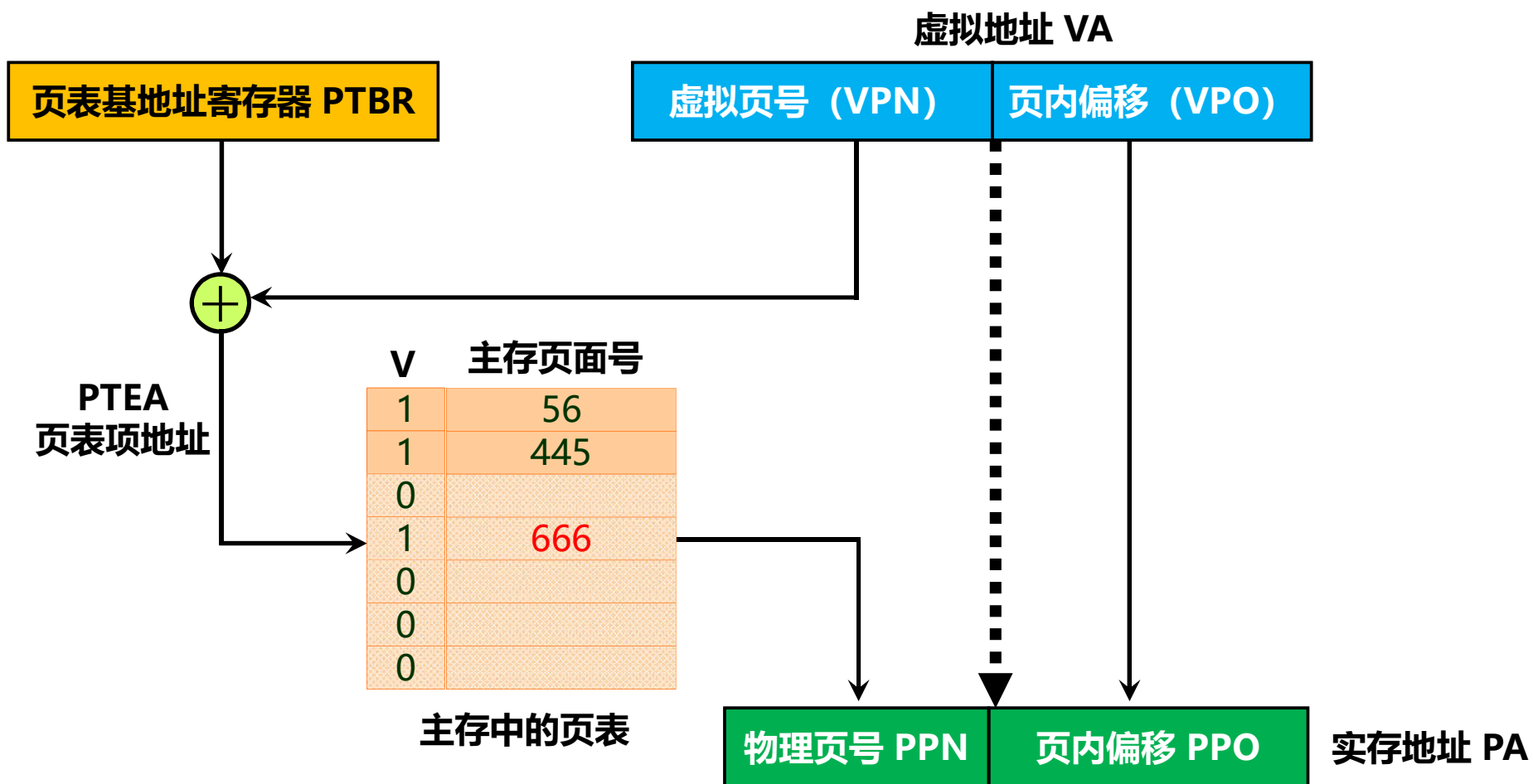
每个进程有一张页表，切换进程只需切换页表首地址

请求分页映像存储举例

中断位=0表示该页在内存，否则触发异常，由OS执行调页操作

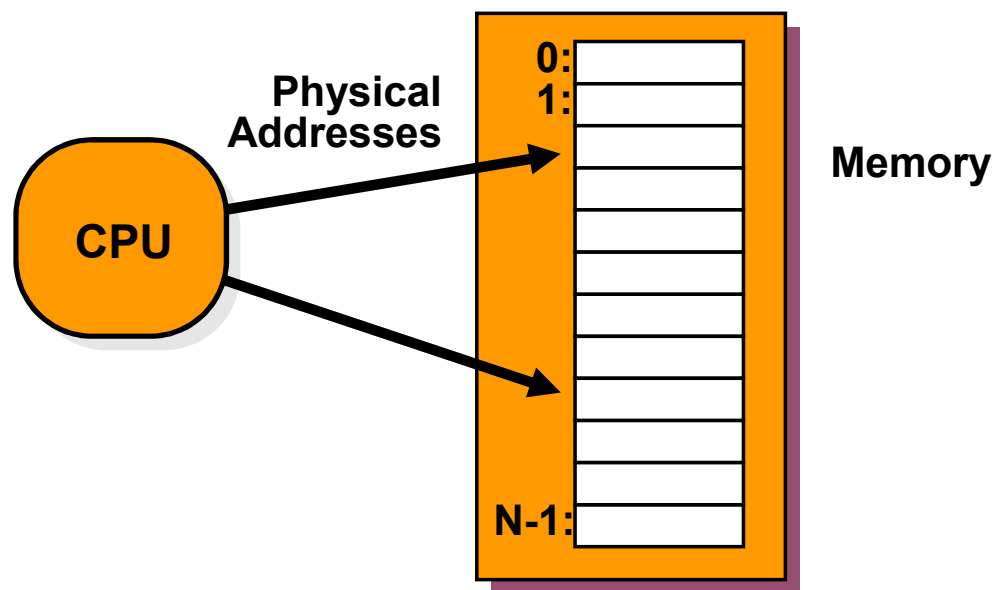


页式虚拟存储器结构



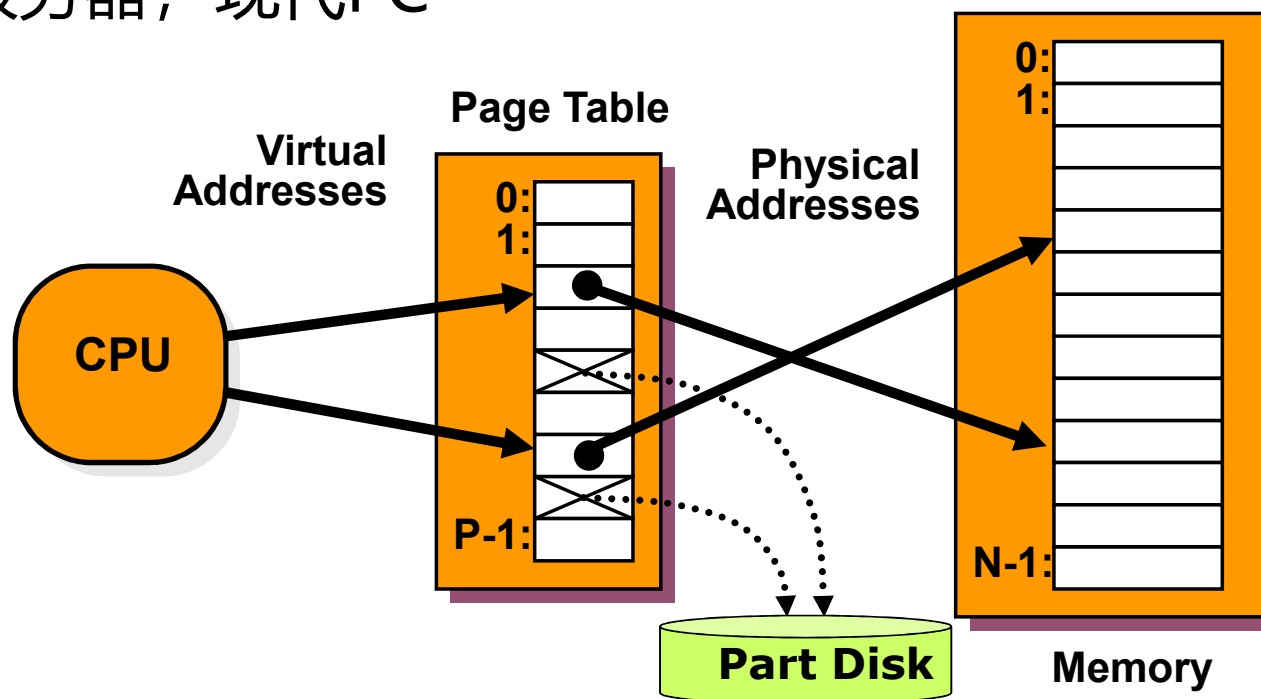
|| 实地址计算机系统

- CPU给出的地址直接访问物理内存
- 大多数Cray计算机，早期PC，大多数嵌入式系统



虚地址计算机系统

- CPU地址需要虚实变换
 - 硬件通过OS维护的页表将虚拟地址转换为物理地址
- 工作站，服务器，现代PC



缺页 Page Faults

■ 页表指示虚拟地址不在内存中

- 操作系统负责将数据从磁盘迁移到内存中
- 当前进程挂起
- 操作系统负责所有的替换策略
- 唤醒挂起进程

