

虚拟存储器



上海交通大学
SHANGHAI JIAO TONG UNIVERSITY



本节



- 回顾：虚拟存储器、页表等概念
- TLB（快表）的作用
- 了解内存地址转换的全过程

虚拟存储器



▪ 问题的提出

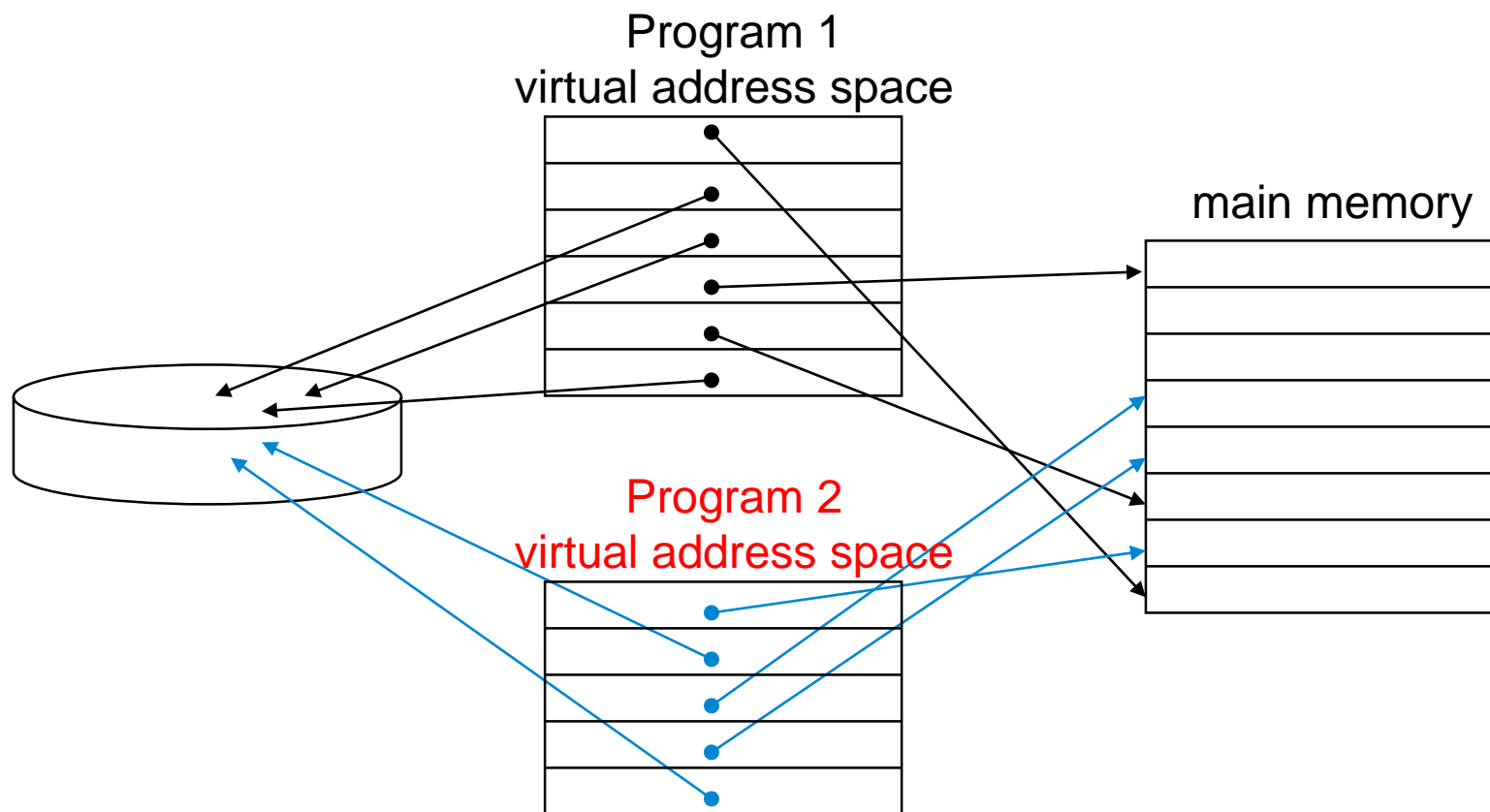
- 程序大于总内存量
- 内存容量无法满足：作业一次性装入和运行时的一直驻留。
- 运行多道程序，内存只能容纳部分作业

▪ 虚拟存储器的思想

- 把程序当前使用的部分代码和数据保留在内存中，把其它部分存在磁盘上，需要在内存和磁盘之间动态交换

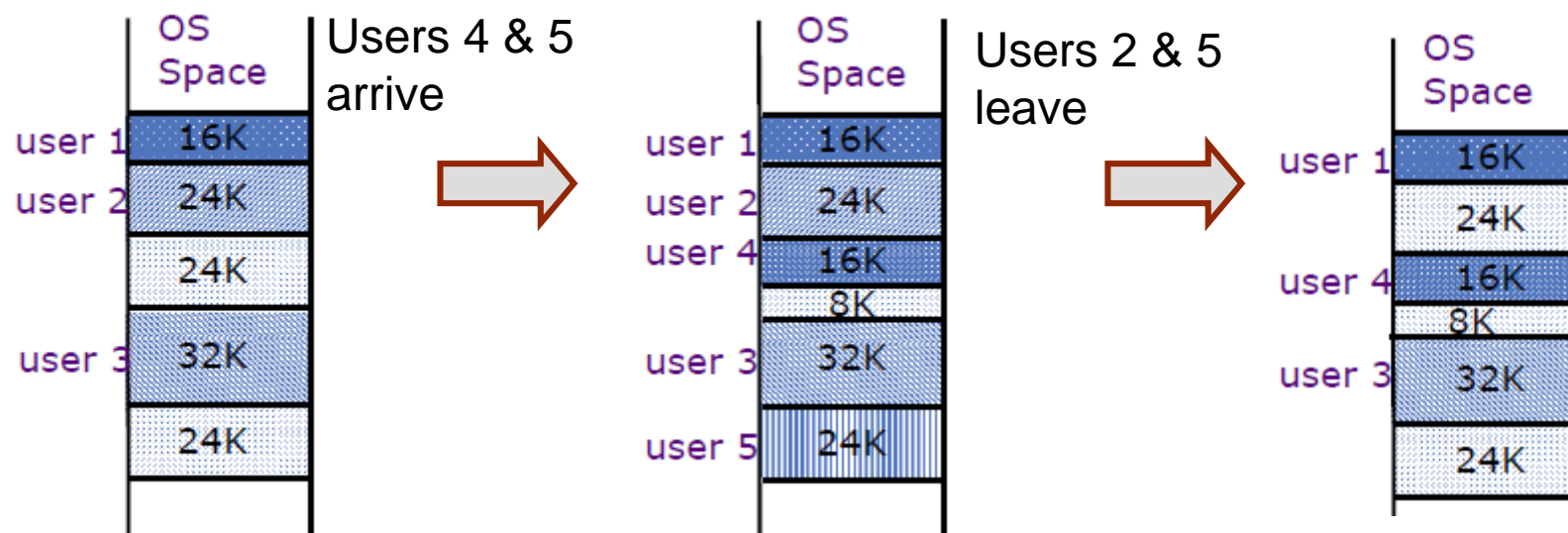
多道程序共享物理内存

- ❑ 一个程序的地址空间可以分成
- ❑ 多个大小相同的页（**pages**：大小固定，一般为**4KB**）



为什么是分页?

避免内存碎片

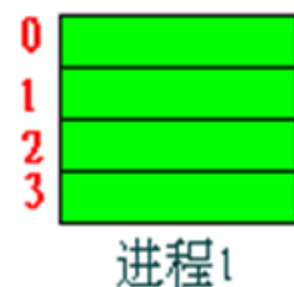


分段: 需要阶段性重新整理内存分配, 合并碎片

请求分页技术



- 当一个用户程序要调入内存时，不是将该程序全部页面装入内存，而是只装入部分页面到内存，就可启动程序运行；
- 在运行的过程中，如果发现要运行的程序或要访问数据不在内存，则向系统发出缺页中断请求，系统在处理这个中断时，将在外存相应的页调入内存，该程序继续运行。
- 为了实现请求分页，系统需要有：
 - (1) 页表机制
 - (2) 地址变换机构
 - (3) 缺页中断机构
 - (4) 页面置换算法



页号 ... 中断 页框架号

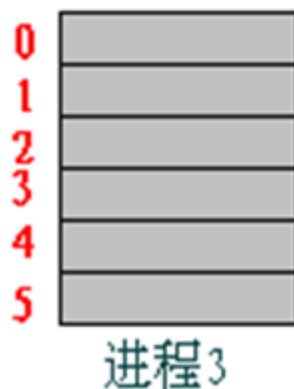
0			1	—
1			1	—
2			0	7
3			0	5

进程1 页表



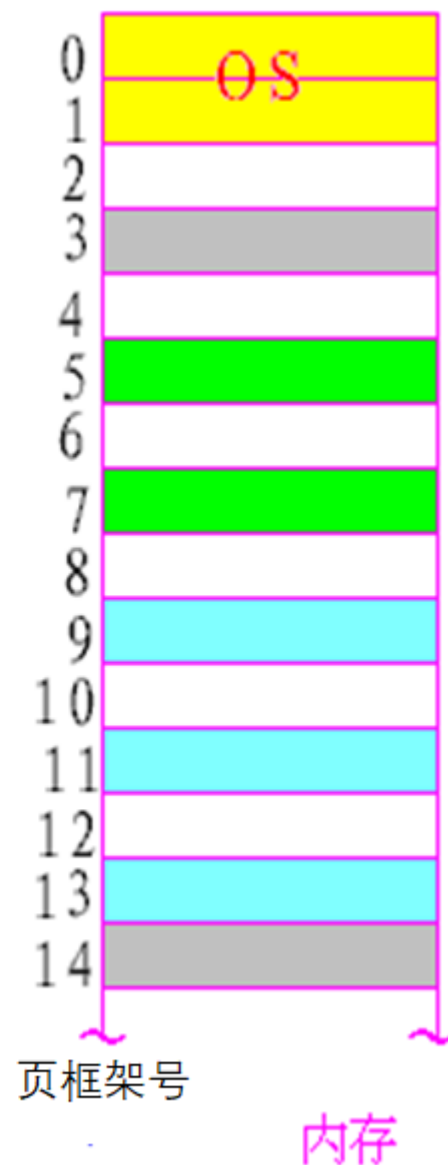
0			1	—
1			0	13
2			1	—
3			0	11
4			0	9

进程2 页表



0			1	—
1			0	14
2			1	—
3			1	—
4			0	3
5			1	—

进程3 页表



(1) 页表机制

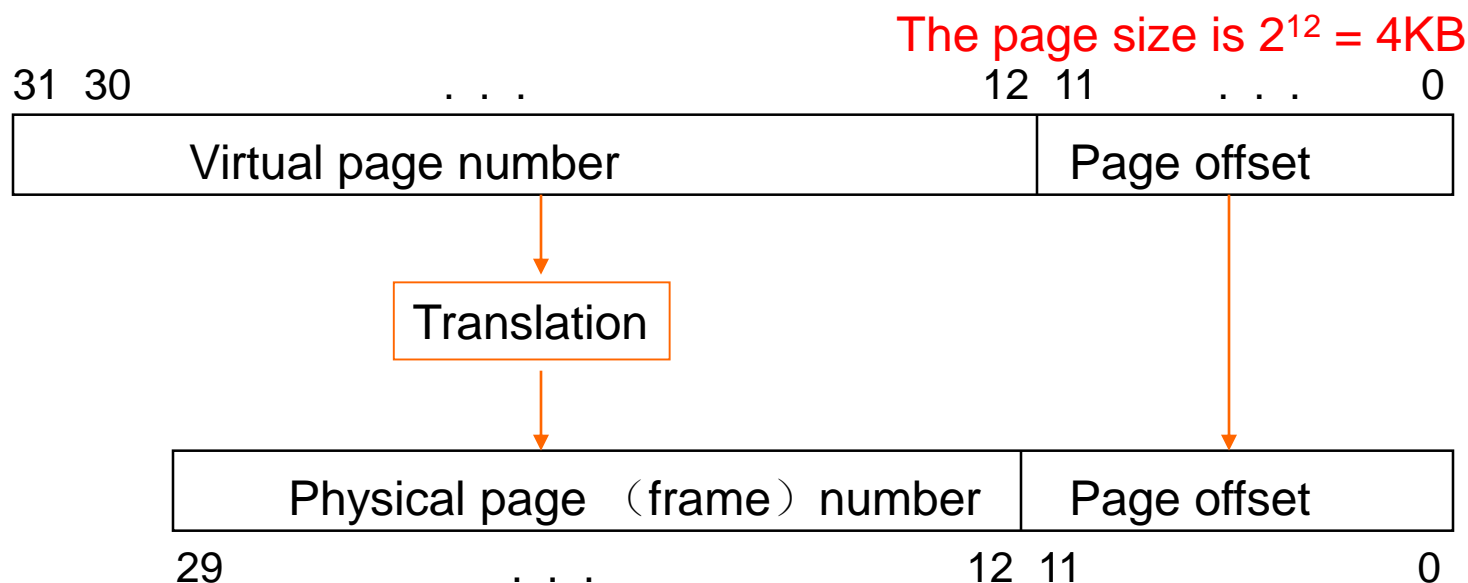


页号	页框架号	状态位P	访问字段A	修改位M	外存地址
----	------	------	-------	------	------

- **状态位P**: 用于指示该页是否已调入内存，供程序访问时参考；
- **访问字段A**: 用于记录本页在一段时间内被访问的次数，或记录本页最近已有多长时间未被访问，供选择置换页面时参考；
- **修改位M**: 表示该页在调入内存后是否被修改过，供置换页面时参考；
- **外存地址**: 用于指示该页在外存上的地址，供调入页时使用。

(2) 地址转换策略

虚拟地址 Virtual Address (VA) : 4GB



1GB 物理地址 Physical Address (PA)

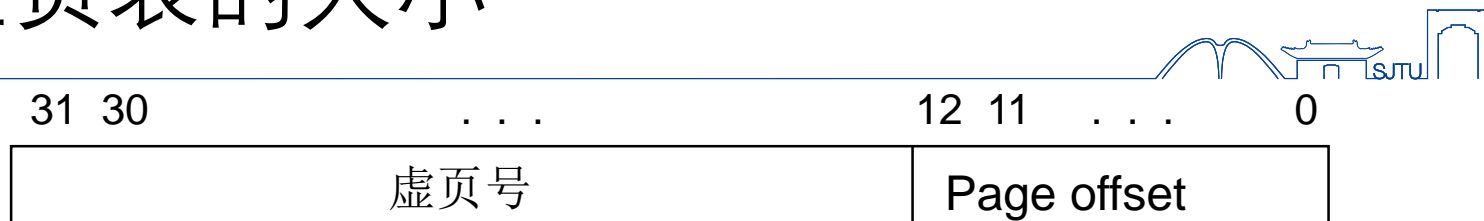
- 每一次访存请求，都需要查询页表，将虚页号转化为物理页框架号
- 如果某个虚拟地址无法转换为物理地址，称为：页面失效 (page fault)

页表驻留在哪里？



- 页表的长度：
 - 等比于地址空间
 - 等比于进程数目
 - 需要空间很大
 - 无法放置在寄存器中
- 页表驻留在内存里

线性页表的大小



- 32位虚拟地址, 4KB 页面长度 & 4byte 页表项! 
 - 页表项的个数: 2^{20} 个,
 - 页表的大小: $2^{20} \times 4\text{byte} = 4 \text{ MB}$
- 48位虚拟地址, 4KB 页面长度 & 8byte 页表项 (PTE)
 - 页表项的个数: 2^{36} 个,
 - 页表的大小: $2^{36} \times 8 \text{ byte} = 2^{39} \text{ bytes} = 512 \text{ GB}$

所以: 页表一般不是线性的

多级页表

- 解决方法: 多级页表

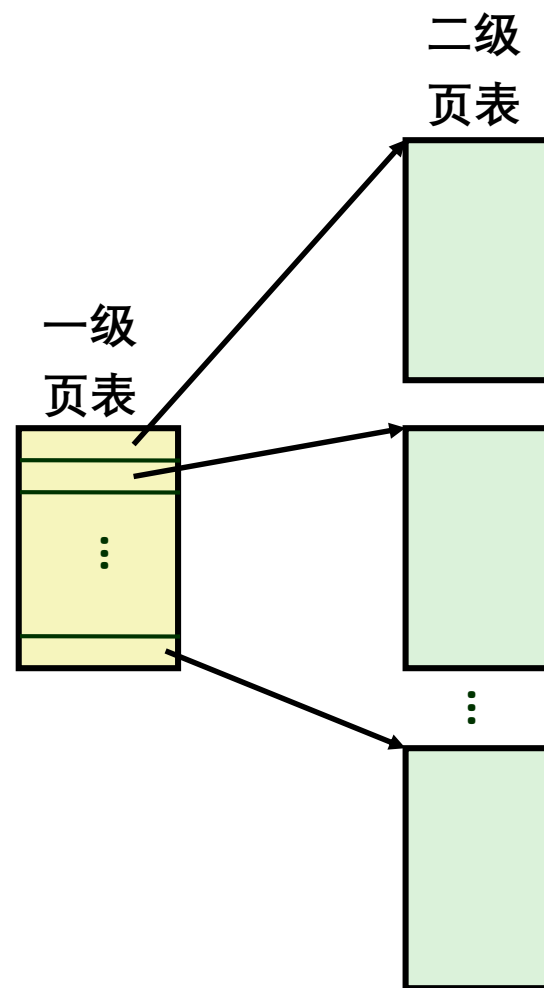
- 举例: 二级页表

- 第一级页表:

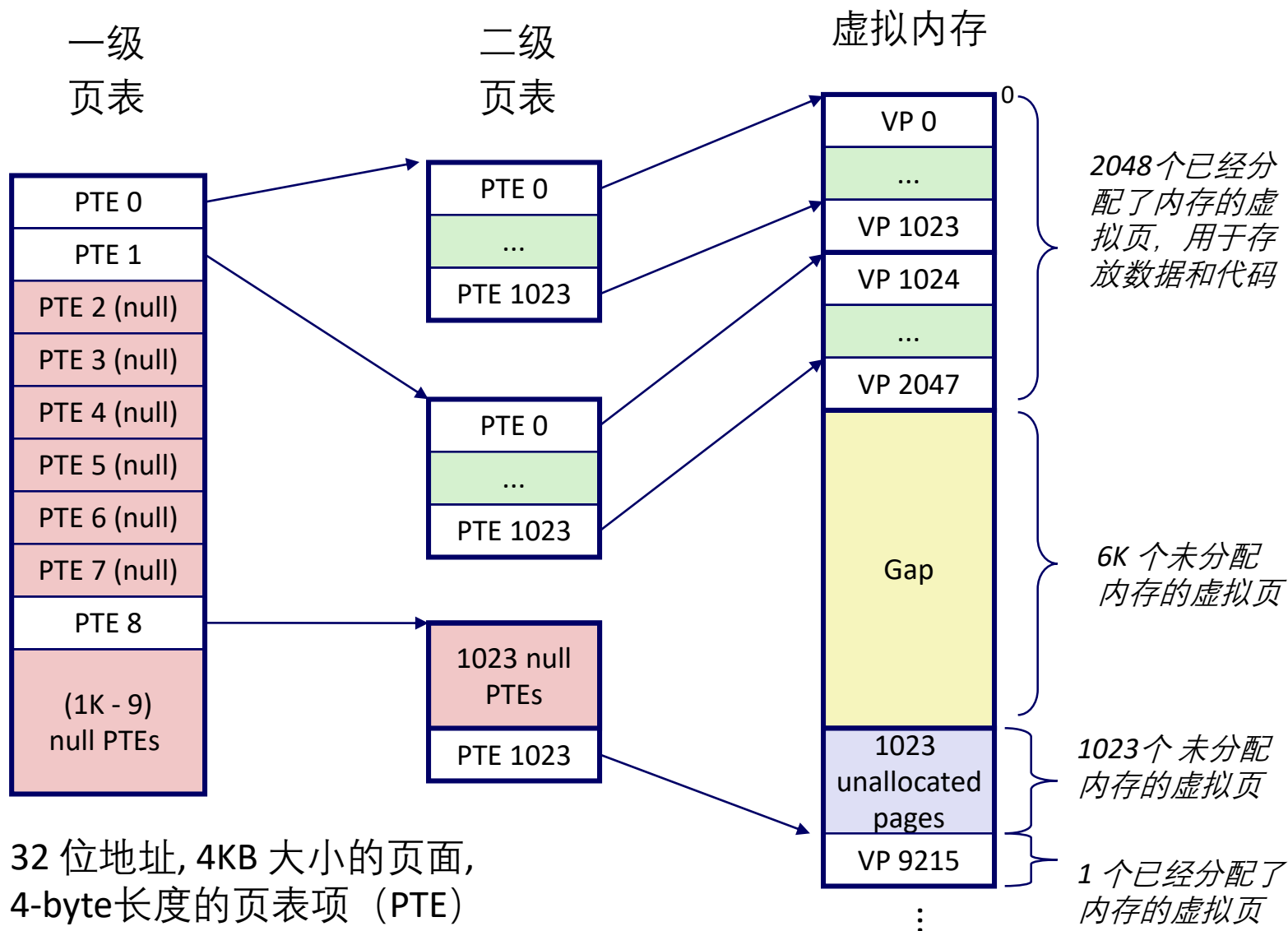
- 每一项指向一个页表,
 - 记录页表所在位置

- 第二级页表:

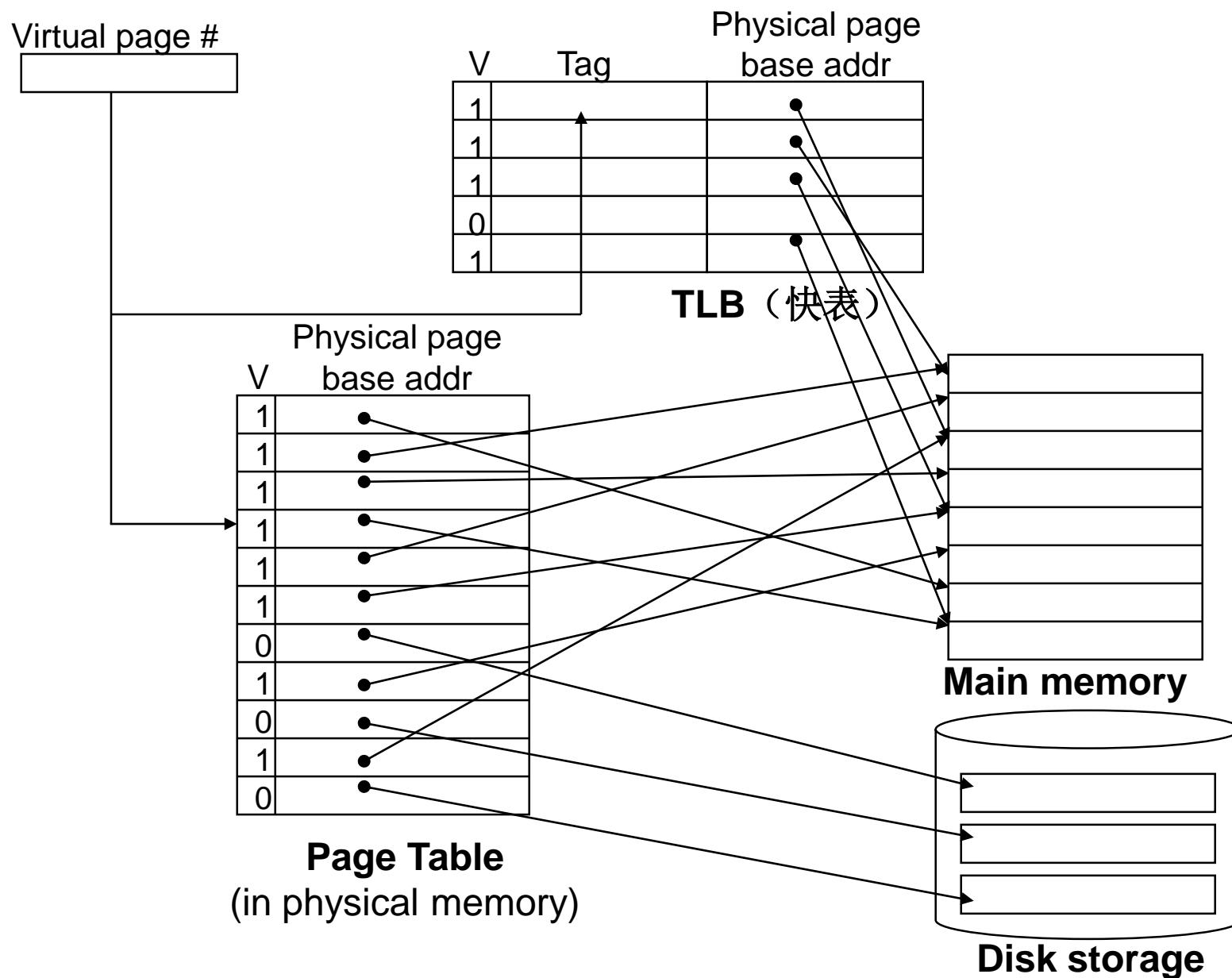
- 每一项指向一页
 - 记录页所在位置



一个二级页表的层次结构



利用TLB (Translation Lookaside Buffers) 加快地址转换



Translation Lookaside Buffers (TLBs)

- 和Cache一样，TLB 可以组织为：**fully associative（全相联）**，set associative（组相联），or direct mapped（直接相联）

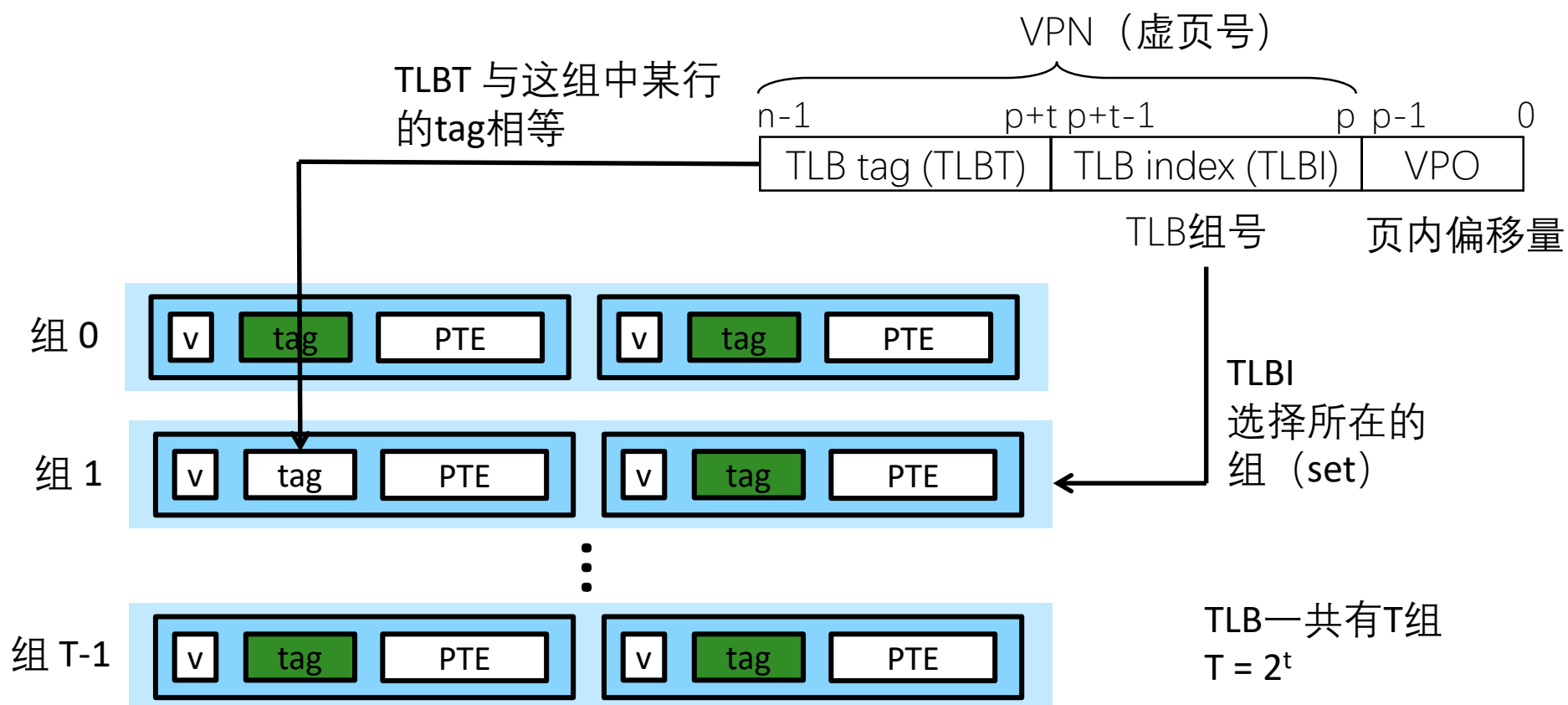
一个全相联的TLB

V	Virtual Page #	Physical Page #	Dirty	Ref	Access

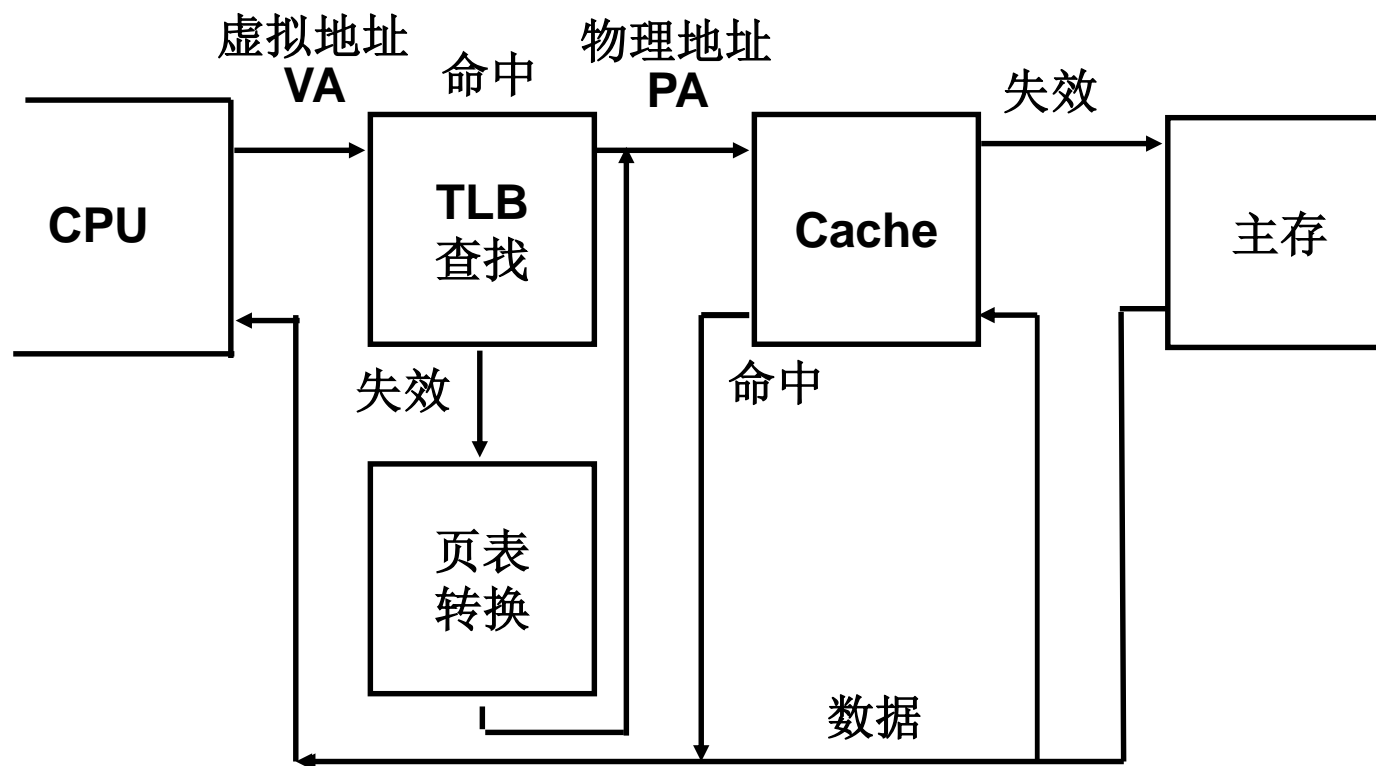
- TLB 访问时间一般小于 cache 访问时间(因为 TLBs 比 caches 小很多)

一般不多于 128 到 256 项（即使高端机器中）

组关联映射 (set associative) TLB



TLB 在地址变换过程中的作用



页面失效



- TLB 失效
 - 如果本页已经载入内存，就仅仅是一次TLB miss
 - 只需要将信息从页表填入TLB
 - 一般花费 10个周期
- 页面失效（page fault）
 - 本页没有载入内存
 - 需要从磁盘调页
 - 一般花费1,000,000个周期来处理页面失效

TLB 事件组合

TLB	页表	Cache	可能发生吗？ 在何种情况下？
Hit	Hit	Hit	
Hit	Hit	Miss	
Miss	Hit	Hit	
Miss	Hit	Miss	
Miss	Miss	Miss	
Hit	Miss	Miss/ Hit	
Miss	Miss	Hit	

TLB 事件组合

TLB	Page Table	Cache	可能发生吗？ 在何种情况下？
Hit	Hit	Hit	是的 – 我们希望这种情况发生!
Hit	Hit	Miss	是的– TLB 命中，不需要访问页表，但数据没有调入缓存，只能从内存读取
Miss	Hit	Hit	是的 – TLB 失效, 地址转换通过查页表获得
Miss	Hit	Miss	是的 – TLB 失效, 地址转换通过查页表获得，但数据没有调入缓存，只能从内存读取
Miss	Miss	Miss	是的 – 页面失效
Hit	Miss	Miss/ Hit	不可能 – 如果这页没有调入内存， TLB 不可能命中
Miss	Miss	Hit	不可能 –如果这页没有调入内存，就不可能存放在 cache 中

Summary

- 页表实现虚拟地址到物理地址的映射
- TLB可以加快虚实地址转换的速度
- 了解内存地址转换的全过程

谢谢！

