《计组II》

第七章——存储系统

李瑞　副教授

蒋志平　讲师

计算机科学与技术学院

Cache/内存一致性问题

7.3.3 主存与Cache内容的一致性问题

采用Cache的访存操作

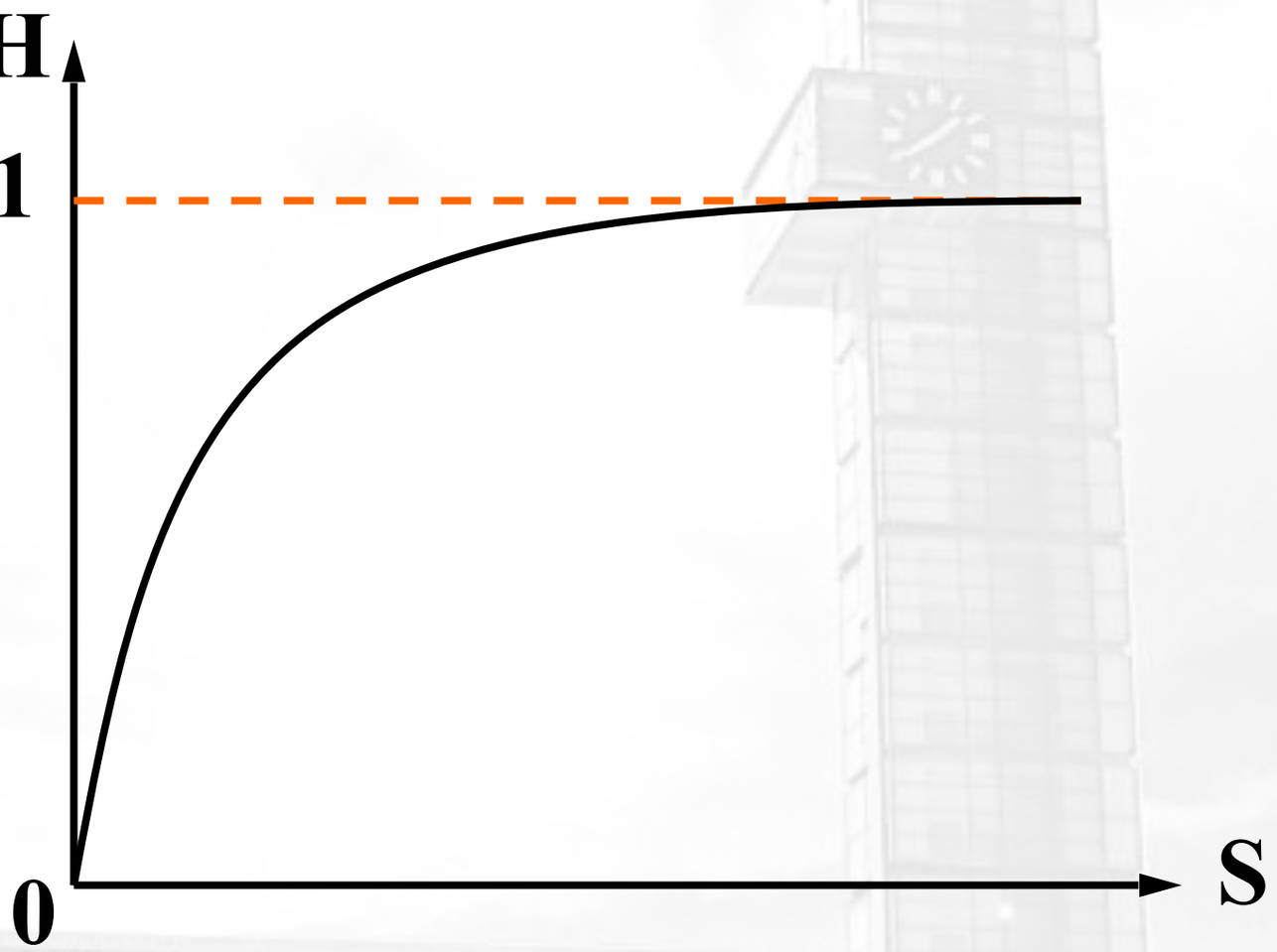
Cache类型	操作	访存操作	访存时间
写直达Cache	读命中	只读Cache	T_C
	写命中	写Cache，同时写内存	T_C(隐藏T_M)
	读不命中	调入Cache块，再读Cache	$T_B + T_C$
	写不命中	只写内存	T_M
写回Cache	读命中	只读Cache	T_C
	写命中	只写Cache	T_C
	读不命中	调入Cache块，再读Cache	$T_B + T_C$
	写不命中	调入Cache块，再写Cache	$T_B + T_C$

Cache性能分析

7.3.4 Cache性能分析

3. 命中率与Cache容量的关系

$$H = 1 - S^{-0.5}$$

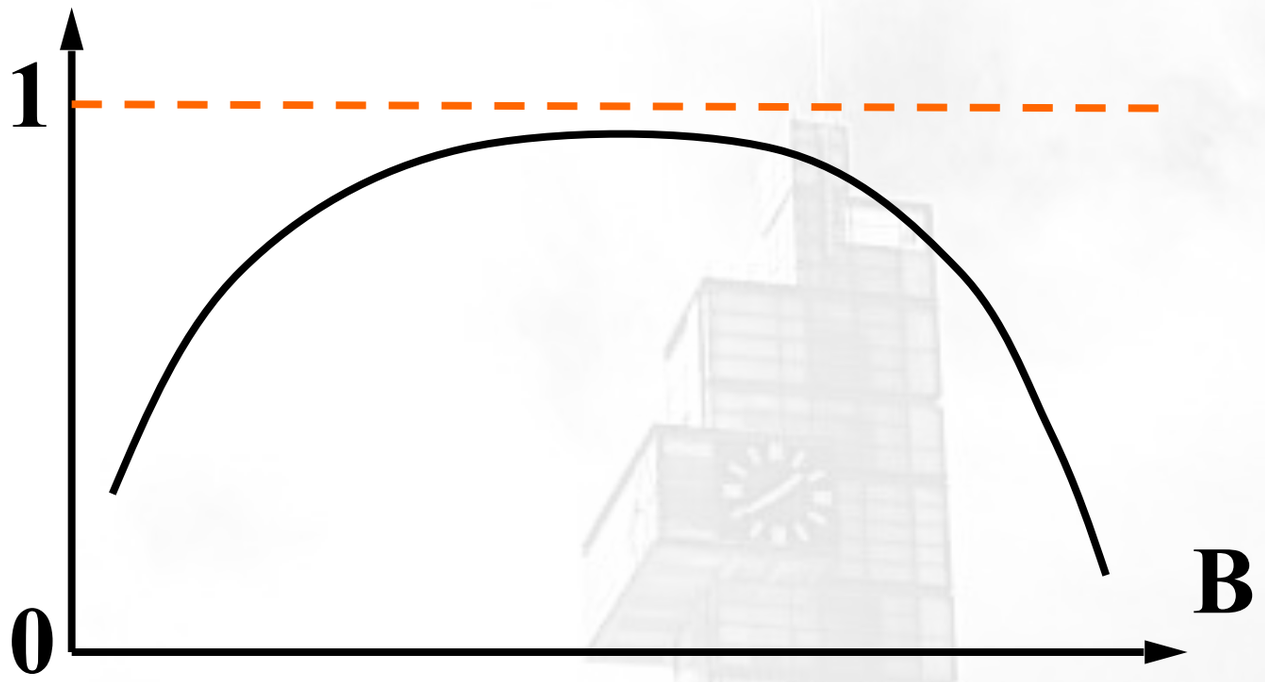


Cache容量S与命中率H的关系

7.3.4 Cache性能分析

4. 命中率与块大小的关系

在组相联组Cache容量一定时，块大小B与命中率的关系?



块大小B与命中率H的关系

假设A_t和A_{t+1}是相邻两次访问主存的地址，

$d = |A_t - A_{t+1}|$。

- 如果$d < B$，随着B的增大，A_t和A_{t+1}在同块的可能性增加，即H随着B的增大而提高。
- 如果$d > B$，A_t和A_{t+1}一定不在同一块内。随着B的增大，Cache块数减少，块替换将更加频繁。H随着B的增大而降低。

7.3.4 Cache性能分析

5. 多级Cache的命中率

- 两级**Cache**的总未命中率(总失效率):

  总失效率 = $(\text{失效率})_{\text{第一级}} \times (\text{失效率})_{\text{第二级}} \ldots\ldots \times (\text{失效率})_{\text{第N级}}$

- **【例】10000次访存，第一级Cache失效400次，第二级Cache失效4次。**
  - $(\text{失效率})_{\text{第一级}} = 400/10000 = 4\%$
  - $(\text{失效率})_{\text{第二级}} = 4/400 = 1\%$
  - 利用两级**Cache**后总的失效率 = **0.04**%。

7.3.4 Cache性能分析

5. 多级Cache的平均访问时间

【例】访问内存需50ns, L1 1ns 10%失效率，L2 5ns 1%失效率，L3 10ns 0.2%失效率，求L1， L1+L2，L1+L2+L3构架下的平均访问时间。

L1 ： $T = 1\ ns + (0.1 \times 50\ ns) = 6\ ns$

L1+2 ： $T = 1\ ns + (0.1 \times [5\ ns + (0.01 \times 50\ ns)]) = 1.55\ ns$

L1+2+3 : $T = 1\ ns + (0.1 \times [5\ ns + (0.01 \times [10\ ns + (0.002 \times 50\ ns)])]) = 1.5001\ ns$

虚拟存储器

Virtual Memory, VM

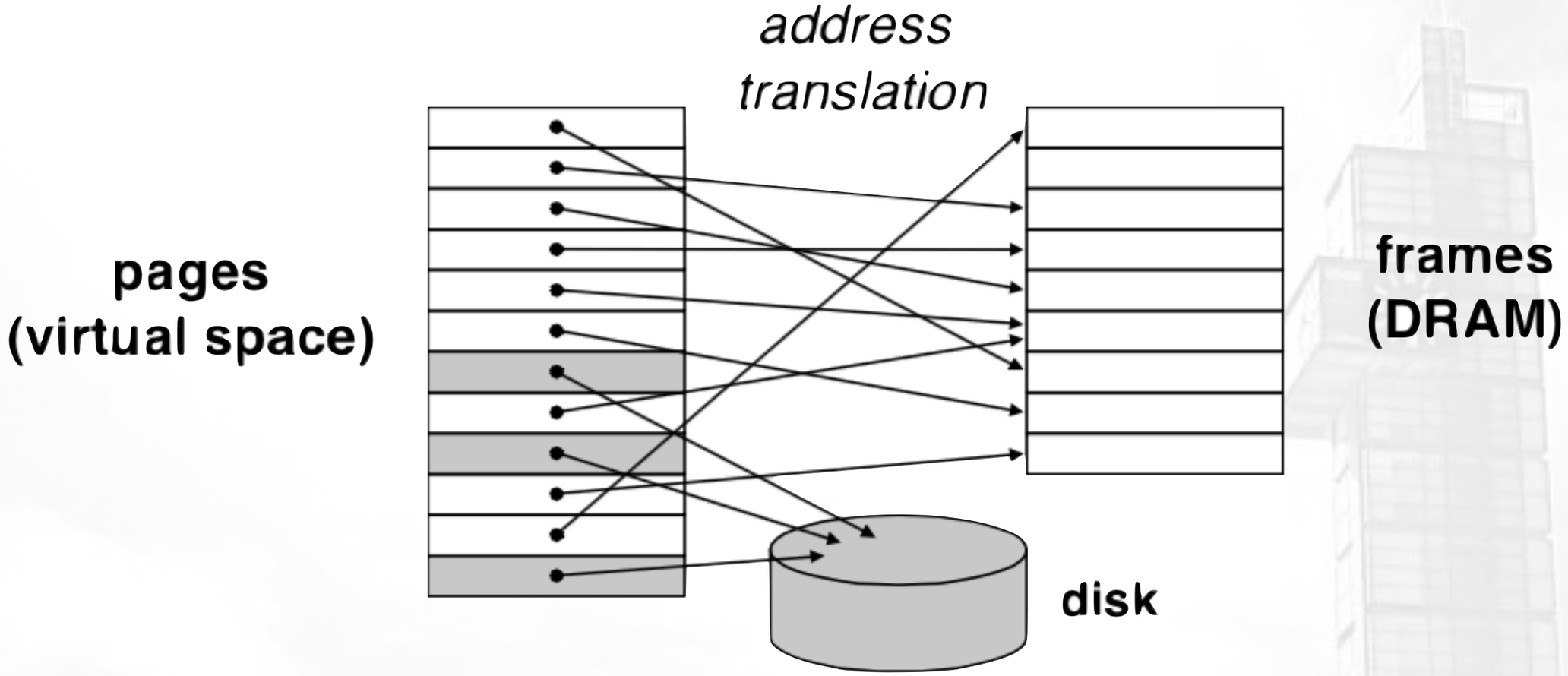
VM：两种地址空间

- 虚拟地址空间
  - 每个进程独占的“巨大 + 连续”的虚拟内存上的地址
- 物理地址空间
  - 实际的物理内存上的地址
  - 被操作系统内存+硬件完全屏蔽

VM的原理

- 1. 把内存（物理&虚拟）分为同样大小的块或段
- 2. 每个进程在各自的互相隔离的VM中占用一些内存块或段
- 3. 通过“地址变换表”Address Translation Table，将VM中的块或段映射到物理内存
- 虚拟地址<—>物理地址变换由硬件完成~~

VM的原理



- 某种角度来说，内存成了外存的Cache
  - 实现“进程连续并且更大”的内存
  - 程序在内存中的实际位置无意味
  - 进程隔离、进程共享、内存动态管理等优势

7.4 虚拟存储器

- 因地址映象和变换方法不同，

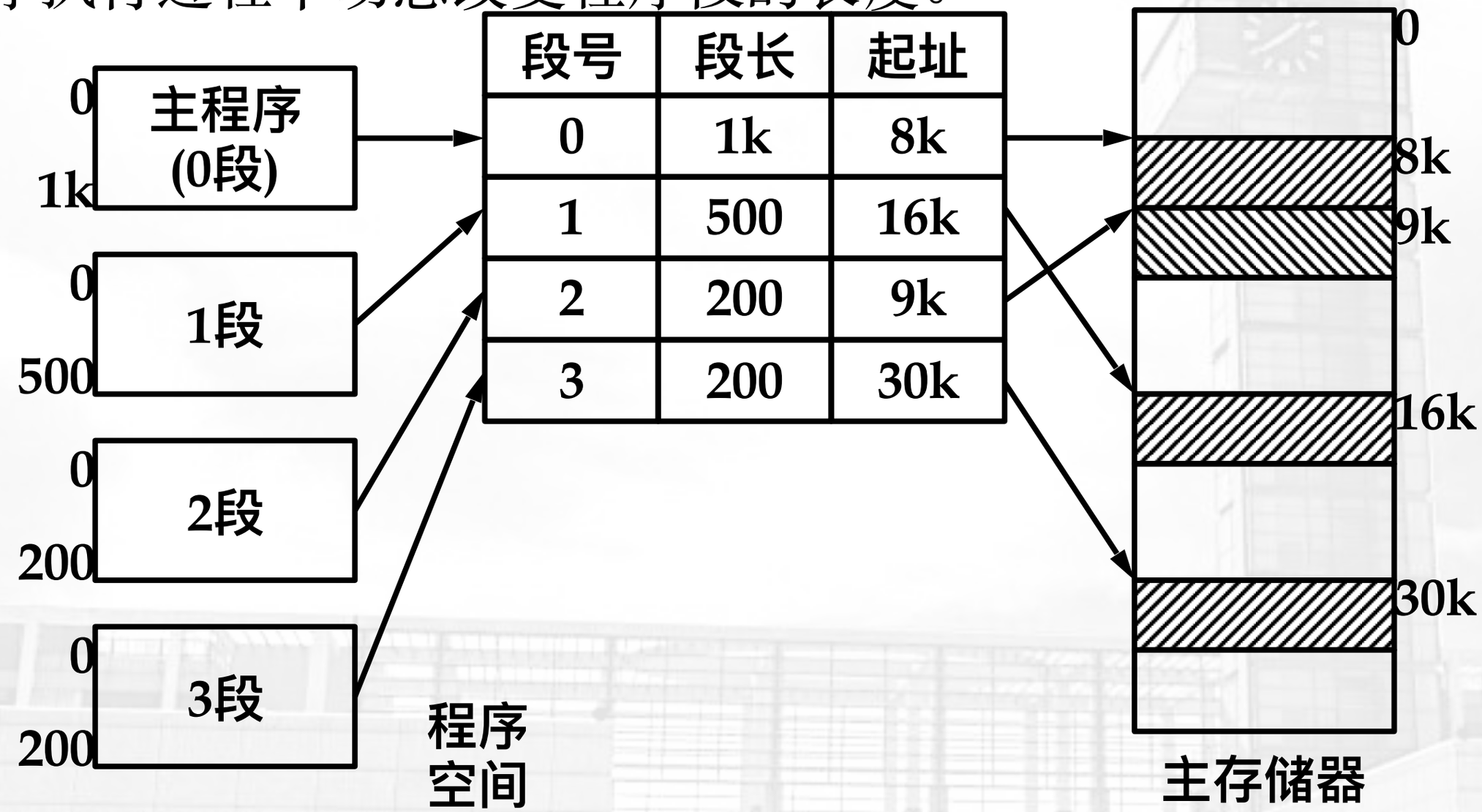
  有三种虚拟存储器：

  - **段式虚拟存储器**
  - **页式虚拟存储器**
  - **段页式虚拟存储器**

段式虚拟存储器

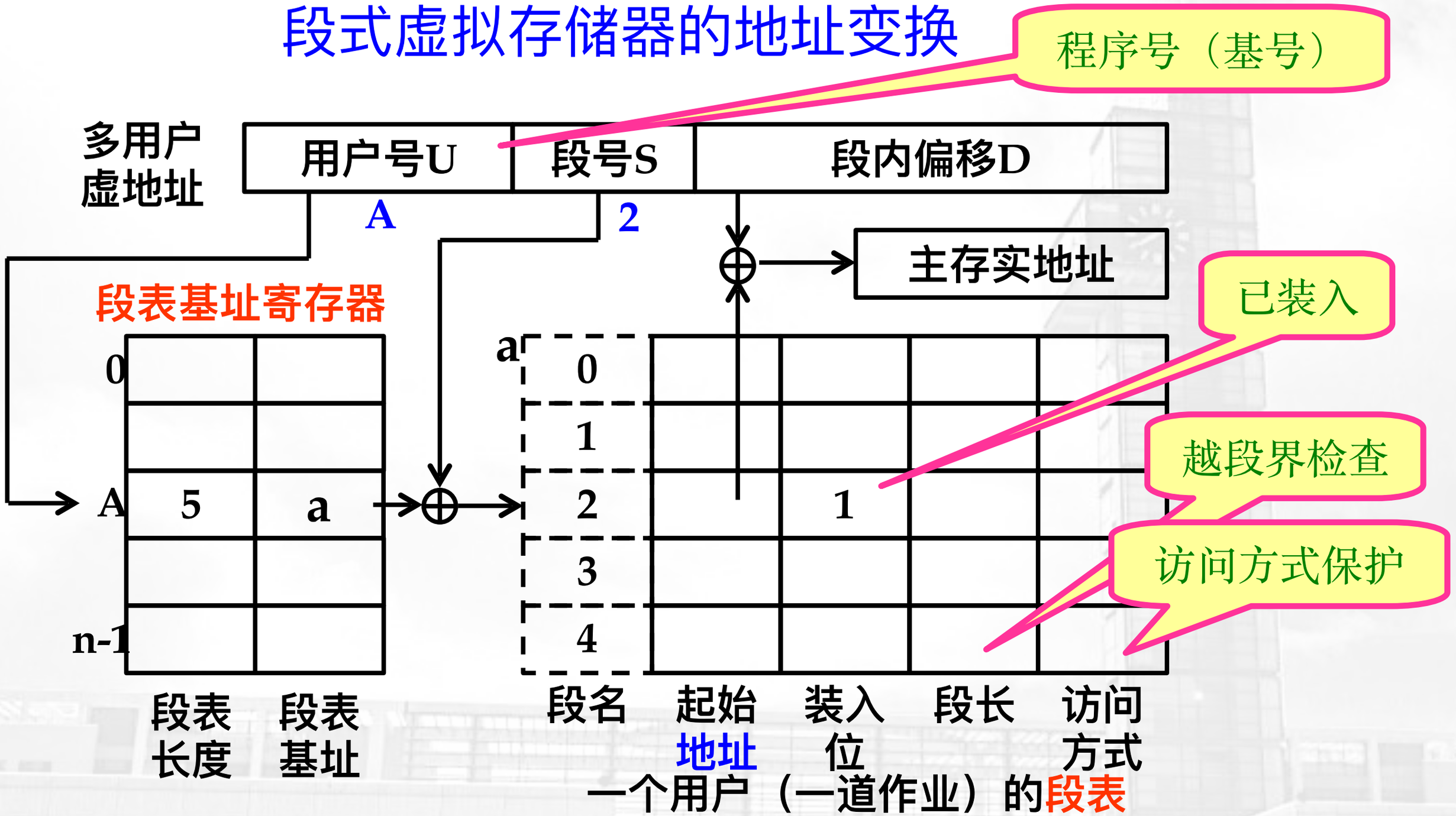
- 地址映象方法：

每个程序段都从0地址开始编址，长度可长可短，可以在程序执行过程中动态改变程序段的长度。



段式虚拟存储器

段式虚拟存储器的地址变换



段式虚拟存储器

- 主要优点：

① 程序的模块化性能好。

② 便于多道程序共享主存中的某些段。

③ 程序的动态链接和调度比较容易。

④ 便于按逻辑意义实现存储器的访问方式保护。

- 主要缺点：

① 地址变换所花费的时间长，两次加法。

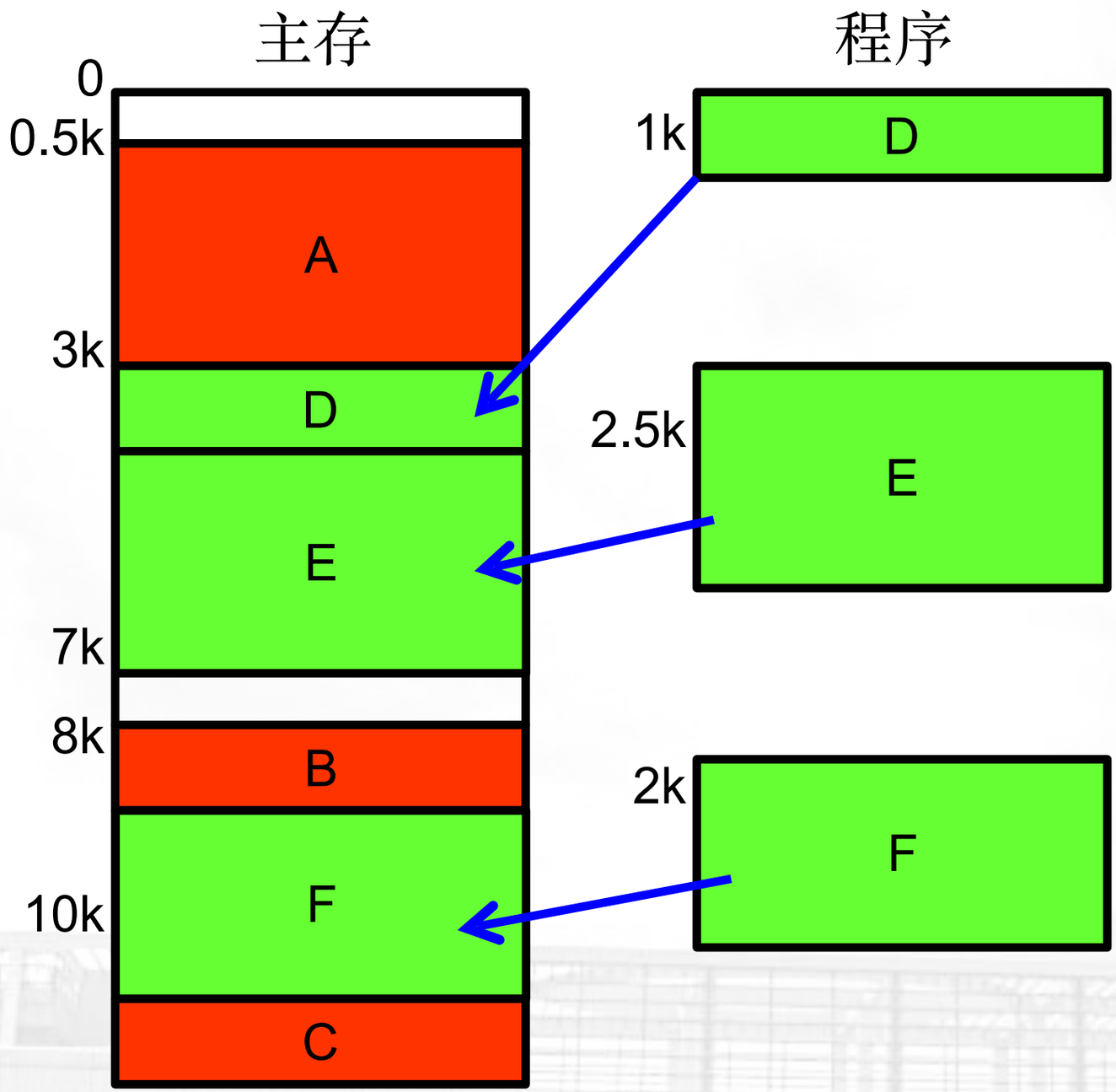
② 段映象表庞大，地址、段长字段太长。

③ 主存储器的利用率往往比较低——存储管理复杂；段间“零头”。

④ 对辅存（磁盘存储器）的管理比较困难。

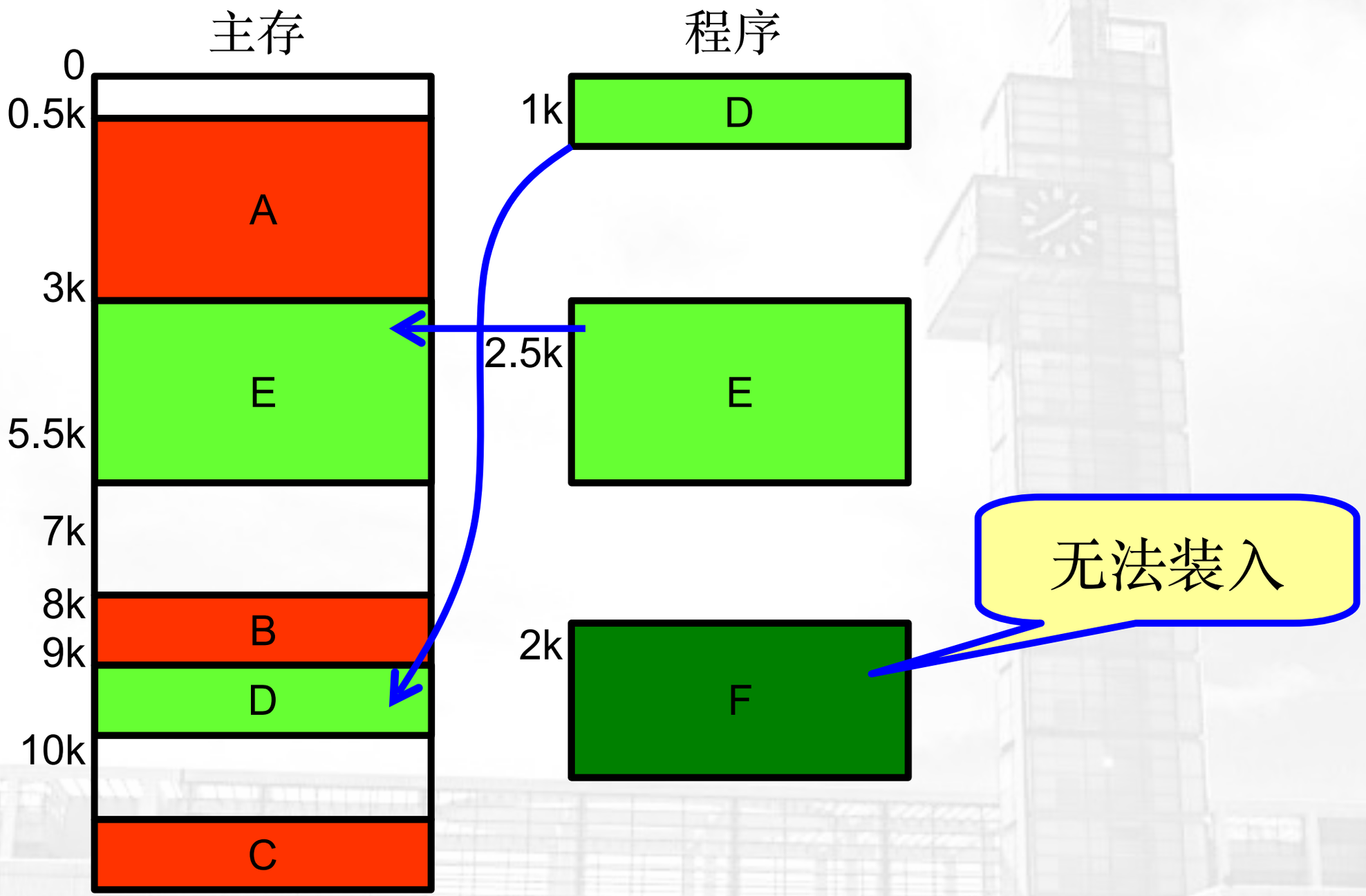
段式虚拟存储器

首先分配法



段式虚拟存储器

最佳分配法

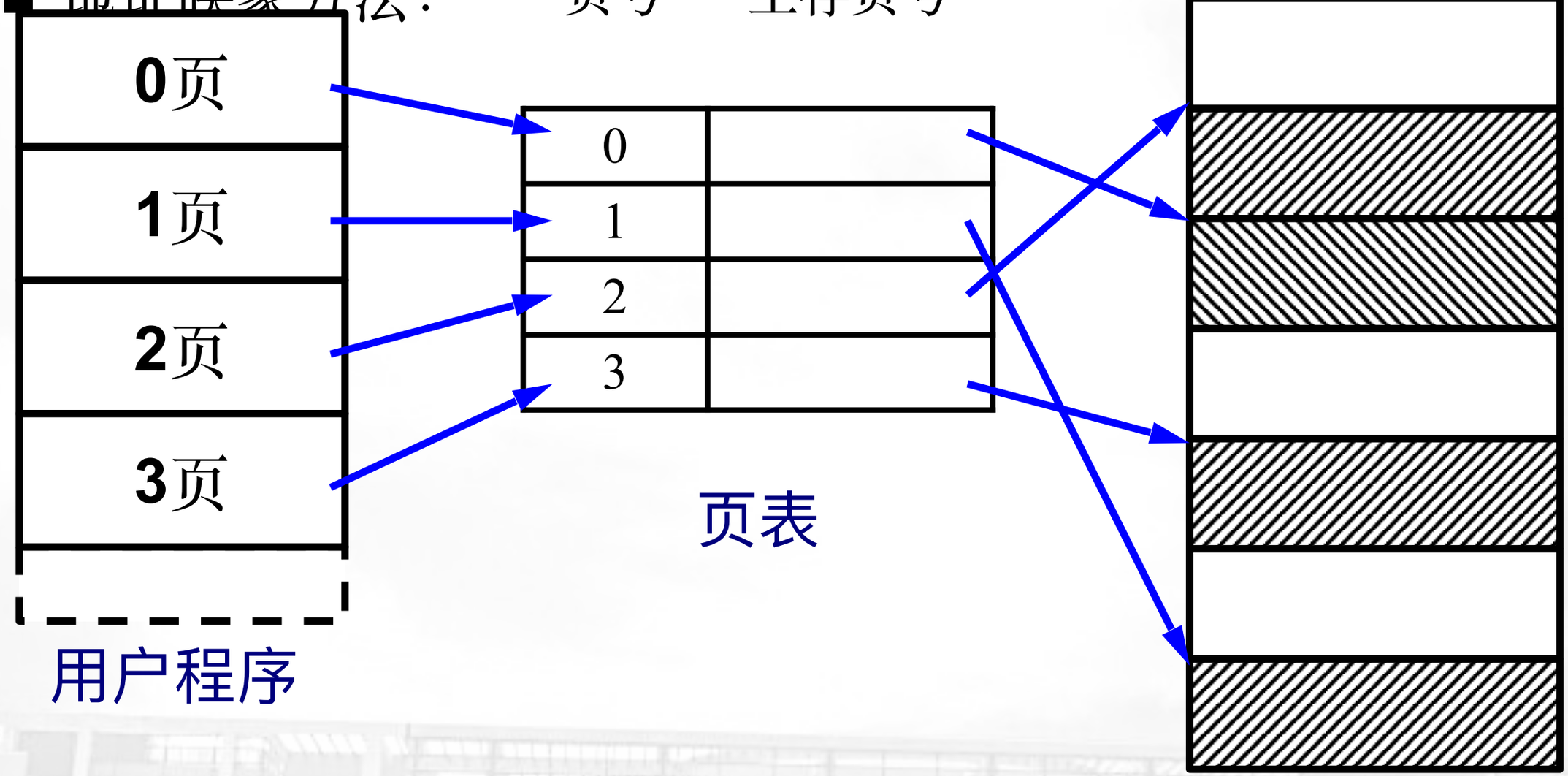


页式虚拟存储器

虚页/逻辑页

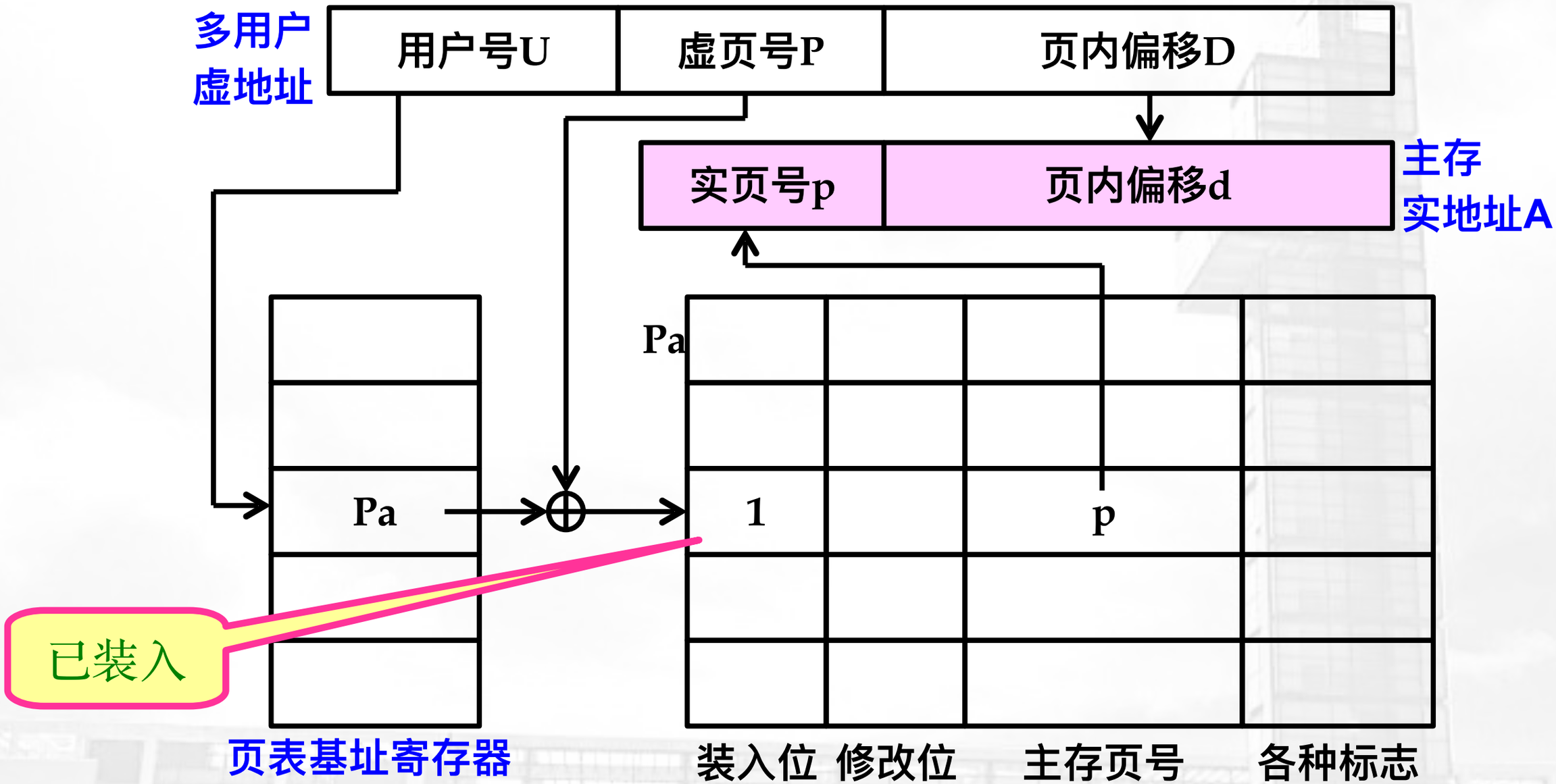
实页/物理页

地址映象方法：　　页号　　主存页号



页式虚拟存储器

- 地址变换方法：



7.4 虚拟存储器　　三、页式虚拟存储器

■主要优点：

① 主存储器的利用率比较高。

② 页表相对比较简单，使用硬件少。

③ 地址变换的速度比较快。

④ 对磁盘的管理比较容易。

■ 主要缺点：

① 程序的模块化性能不好

② 页表很长，需要占用很大的存储空间

例如：虚拟存储空间**4GB**，页大小**1KB**，则页表的容量为**4M**存储字。如果每个页表存储字占用**4**个字节，则页表的存储容量为**16MB**。

段页式虚拟存储器

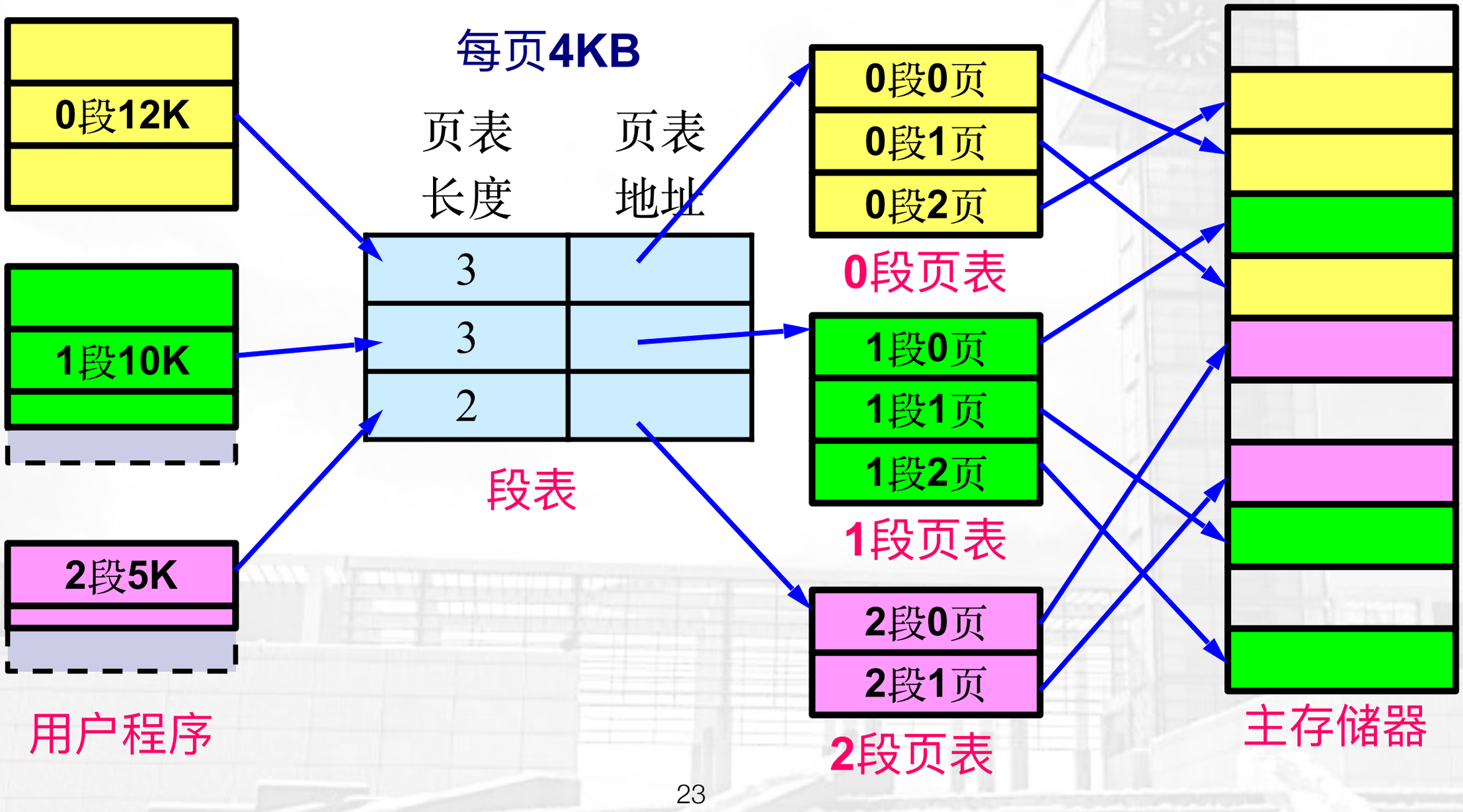
- 段页式存储管理方式：

  将程序按逻辑意义先分成段，再让各段和实主存都机械等分成相同大小的页面。每道程序通过一个段表和相应的一组页表来进行程序在主存空间中的定位。

段页式虚拟存储器

■ 地址映象方法：

每个程序段在段表中占一行， 在段表中给出页表长度和页表的起始地址，页表中给出每一页在主存储器中的实页号。



■ 地址变换方法：

① 先查段表，得到页表起始地址和页表长度；

② 再查页表找到要访问的主存实页号；

③ 把实页号**p**与页内偏移**d**拼接得到主存实地址。

段页式虚拟存储器

“C” “1” “2”

多用户虚地址A_V

用户号U	段号S	虚页号P	页内偏移D

实页号p	页内偏移d

主存实地址A

段表基址寄存器

	S_A
	S_B
	S_C

程序A段表 S_A

程序C段表 S_C

装入位	修改位	标志	页表长	页表地址
				a
1	0/1		3	b
				c

多用户段表

C_0段

a			

C_1段

b			
1	p	0/1	

C_2段

c			

装入位 实页号 修改位 标志

多用户页表

VM页替换

- 物理内存不足时，将暂时不用的物理页替换到外存里可以维持系统运行， 虽然慢
- 常规策略
  - LRU，最近最少使用
    - 需要为每一页记录时间戳
  - FIFO，先进先出

VM页替换

- 如果所需页面不在内存...

1. CPU发现缺页
2. CPU发出Page Fault中断
3. 操作系统对Page Fault中断的处理程序启动
4. PF中断处理程序从外存调数据进内存（文件系统等...)
5. 操作系统更新页表（有效位=1）
6. 操作系统让CPU继续之前工作

Still on Virtual Memory…

在VM架构下，CPU如何访问内存?

Memory Access in VM ARCH

虚存地址(虚存页号 + 页内偏移)

How ?

物理内存地址

访问物理内存地址

Memory Access in VM ARCH

虚存地址(虚存页号 + 页内偏移)

How ?

1. 找到当前进程的页表/段页
2. 查表 + 计算

物理内存地址

访问物理内存地址

So How to accelerate ?

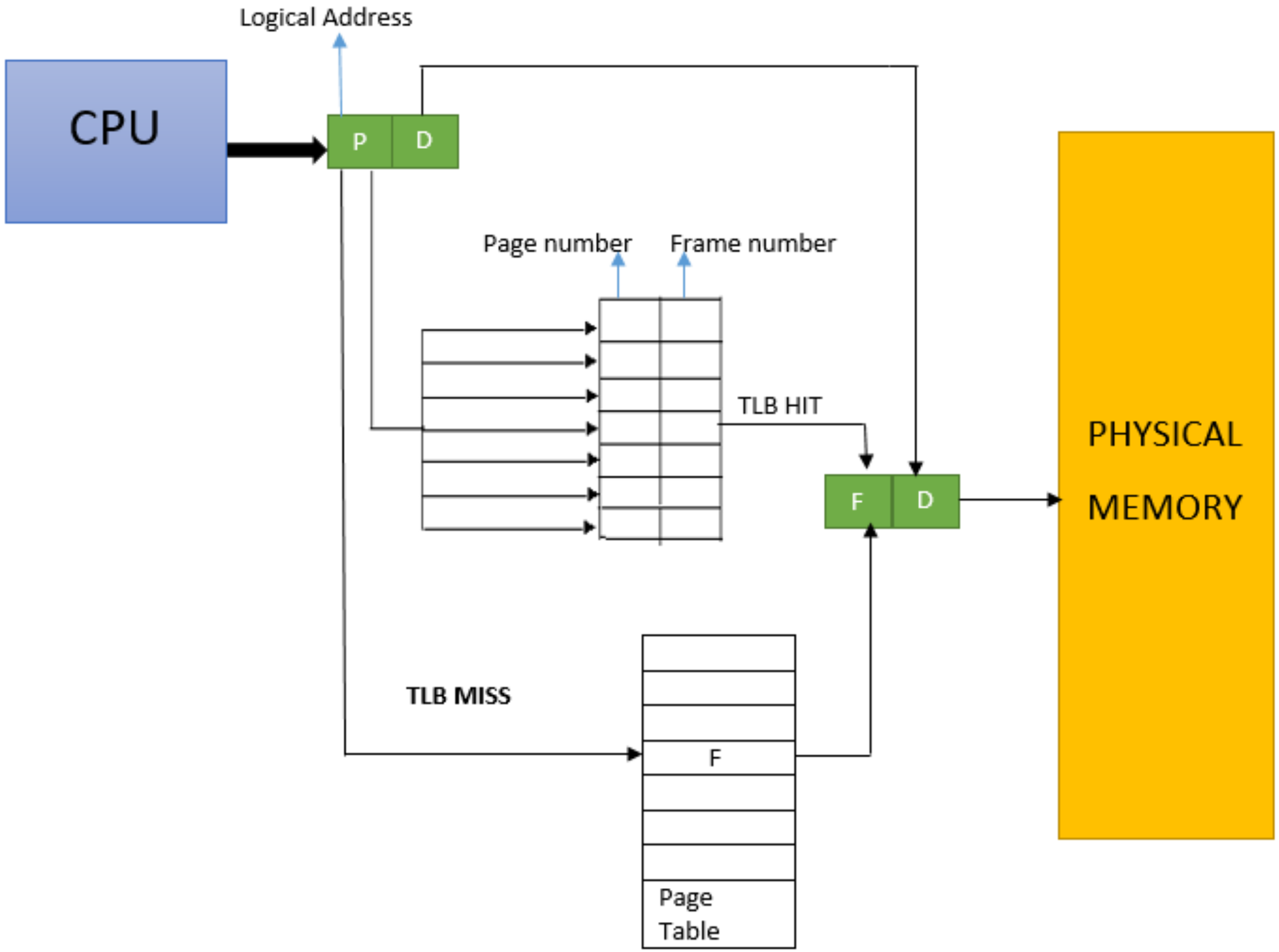
不难推测几点：

1. 地址转换(or 计算)成为CPU极频繁的工作；
2. 页表被极频繁的访问;
3. so, 大量页表内容进驻cache;
4. cache容量有限，进程指令/数据的cache裕度被挤占；
5. 更可能的情况是，页表体积常远大于cache，使页表本身都无法快速访问；
6. 系统性能严重下降

Translation Lookaside Buffer (TLB)

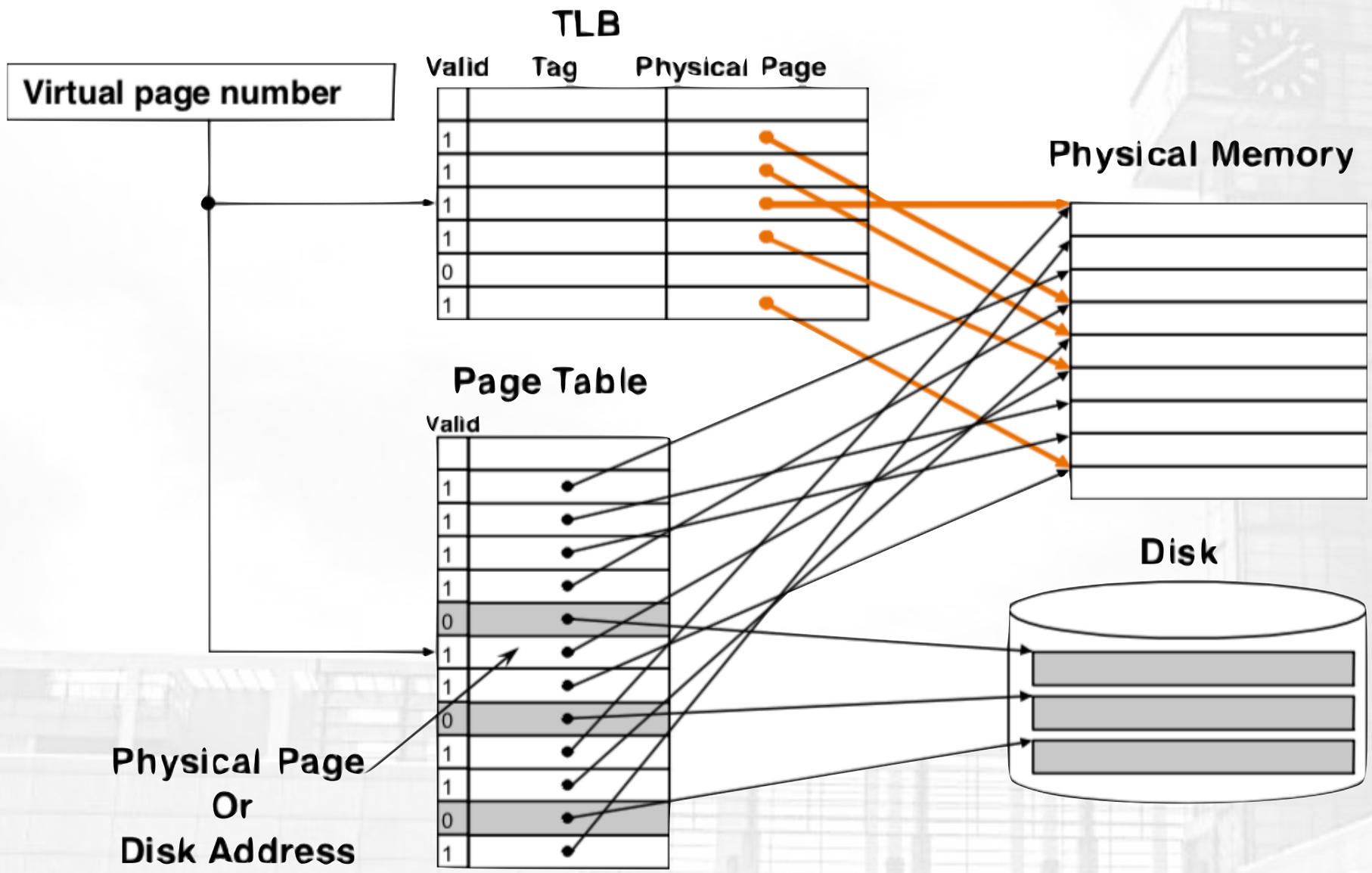
- 专用于页表(段页表)的cache !!!
- 比L1 Cache的级别更高!
  - 在CPU和cache之间
- 同样有hit rate/ miss rate的指标

Translation Lookaside Buffer (TLB)



TLB (Translation Look-Aside Buffer)

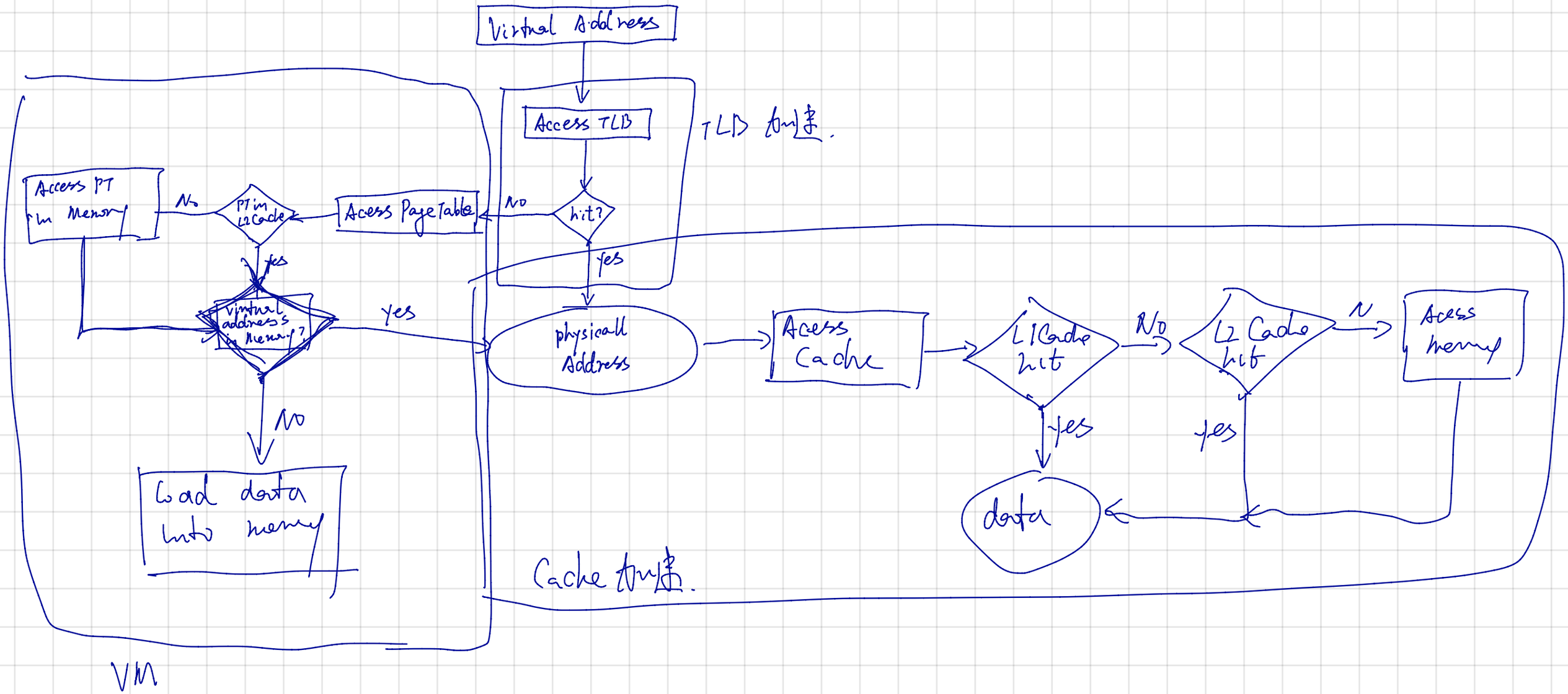
- TLB，是**CPU内部对页表的专用缓冲区**



在TLB加持下，VM架构下内存访问?

Paged Memory Access, Empowered by TLB

- TLB, Cache, VM的访问路径



Paged Memory Access, Empowered by TLB

- 同样有hit rate/ miss rate的指标（Cache的性能指标全部适用)
  - miss rate — 0.01~0.1%
  - hit time — 1 clock cycle
  - miss time — 10~100 cycles
- 也可以有多级TLBs
  - Intel Core系列，两级TLB
- 也可以有分类
  - Data TLB, Instruction TLB

作业

6；　7；　16；　22；　23；　27

第六章 习题讲解