



清华大学  
Tsinghua University

# 第三单元 层次存储系统

## 第四讲 虚拟存储器

刘卫东

计算机科学与技术系

# 内容提要



- ✚ 设置虚拟存储器的目的
- ✚ 虚拟存储器的管理
- ✚ 段页式管理的地址映射
- ✚ Pentium的虚拟存储管理

# 存储管理



## ❖ 单道程序

- ❖ 单一

- ❖ 操作系统

1、程序员编程时需要知道存储空间的使用情况;

2、无法实现存储空间的复用

## ❖ 多道程序

- ❖ 操作系统及若干用户程序同时运行

- ❖ 程序员编程使用的空间与程序运行空间相互独立

- ❖ 操作系统对存储进行管理

  - ◆ 地址空间转换和映射

  - ◆ 内存共享

  - ◆ 内存保护

# 存储容量需求



## 应用需求

### 海量数据处理

天气预报、地震预测、石油勘探、搜索...

### 多媒体信息处理

语音、图形、图象...

## 软件需求

Nathan软件第一定律：软件是一种可以膨胀到充满整个容器的气体。

## 技术需求

### 多进程、多道程序

给程序员一个大容量、可管理的编址空间

# 提高存储器容量



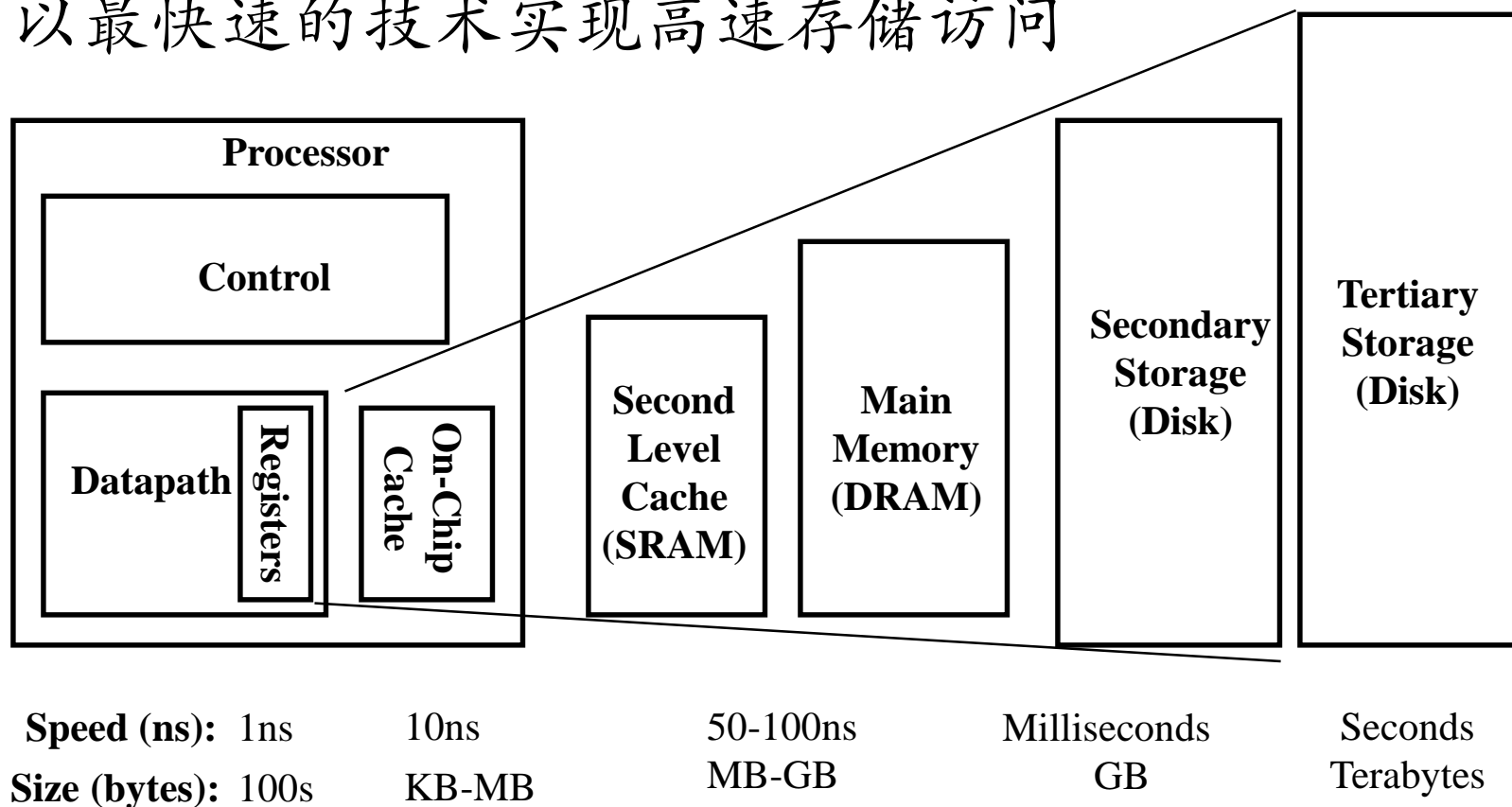
- ❖ 降低主存储器成本，在同样成本下，可以获得更大的主存容量
  - ❖ 主存的价格到今天也依然比较昂贵
  - ❖ 程序对主存的“胃口”的增加和主存价格的降低速度几乎同样的快
- ❖ 采用虚拟存储器
  - ❖ 只在确实需要的时候才把程序和数据装入到主存中
    - ◆ 交换的粒度
    - ◆ 地址转换
    - ◆ 共享及保护

# 现代计算机的层次存储器系统



利用程序的局部性原理:

- 以最低廉的价格提供尽可能大的存储空间
- 以最快速的技术实现高速存储访问

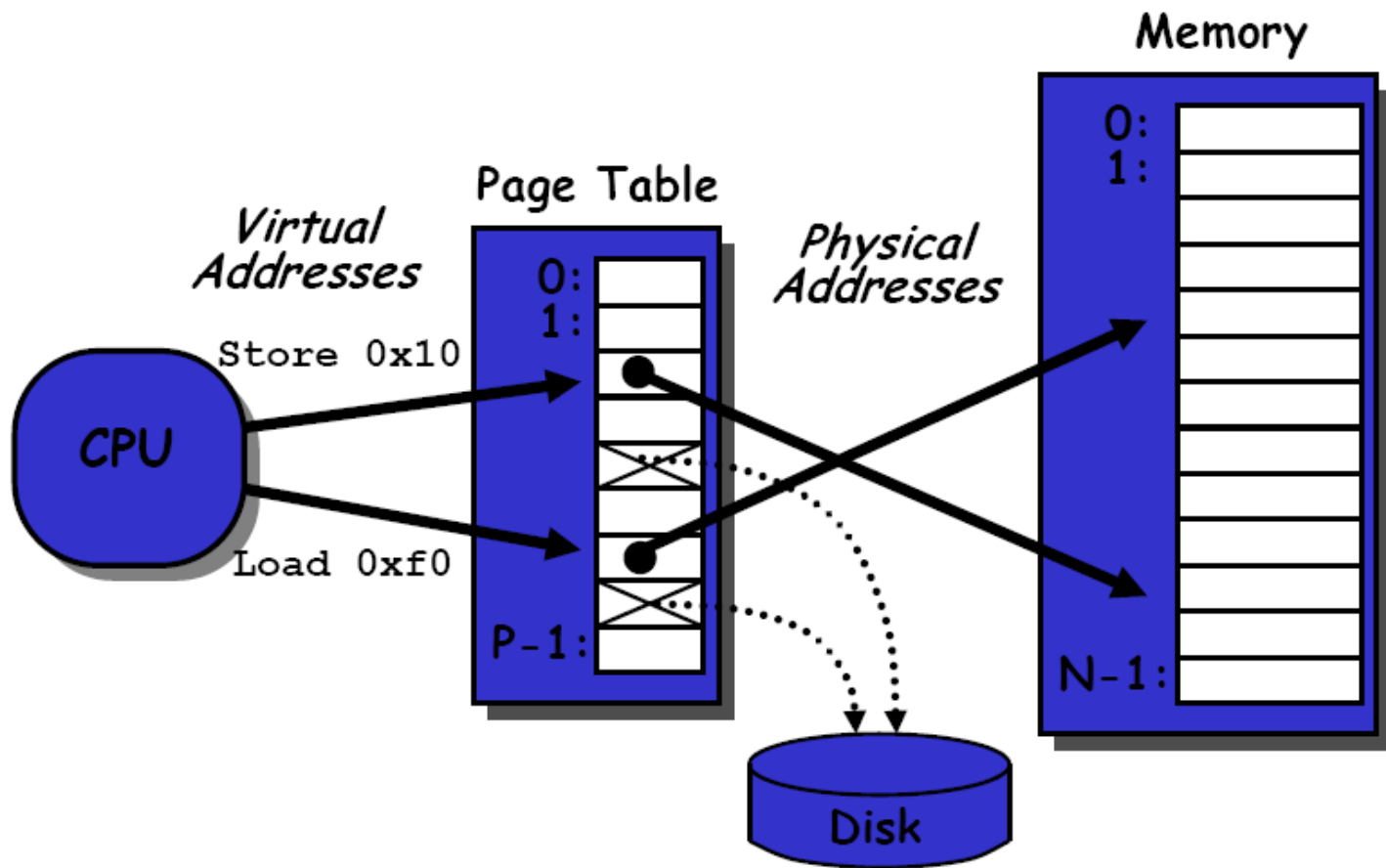


# 虚拟存储器



- ✚ 提供一种容量非常大的存储器
  - ❏ 多个任务所需存储器的总和大于实际存储器空间
  - ❏ 单个程序的地址空间超过了实际存储器
- ✚ 使得珍贵的物理存储器得到更好地利用
- ✚ 简化对存储的管理
- ✚ 地址空间：
  - ❏ 虚拟地址：程序员编程使用的地址
  - ❏ 虚拟地址空间：虚拟地址的集合
  - ❏ 逻辑地址：程序员使用的地址
  - ❏ 存储器地址：物理存储器的地址，也称物理地址（Physical Address）或实地址（Real Address）

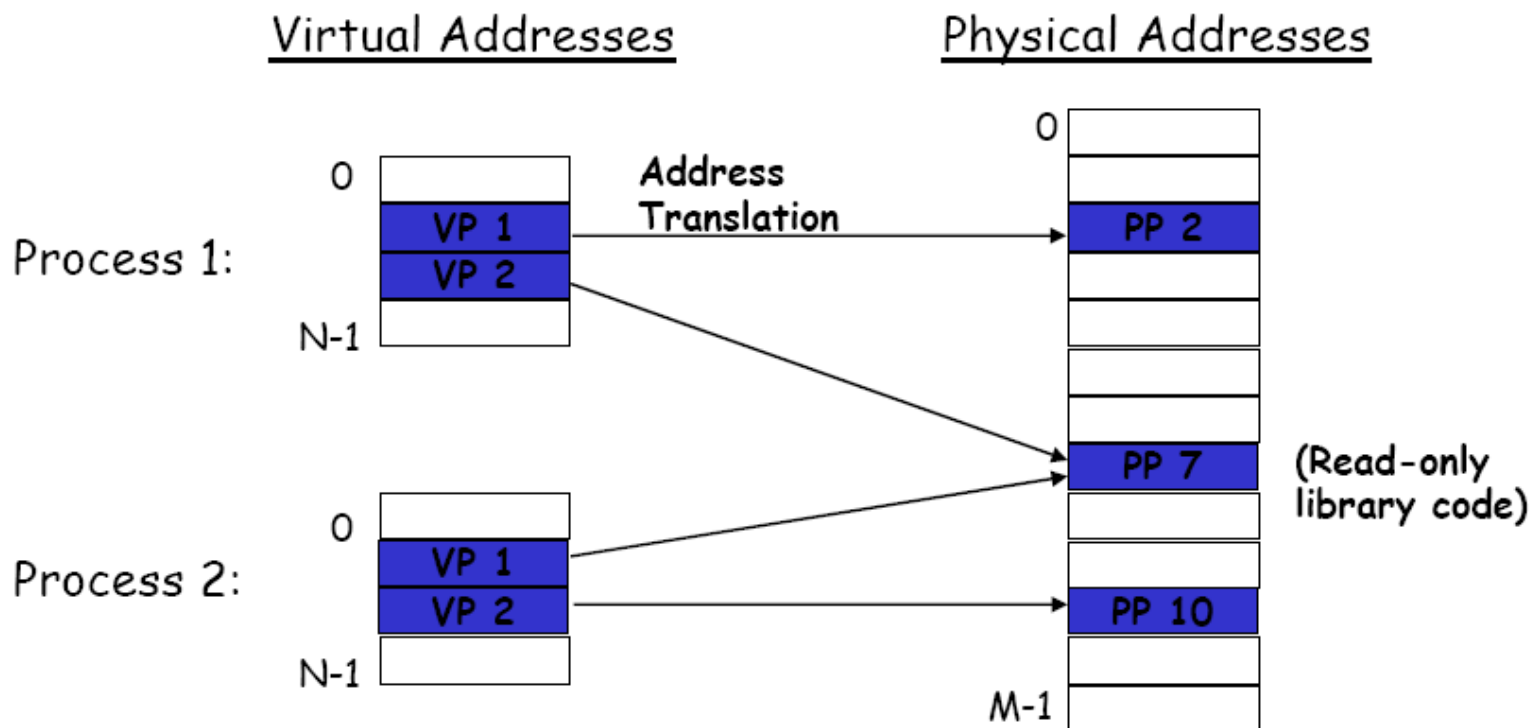
# 更大的逻辑地址空间



通过页表将虚地址转换为实地址

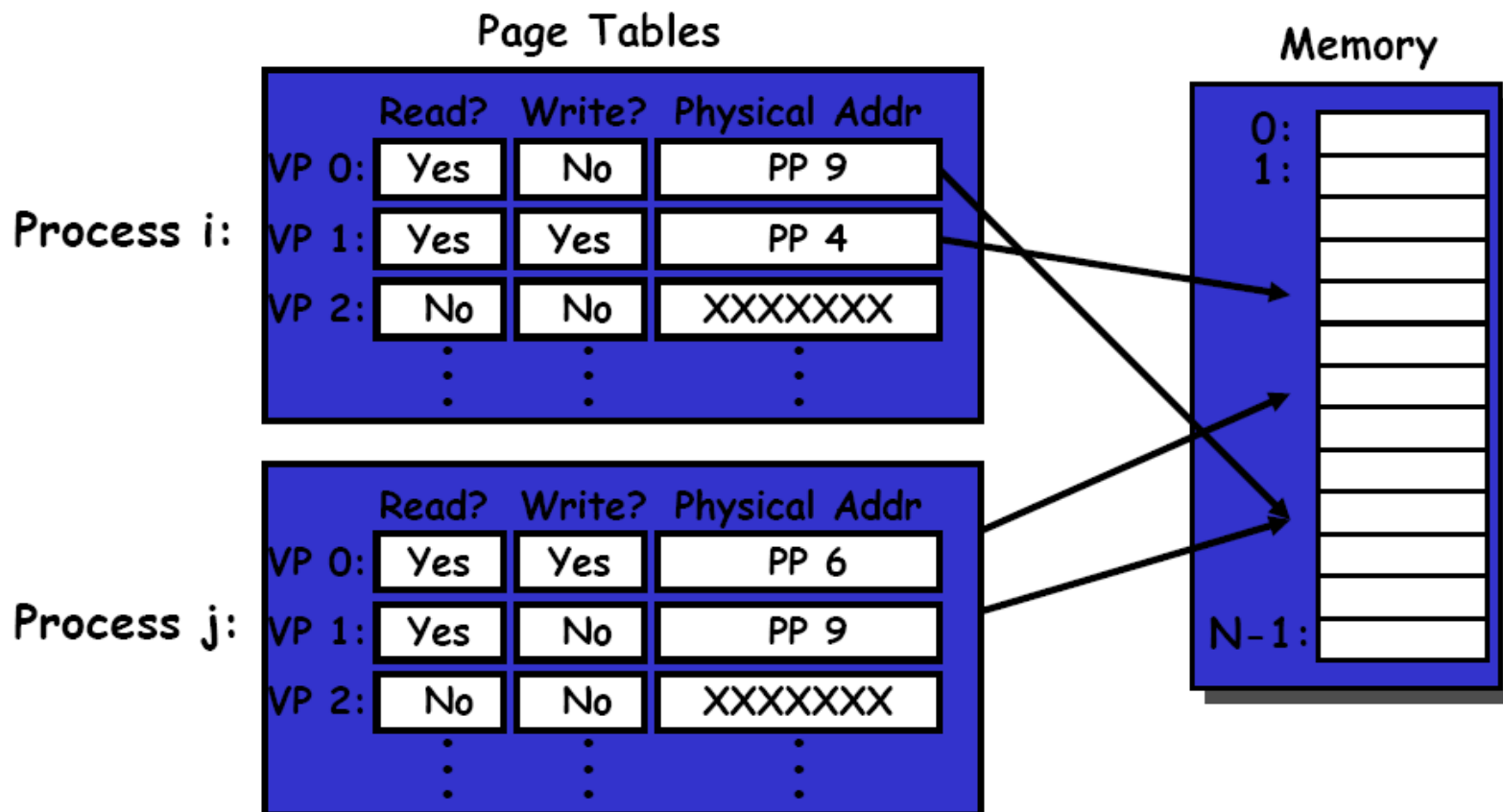


# 实现内存共享



- ✚ 每个进程有独立的逻辑地址空间
- ✚ 建立逻辑地址和物理地址的转换机制

# 实现内存保护



🔗 页表中存放有访问权限

🔗 通过硬件来保证权限（操作系统的“陷阱”操作）

# 虚拟存储器目的



## 容量

- 获得运行比物理存储器更大空间程序的能力

## 存储管理

- 内存的分配以及虚实地址转换

## 保护

- 操作系统可以对虚拟存储空间进行特定的保护...

## 灵活

- 程序的某部分可以装入主存的任意位置

## 提高存储效率

- 只在主存储器中保留最重要的部分

## 提高并行度

- 在进行段页替换的同时可以执行其它进程

## 可扩展

- 为对象提供了扩展空间的能力.

# 虚存与Cache的比较



## 虚存

- ❏ “主存——辅存层次”，主要目的是实现存储管理，并帮助解决存储容量的问题。
- ❏ 单位时间内数据交换次数较少，但每次交换的数据量大，达几十至几千字节。

## Cache

- ❏ Cache主要目的是解决存储速度问题，使存储器的访问速度不太影响CPU的运行速度。
- ❏ 单位时间内数据交换的次数较多，每次交换的数据量较小，只有几个到几十个字节。

# 虚存与Cache的不同



## ❖ 虚拟存储器

- ❖ 克服存储容量的不足
- ❖ 获得对主存储器管理的便利
- ❖ 由操作系统管理

## ❖ 高速缓冲存储器

- ❖ 解决主存储器与CPU性能的差距
- ❖ 获得最小粒度的访问
- ❖ 由硬件实现

# 虚拟存储器管理



## ❖ 逻辑地址与物理地址的对应关系

- ❖ 逻辑地址与物理地址如何转换

## ❖ 主存和辅助存储器数据交换

- ❖ 进行数据管理和调度

- ❖ 提高命中率

- ❖ 保护和共享

## ❖ 段式存储管理

## ❖ 页式存储管理

# 段式存储管理



## ✚ 段 (Segment)

- ✚ 程序模块化设计的结果
- ✚ 过程、函数、数组、...
- ✚ 逻辑上相对独立
- ✚ 共享和保护的最小单元

## ✚ 段式存储管理

- ✚ 以段作为存储管理的对象
- ✚ 容易实现共享、保护
- ✚ 容易产生碎片

# 段式存储管理的实现

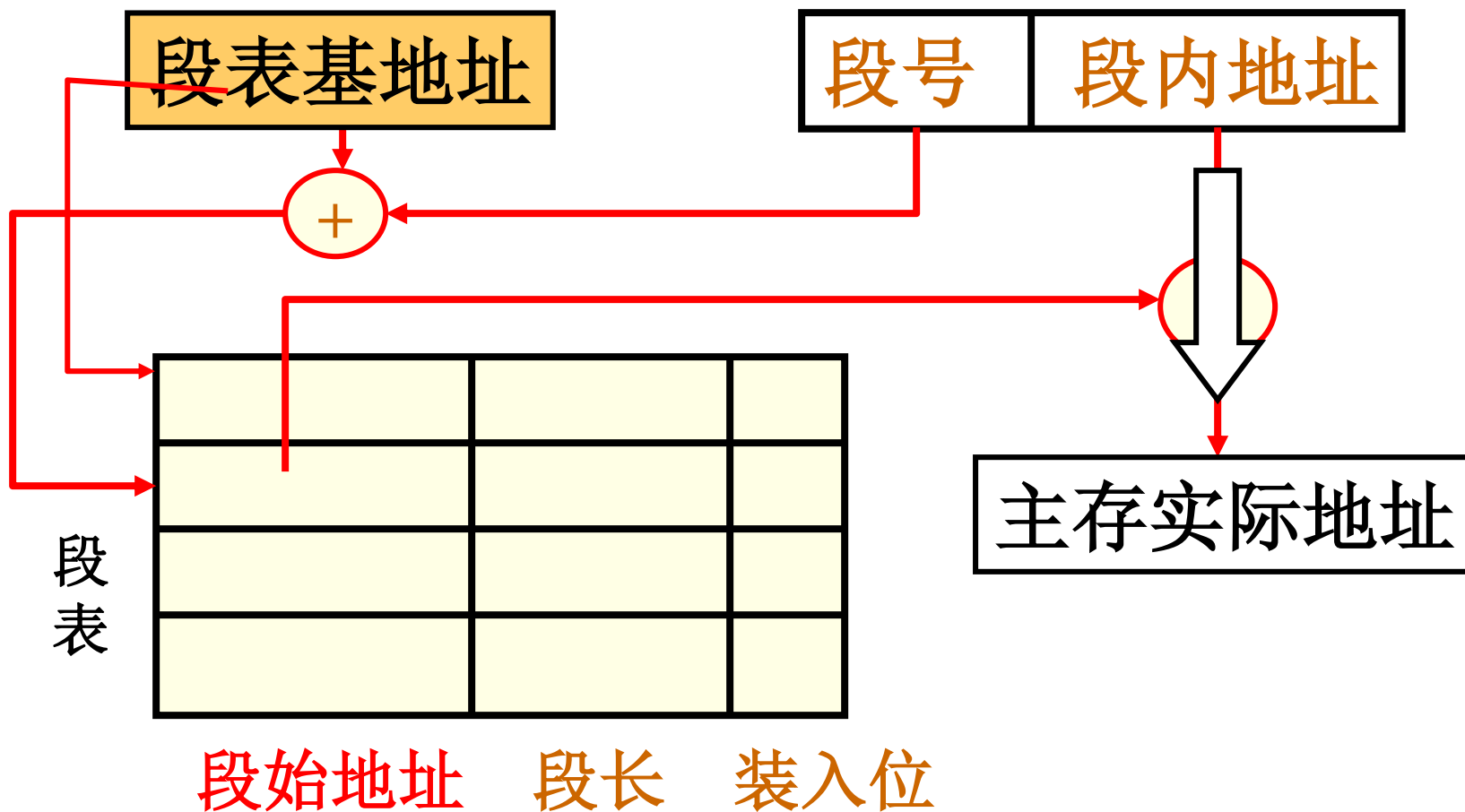


## ✚ 设置段表进行管理

- ▣ 段表基地址
- ▣ 段起始地址
- ▣ 段长
- ▣ 装入位
- ▣ 保护、共享等标志



# 段式管理地址转换



# 段式存储管理



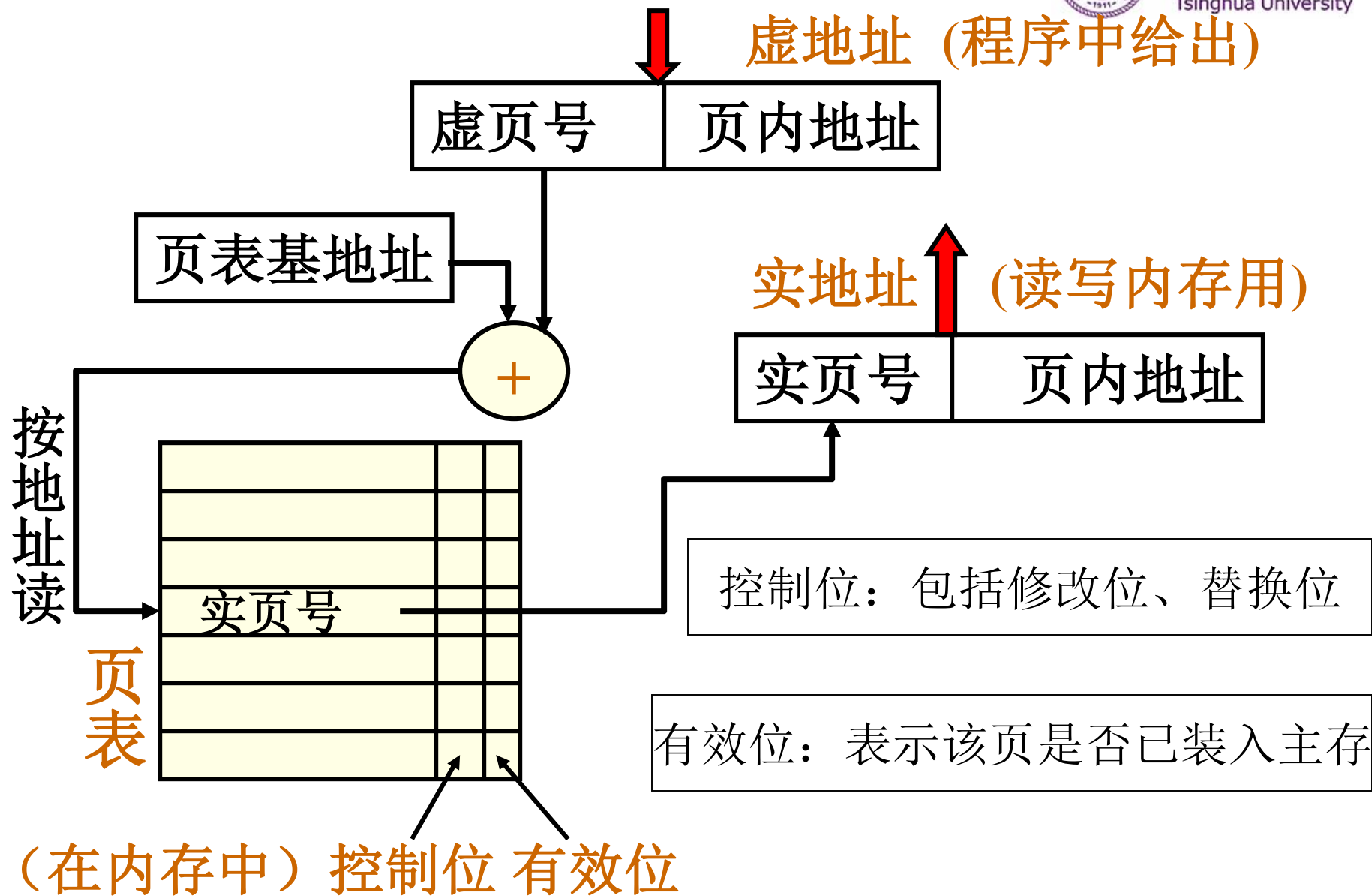
- ✚ 段的分界与程序和数据的自然分界相对应
- ✚ 易于编译、管理、修改和保护，便于多道程序共享
- ✚ 段长动态可变
- ✚ 段起点、终点不定
- ✚ 空间分配困难，容易产生碎片

# 页式存储管理



- ❖ 将主存和虚存划分为固定大小的页
- ❖ 以页为单位进行管理和数据交换
- ❖ 虚地址=虚页号+页内地址
- ❖ 实地址=实页号+页内地址
- ❖ 通过页表进行管理
  - ❖ 页表基地址寄存器
  - ❖ 实页号
  - ❖ 控制位

# 页表内容和页式管理



# 页表大小



✚ 与虚页数直接相关，但是

- ✚ 虽然理论上每个进程的逻辑空间很大，但其实大部分应该是不活跃的
- ✚ 实际调入到内存的内容不可能超过物理存储空间

✚ 如何减少页表本身所占的空间？

✚ 而且还要实现简单

◆ 页表访问频繁

✚ 两种途径

- ◆ 层次页表 (hierarchical page table)
- ◆ 反转页表 (inverted page table)

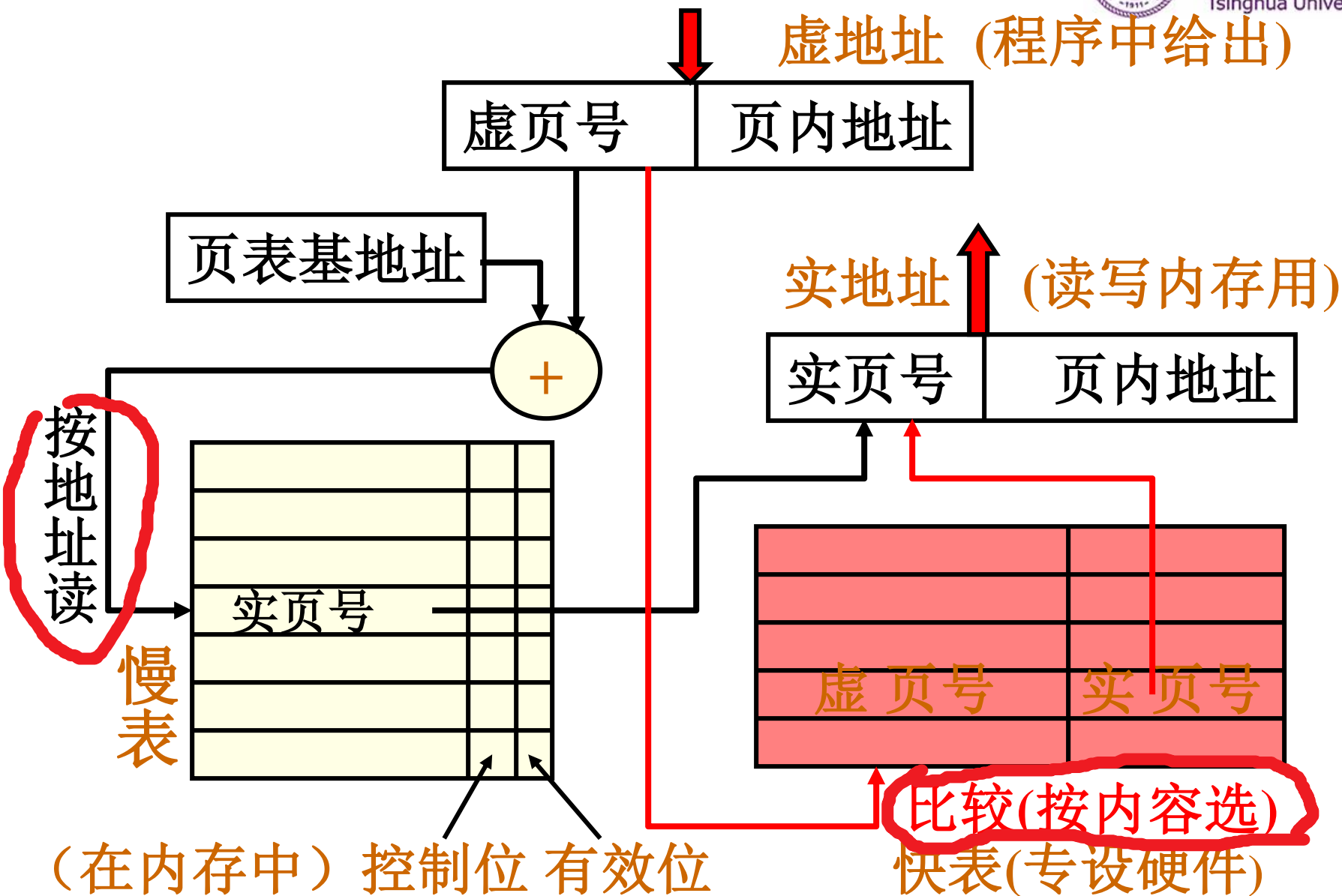
# 页式虚拟存储器的访问过程



1. 得到程序给出的虚地址；
2. 由虚地址得到虚页号；
3. 访问页表，得到对应的实页号；
4. 若该页已在内存中，则根据实页号得到实地址，访问内存；
5. 否则，启动输入输出系统，读出对应页装入主存，再进行访问。

增加由硬件实现的快表，提高访问速度

# 页表内容和页式管理



# 转换旁路缓冲 (TLB)



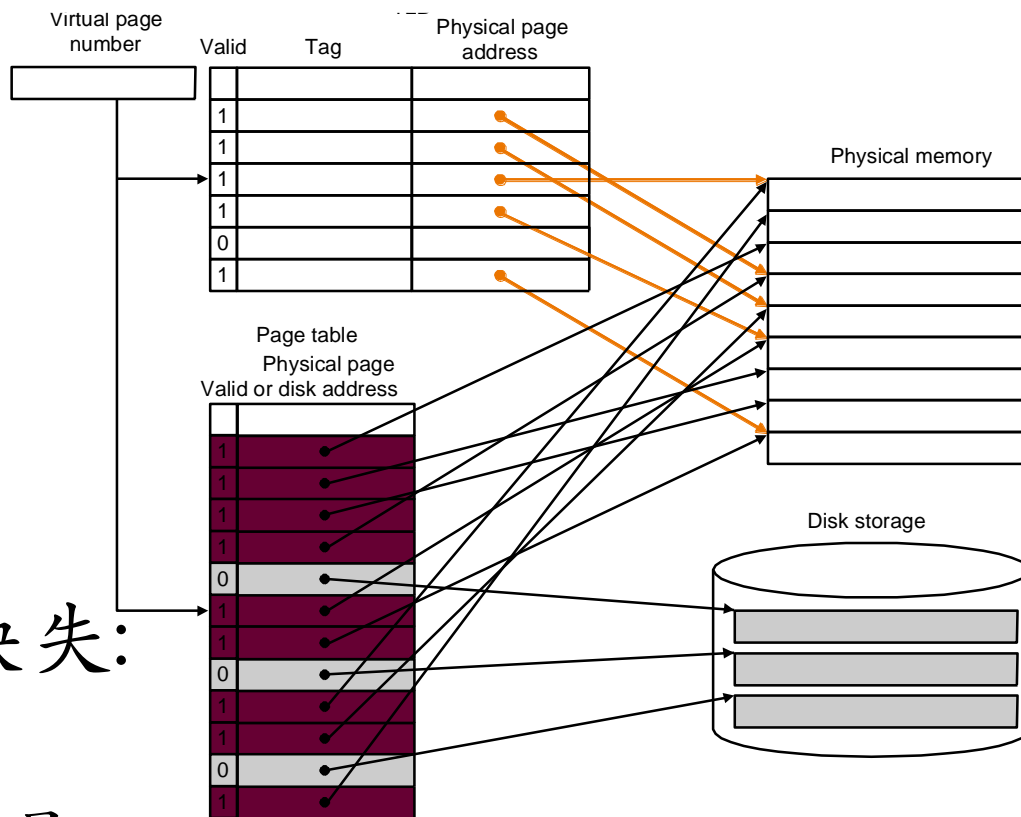
访问频繁:速度第一

TLB 缺失将产生异常:

- ❑ 流水线停止
- ❑ 通知操作系统
- ❑ 读页表
- ❑ 将表项写入 TLB
- ❑ 返回到用户程序
- ❑ 重新访问

因此, 应尽量减少缺失:

- ❑ 多路组相联
- ❑ 再尽量提高TLB的容量

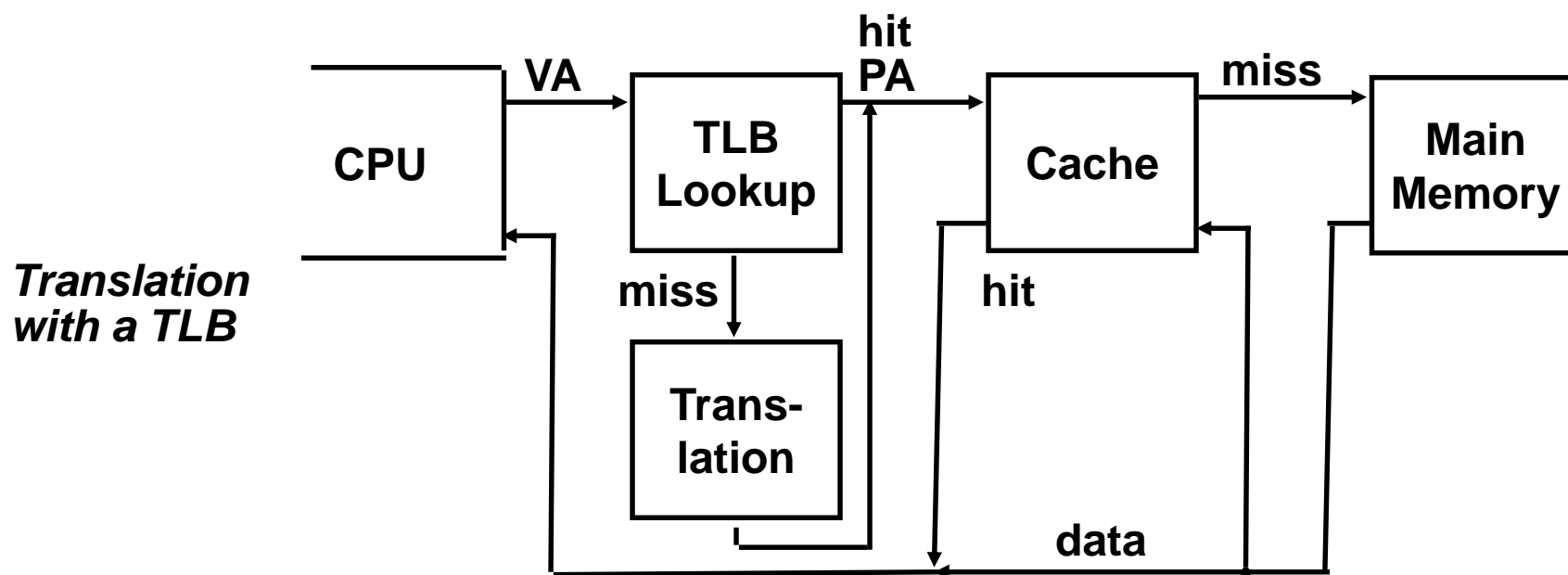




# TLB



- 为页表设置的专用Cache，高效实现虚页号到实页号的转换
- 多路组相联、全相联
- 容量较小，128~256个表项



# 页面大小的选择



- ❖ 减少内部碎片
- ❖ 缩小页面大小可以减少内部碎片
- ❖ 但是：需要更大的页表
- ❖ 趋势：增大页面大小
  - ❖ RAM价格下降，内存存储器容量增大
  - ❖ 内存和外存性能差距增大
  - ❖ 程序员需要更大的地址空间
- ❖ 目前：页面大小为4K左右

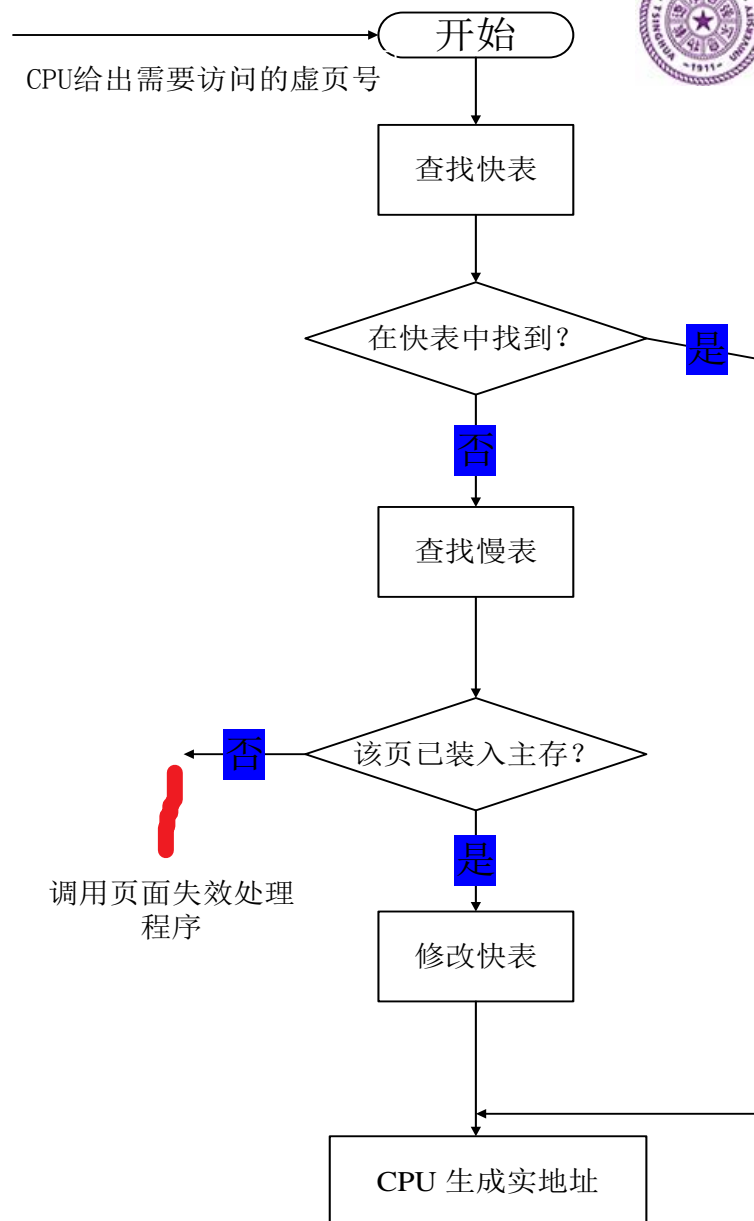
# 页面替换算法



## 最近最少使用 (LRU)

- 将页帧按照最近最多使用到最近最少使用进行排序，再次访问一个页帧时，将该页帧移到表头，替换时将表尾的页帧换出。
- 一点改进：替换出其中一个“干净”的页帧。

# 地址转换

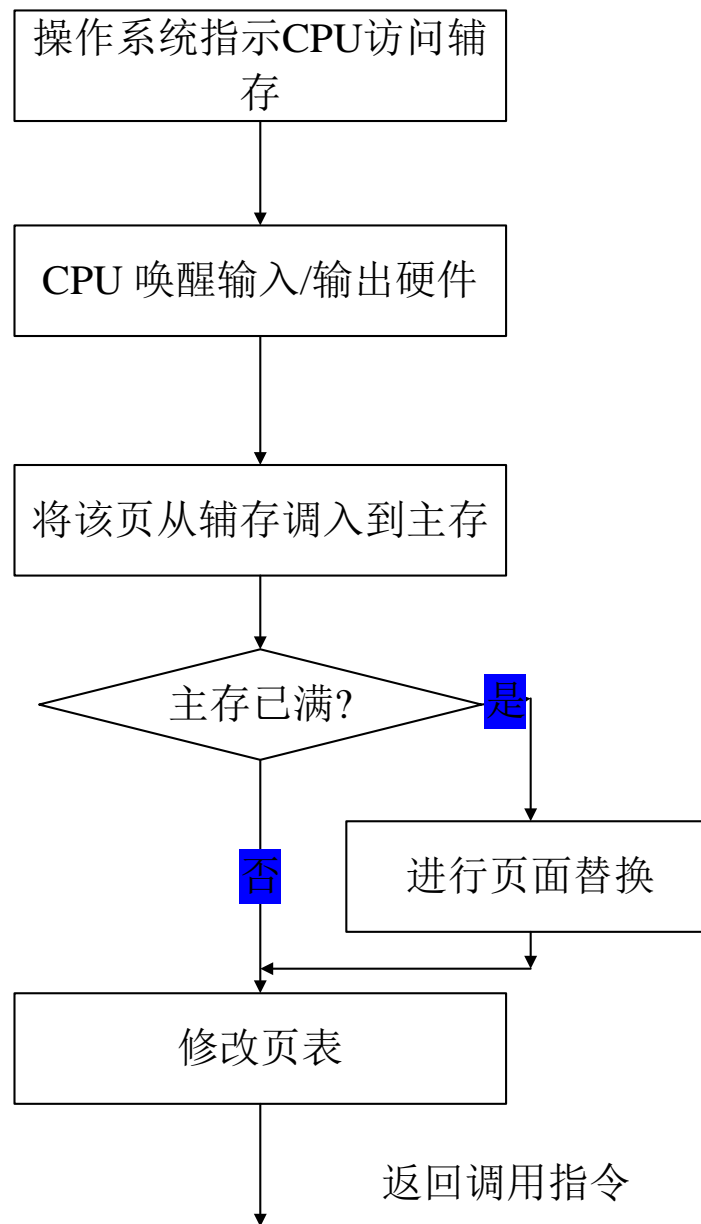


# 页失效处理

页面失效处理程序



清华大学  
Tsinghua University



# 段页式虚拟存储管理



是段式虚拟存储器和页式虚拟存储器的综合。它先把程序按逻辑单位分为段，再把每段分成固定大小的页。操作系统对主存的调入调出是按页面进行的，但它又可以按段实现共享和保护，可以兼取页式和段式系统的优点。其缺点是需要在地址映射过程中多次查表。其地址映射通过一个段表和一组页表来进行。

# Pentium的虚存管理



- 不分段也不分页模式。在这种模式下，虚拟存储的地址空间和物理存储空间大小相同，可以用在复杂度较低但对性能有较高要求的场合。
- 页式管理模式。这种模式将主存分成固定长度的页，通过页进行存储保护和管理。
- 段式管理模式。段式管理模式按程序本身的逻辑段来划分主存空间，与页式管理相比，段的长度可变，而且，由于段表存放在Pentium的内部寄存器中，段式管理的存储访问时间是可以预知的。
- 段页式管理模式。段页式管理模式首先按程序的逻辑结构将主存分段，对段进行访问控制，然后，在段内分页，按页进行数据交换。

# Pentium虚实地址的转换



## ❖ 虚地址（逻辑地址）：

- ❑ 程序员给出的虚拟地址，格式为段号+段内偏移（16位+32位），每段大小不超过4GB，一共不超过 $2^{14}$ 段。（段号中有两位用来表示段优先级）

## ❖ 实地址：

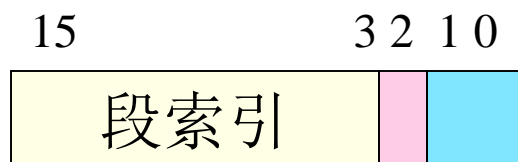
- ❑ 32位的实际内存地址。



# 段号和段表的格式



段号:



0: GDT 1: LDT

段表基地址

段类别

段表:

BASE 31..24	G	D / B		Segment limit 19..16		BASE 23..16
BASE 15..0				Segment Limit 15..0		

线性地址=BASE + OFFSET

# 两级页表



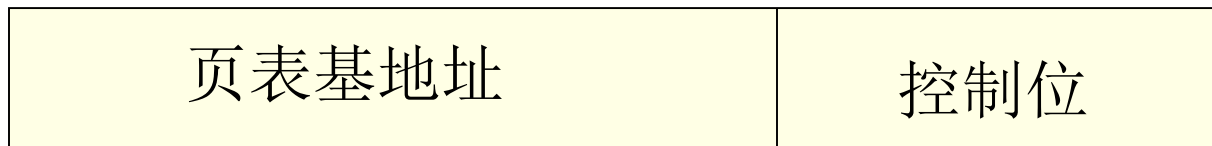
线性地址格式：



10位

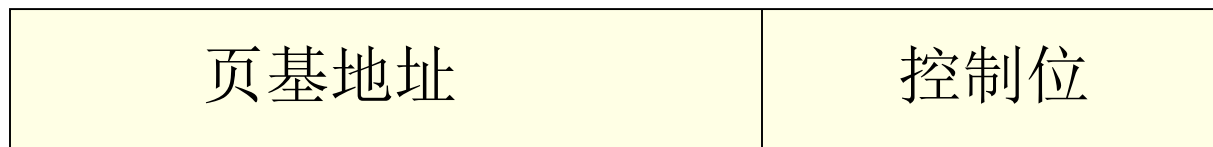
10位

12位



页表目录

20位



页表

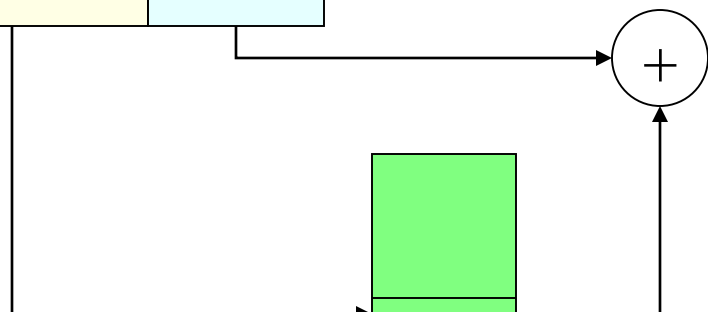
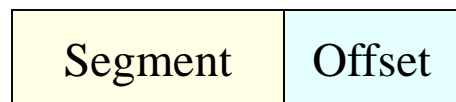
20位

实际地址 = 页基地址 + OFFSET

# Pentium逻辑地址转换

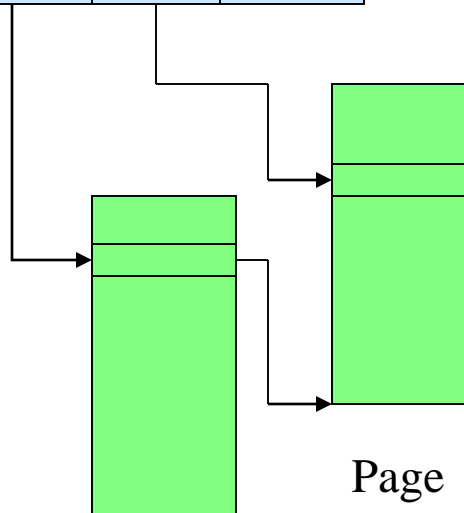


逻辑地址



Segment Table

线性地址



Page  
Directory

Page  
Table

物理  
地址

# 虚拟存储器



- ❖ 虚拟存储器：增加程序员可以使用的地址空间大小。由操作系统进行管理。
- ❖ 管理模式：段式、页式、段页式。
- ❖ 虚实地址转换。
- ❖ 使用快表提高地址转换的速度。

# 阅读和思考



## ✚ 阅读

- ▣ 教材相关章节

## ✚ 思考

- ▣ 是否还有继续使用虚拟存储器的必要?

## ✚ 实践

- ▣ 继续完成实验报告