

计算机组织结构

12 虚拟存储器

刘博涵

2022年11月17日

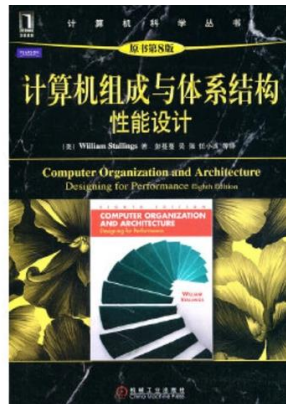


南京大學
NANJING UNIVERSITY

教材对应章节



第7章 存储器分层体系结构



第8章 操作系统支持



操作系统的出现

- 第一台计算机诞生时，采用手工操作的方式
 - 一个用户独占全机：不会出现因资源已被其他用户占用而等待的现象，但资源的利用率低
 - CPU 等待手工操作：CPU的利用不充分
- 批处理系统：加载在计算机上的一个系统软件，在它的控制下，计算机能够自动地、成批地处理一个或多个用户的作业（包括程序、数据和命令）
 - 联机批处理系统，脱机批处理系统

计算机简史

- 第一代：真空管（1946-1957）
 - ENIAC（1946-1955）：第一台通用计算机，十进制，手动编程
 - Electronic Numerical Integrator And Computer
 - ABC（1937）：世界上第一台电子计算机，不可编程
 - Atanasoff-Berry Computer



8

计算机简史

- 第二代：晶体管（1958-1964）
 - NCR和RCA, IBM 7000：晶体管体积更小、更便宜、发热更少，而且能以与电子管相同的方式建造计算机
 - 采用更复杂的算术逻辑单元和控制器，使用高级编程语言，并为计算机提供了系统软件

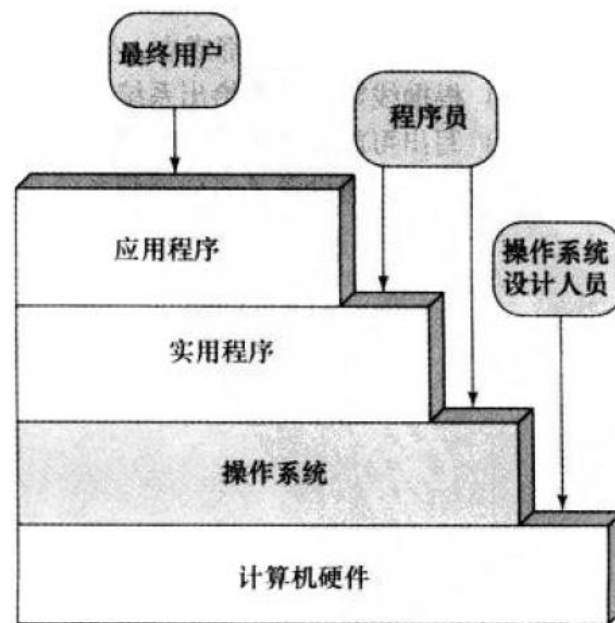
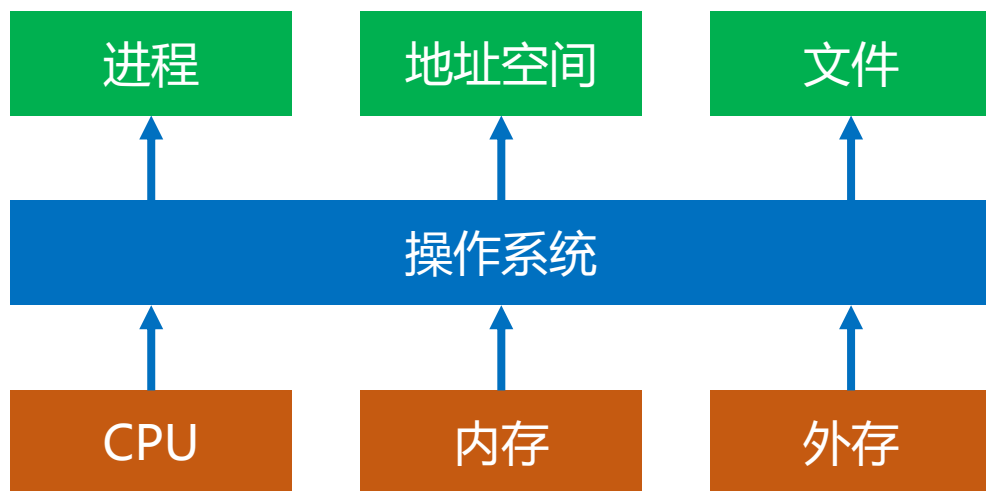


12



操作系统的出现（续）

- **操作系统：**一种控制应用程序运行和在计算机用户与计算机硬件之间提供接口的程序
- **目标**
 - 使计算机使用起来更方便
 - 允许计算机系统的资源以有效的方式使用



存储器管理

- **早期**计算机的主存中仅包含系统软件和一个用户程序
 - 单道程序设计
- **现在**计算机的主存中包含操作系统和若干个用户程序
 - 当所有任务都需要等待I/O时，为了避免处理器处于空闲状态，需要尽可能让更多的任务进入主存
 - 多道程序设计：让处理器一次处理多个任务，提高处理器的利用率
- **存储器管理**
 - 在多道程序系统中，主存需要进一步划分给多个任务，划分的任务由操作系统动态执行
 - 本门课不区分“进程”和“任务”这个更抽象的概念



如何将更多更大的任务装入主存

- 增大主存容量
- 使用交换 (exchange) 技术
 - 当主存中没有处于就绪的任务时，操作系统调入其他任务来执行
 - 分区 (partitioning) 和分页 (paging)
- 虚拟存储器
 - 请求分页：每次访问仅将当前需要的页面调入主存，而其他不活跃的页面放在外存磁盘上
 - 虚拟地址

最低配置:

操作系统: Windows 10 64-bit

处理器: Intel Core i5 6600k or AMD Ryzen 5 1600

内存: 8 GB RAM

8GB内存

显卡: NVIDIA GeForce GTX 1050 Ti or AMD Radeon RX 570

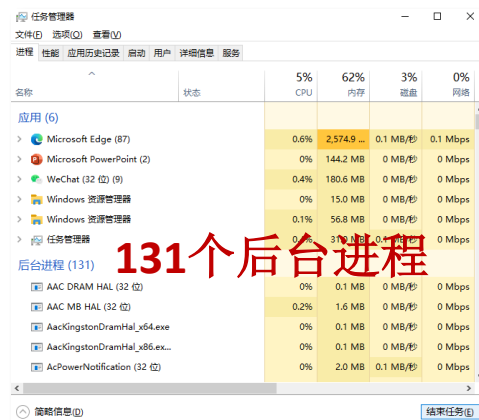
DirectX 版本: 12

网络: 宽带互联网连接

存储空间: 需要 100 GB 可用空间

100GB外存

FIFA23的配置要求



The screenshot shows the Windows Task Manager window. The 'Processes' tab is selected, displaying a list of running applications and background processes. The columns show the name of the process, its status, and its resource usage (CPU, Memory, Disk, Network). The 'Applications' section lists Microsoft Edge, Microsoft PowerPoint, WeChat, Windows Resource Manager, and Windows Task Manager. The 'Background processes' section lists various system services like AAC DRAM HAL, AAC MB HAL, AacKingtonDramHal_x64.exe, AacKingtonDramHal_x86.exe, and AcPowerNotification. A red text overlay '131个后台进程' (131 background processes) is placed over the background processes list.

名称	状态	CPU	内存	磁盘	网络
应用 (6)					
Microsoft Edge (87)		0.6%	2,574.9 ...	0.1 MB/秒	0.1 Mbps
Microsoft PowerPoint (2)		0%	144.2 MB	0 MB/秒	0 Mbps
WeChat (32 位) (9)		0.4%	180.6 MB	0 MB/秒	0 Mbps
Windows 资源管理器		0%	15.0 MB	0 MB/秒	0 Mbps
Windows 任务管理器		0.1%	56.8 MB	0 MB/秒	0 Mbps
后台进程 (131)					
AAC DRAM HAL (32 位)		0%	0.1 MB	0 MB/秒	0 Mbps
AAC MB HAL (32 位)		0.2%	1.6 MB	0 MB/秒	0 Mbps
AacKingtonDramHal_x64.exe		0%	0.1 MB	0 MB/秒	0 Mbps
AacKingtonDramHal_x86.exe		0%	0.1 MB	0 MB/秒	0 Mbps
AcPowerNotification (32 位)		0%	2.0 MB	0.1 MB/秒	0 Mbps

我在做PPT时的任务管理器



分区方式

- 分区方式将主存分为**两大区域**
 - 系统区：**固定**的地址范围内，存放操作系统
 - 用户区：存放所有用户程序
- **简单固定分区**
 - 用户区划分成长度**不等**的**固定**长的分区
 - 当一个任务调入主存时，分配一个**可用的、能容纳它的、最小的**分区
 - 优点：简单
 - 缺点：浪费主存空间



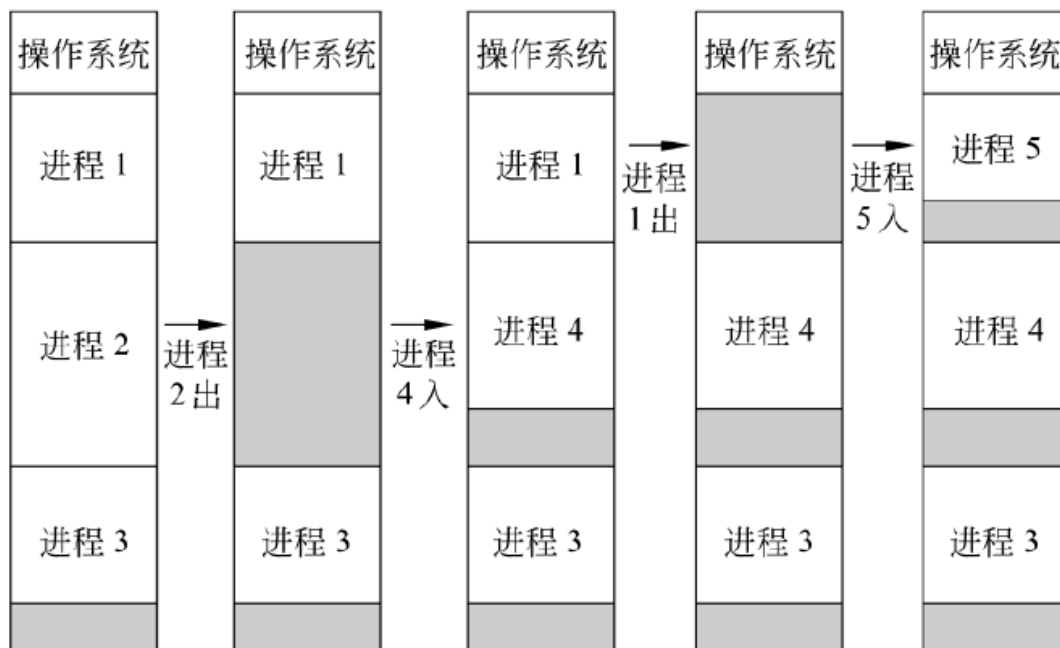
(a) 简单固定分区



分区方式（续）

- **可变长分区**

- **用户区**按每个任务所需要的内存大小进行分配
- 优点：提高了主存的**利用率**
- 缺点：时间越长，存储器中的**碎片**就会越多

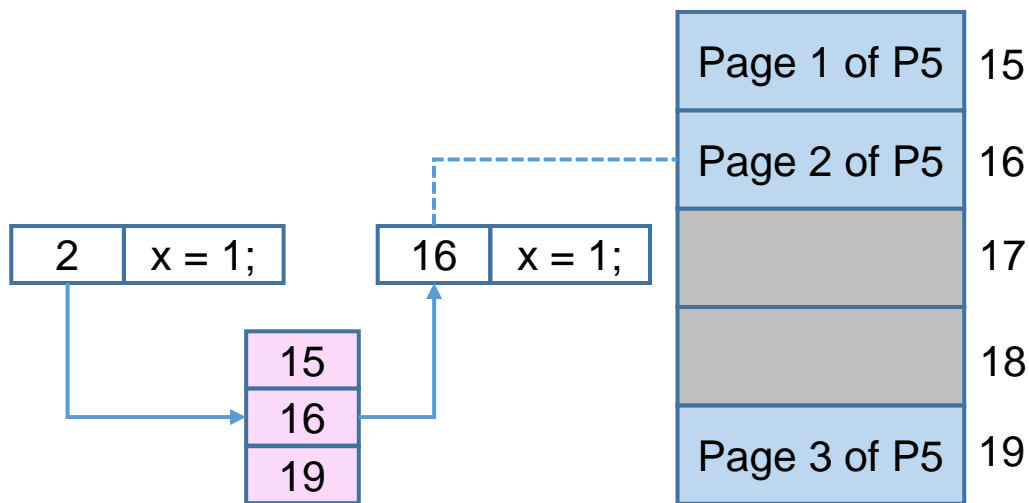


(b) 可变长分区



分页方式

- 目的：减少碎片的产生
- 基本思想
 - 把主存分成固定长且比较小的存储块，称为页框 (page frame) ，每个任务也被划分成固定长的程序块，称为页 (page)
 - 将页装入页框中，且无需采用连续的页框来存放一个任务中所有的页
- 逻辑地址：指令中的地址
- 物理地址：实际主存地址



页表

虚拟存储器

- 问题

- 内存的大小是有限的，但对内存的需求不断增加

- 基本思想

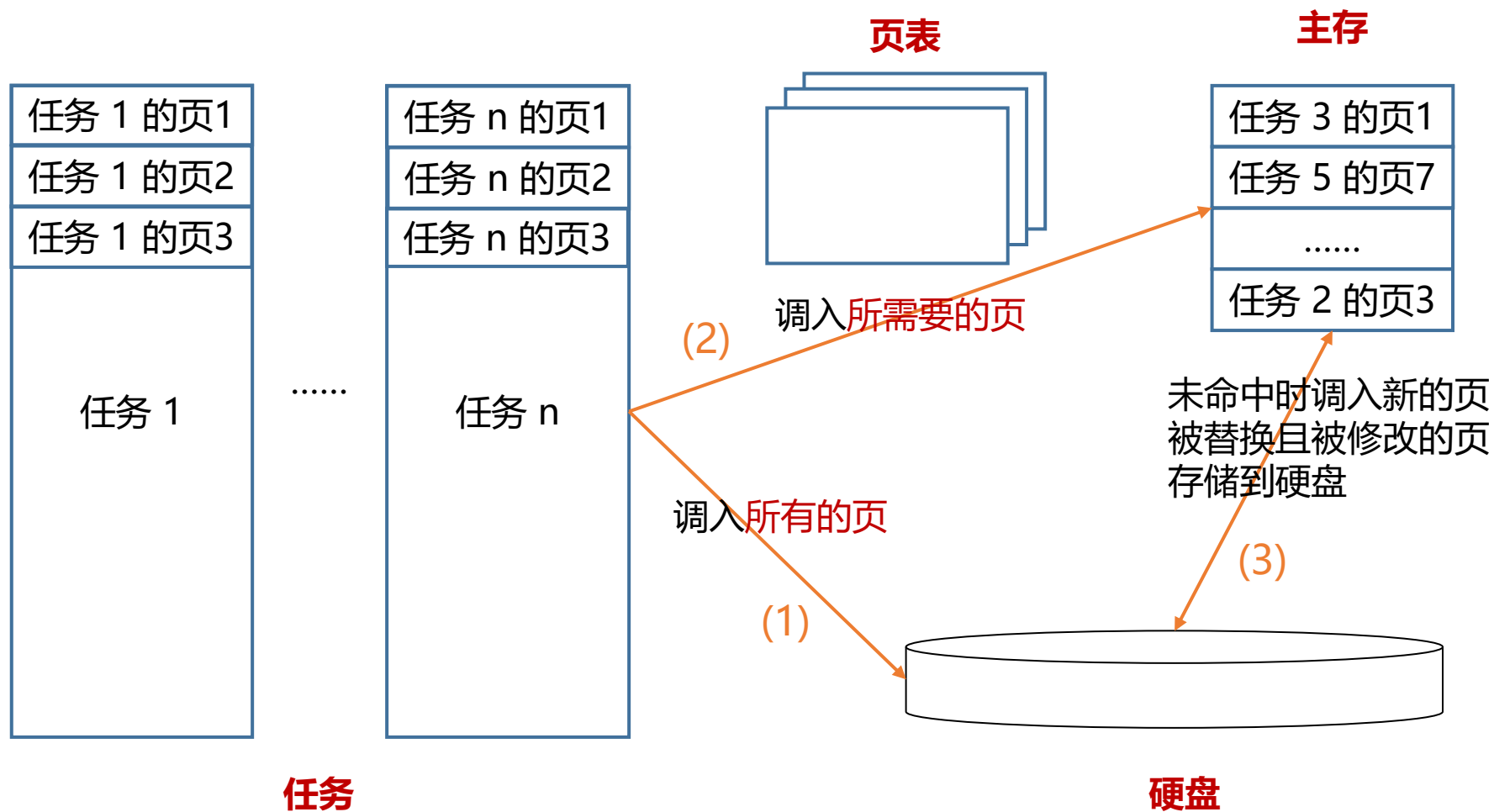
- **请求分页**：仅将当前需要的的**页**调入主存
 - 通过硬件将逻辑地址转换为物理地址
 - 未命中时在主存和硬盘之间交换信息

- 优点

- 在不扩大物理内存的前提下，可以载入更多的任务
- 编写程序时不需要考虑可用物理内存的状态
 - 程序员认为可以独享一个连续的、很大的内存
- 可以在大于物理内存的逻辑地址空间中编程



虚拟存储器：流程

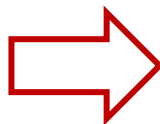


虚拟存储器：设计的一些问题

- 页大小
 - 4KB, 8KB, ...
- 映射算法
 - 全相联映射
- 写策略
 - 写回
- 类型
 - 分页式虚拟存储器
 - 分段式虚拟存储器
 - 段页式虚拟存储器

Cache比主存快10倍

主存比硬盘快100000多倍



减少硬盘访问，提高主存命中率



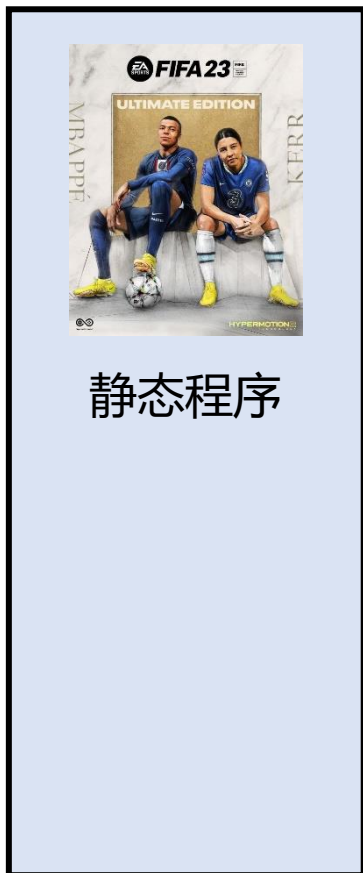
分页式虚拟存储器

- 主存储器和虚拟地址空间都被划分为**大小相等**的页面
 - 虚拟页 (virtual page, VP) / 逻辑页 (logical page) : 虚拟地址空间中的页面
 - 物理页 (physical page, PP) / 页框 (page frame) : 主存空间中的页面
- 页表
 - 页表中包含了**所有虚拟页**的信息, 包括虚拟页的存放位置、装入位 (valid)、修改位 (dirty)、存取权限位等等
 - 保存在主存中
 - 虚拟地址
 - 虚拟页号 + 页内偏移量



分页式虚拟存储器

硬盘



1、启动硬盘中
存储的程序

硬盘



2、载入运行时
程序到硬盘中

主存



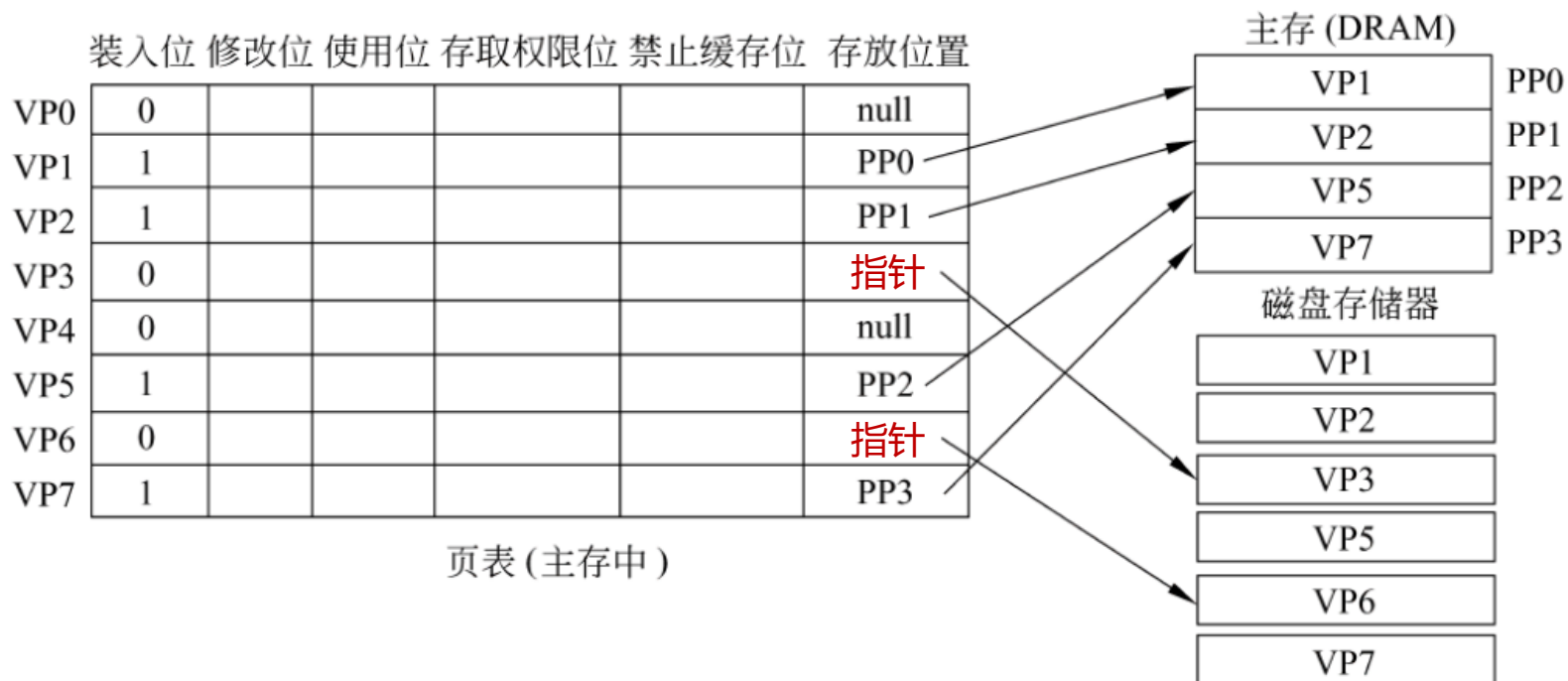
3、将需要的页
载入内存

分页式虚拟存储器：页表

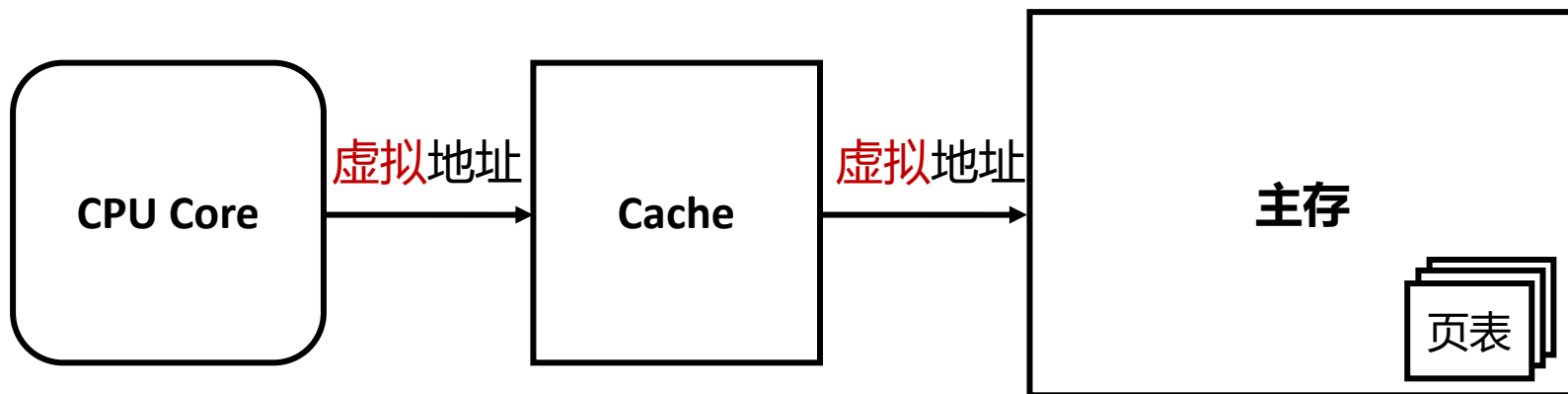
根据页表中记录的物理页存放位置，可以将虚拟地址转化为物理地址

$\text{虚拟页号} + \text{页内偏移量} \rightarrow \text{物理页号} + \text{页内偏移量}$

虚拟地址 物理地址



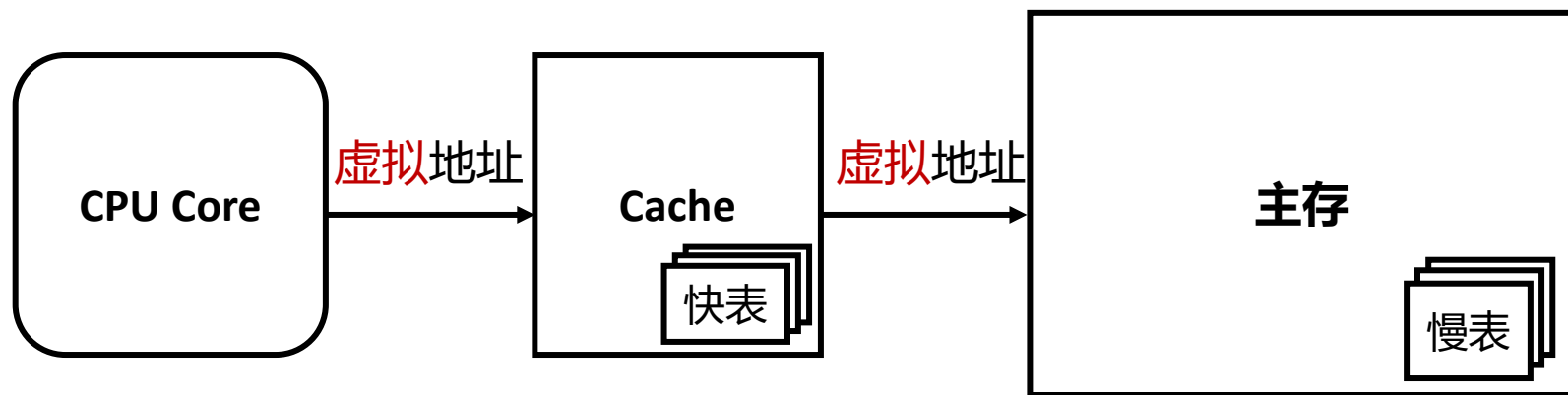
分页式虚拟存储器：问题



**页表在主存中，每次访问主存都需要查页表
Cache不就失效了吗？**

快表 (Translation Lookaside Buffer, TLB)

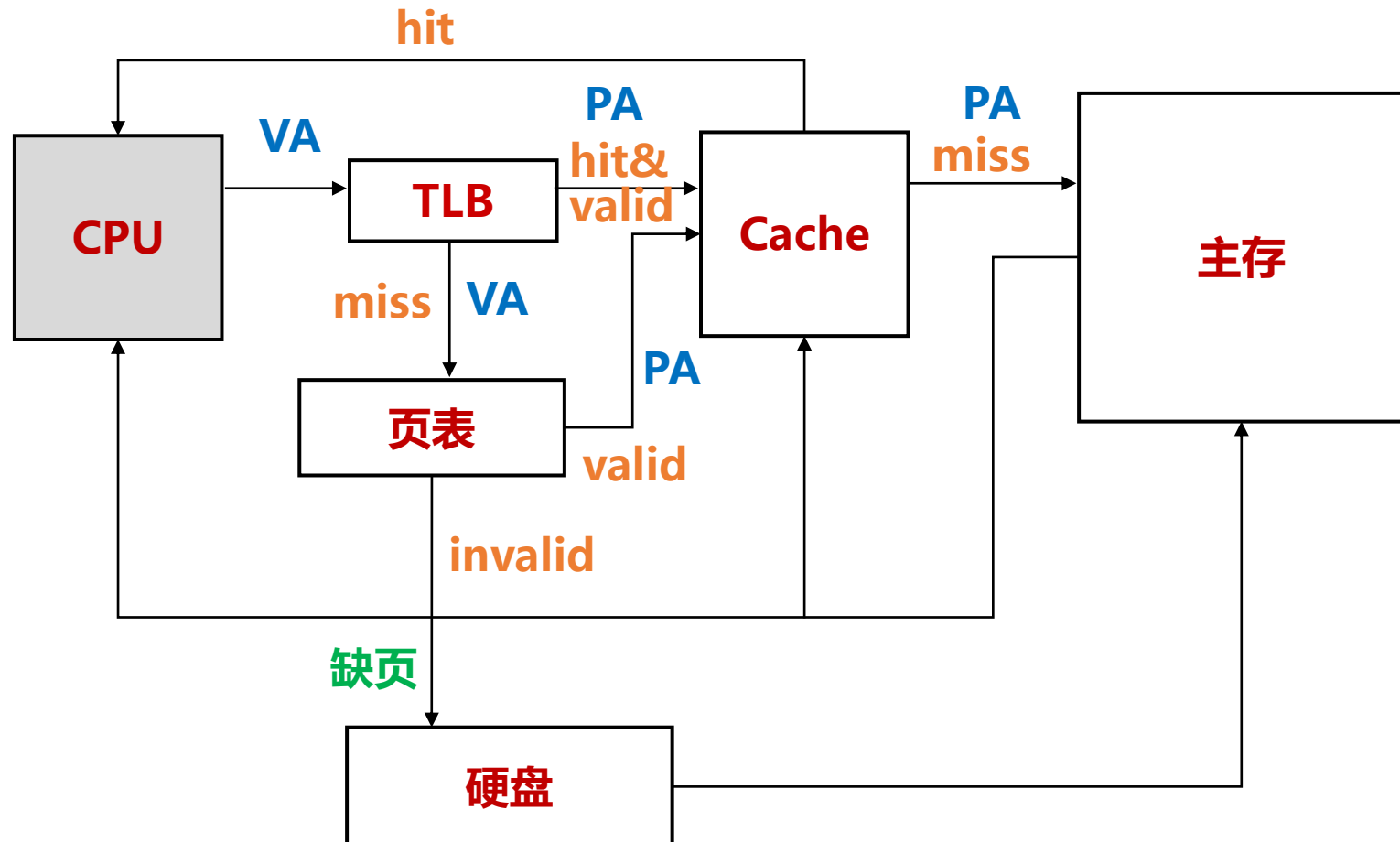
- 页表的使用增加了主存的访问次数
- 为了减少访存次数，把页表中**最活跃的几个页表项**复制到高速缓存中
- 后备转换缓冲器（简称“快表”）：将页表项放入高速缓存中
 - 映射：关联映射，组关联映射
 - 替换：随机替换
- 主存中的页表相应地称之为“慢表”



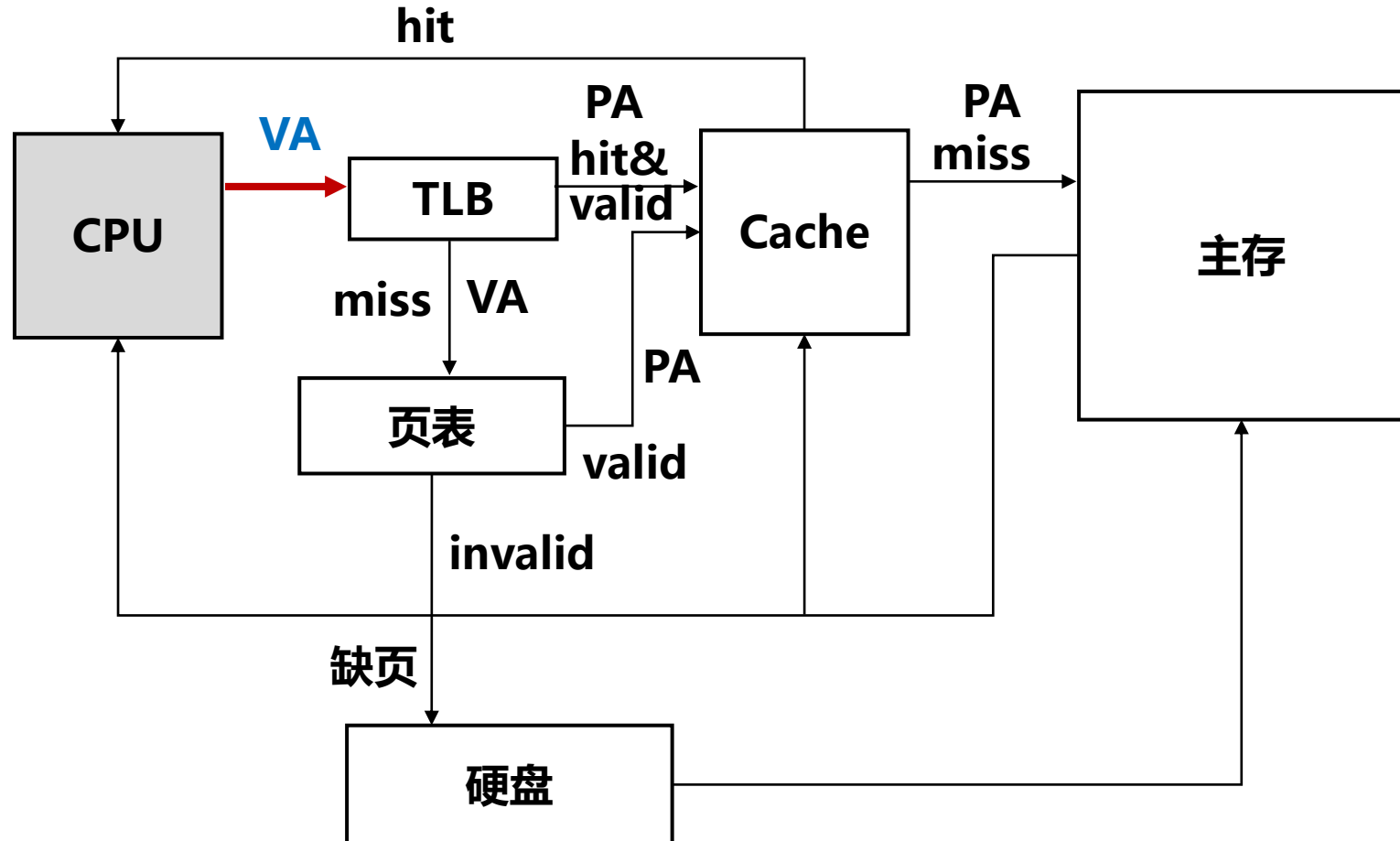
注：现在的计算机中，TLB和Cache在物理上是两个存储器



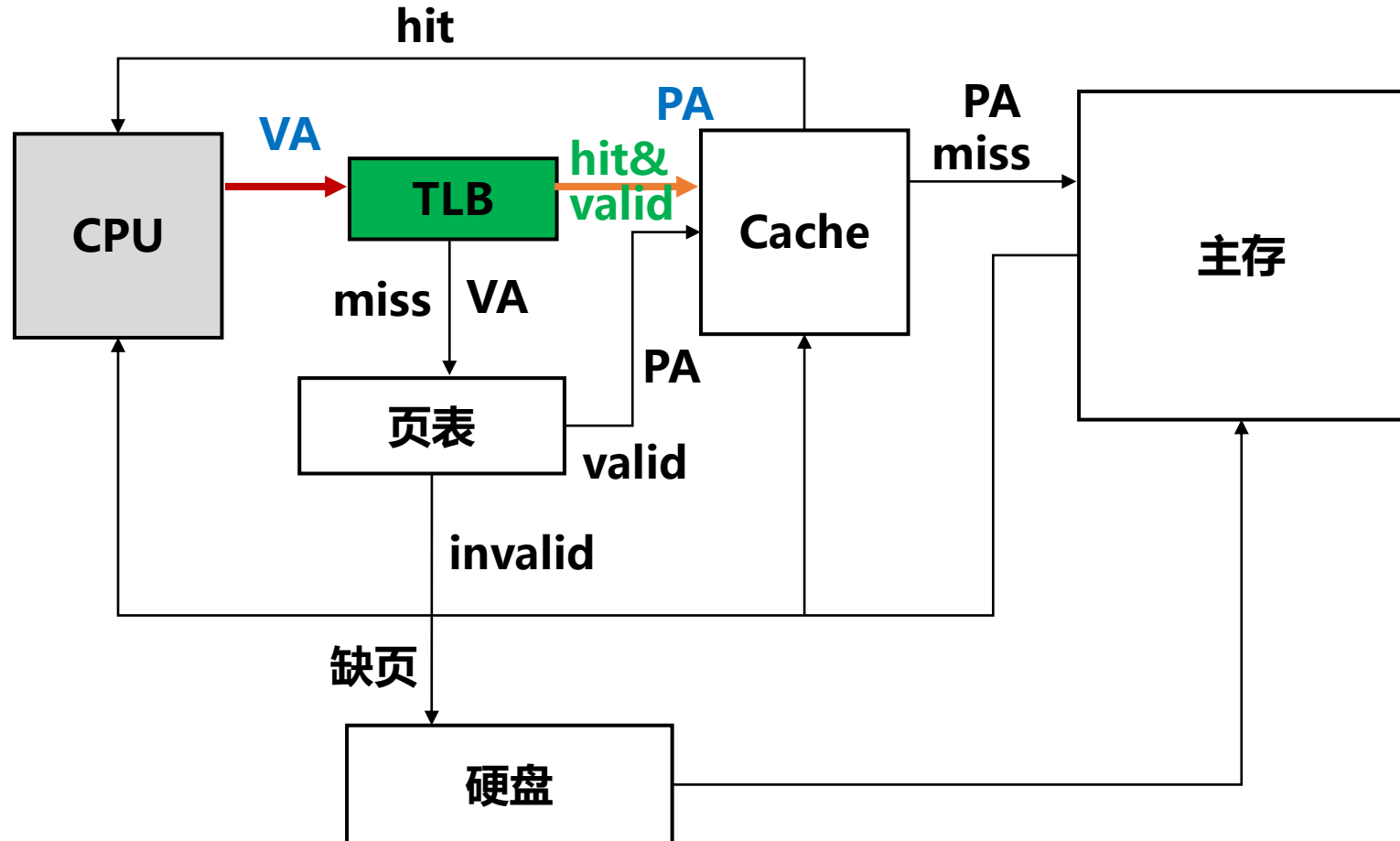
CPU访存过程



CPU访存过程 (续)



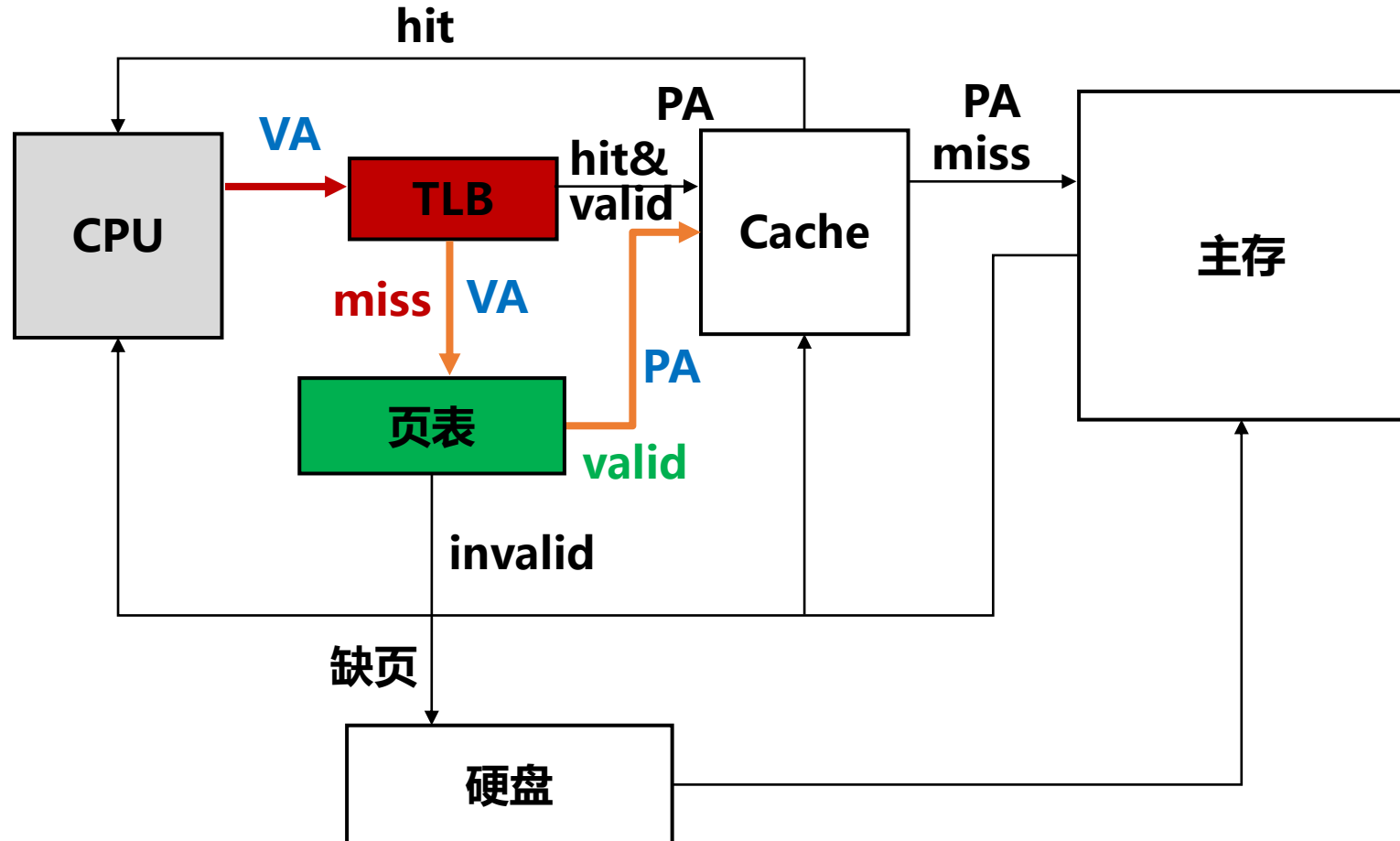
CPU访存过程 (续)



➤ TLB命中



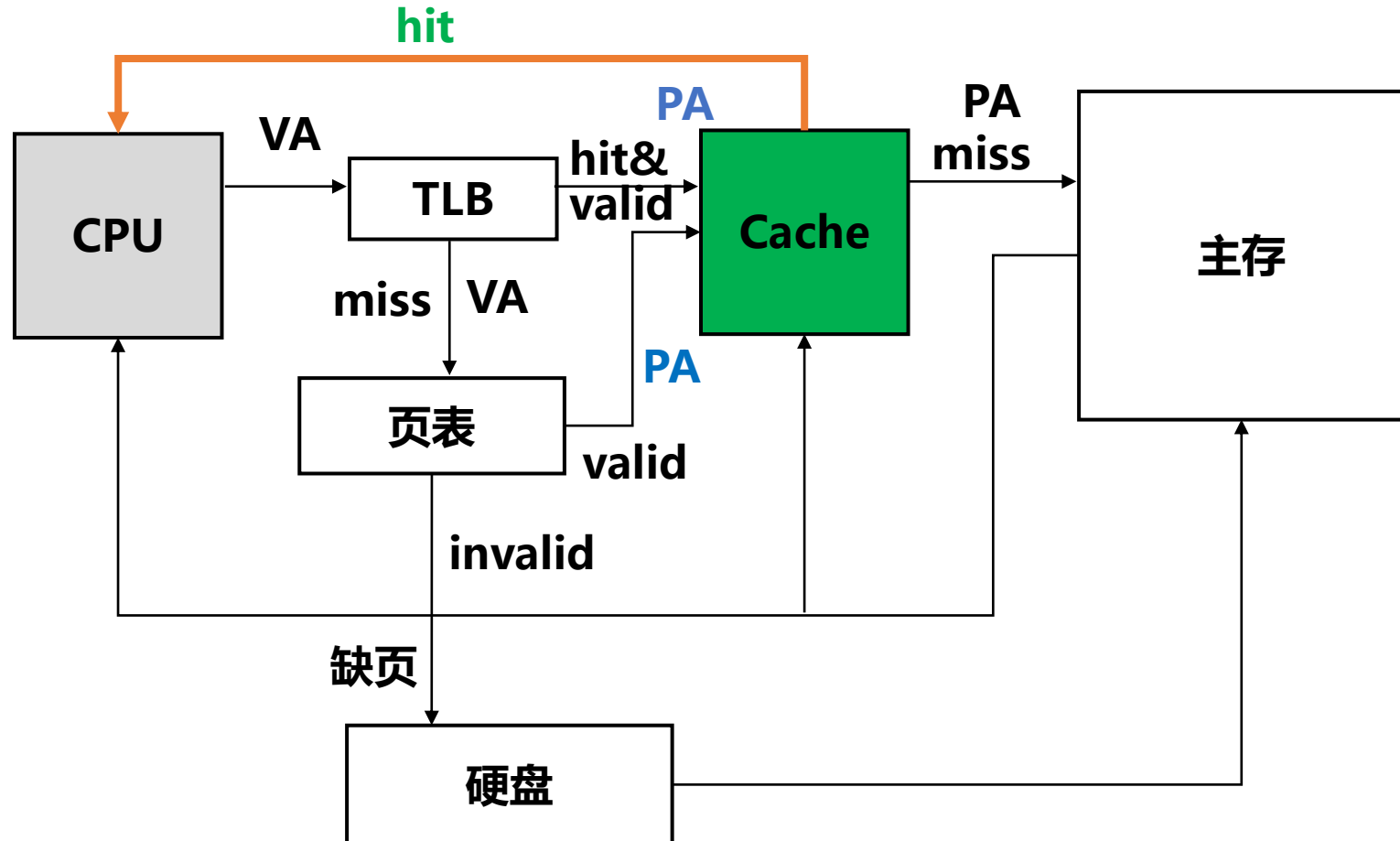
CPU访存过程 (续)



➤ TLB未命中，页表命中



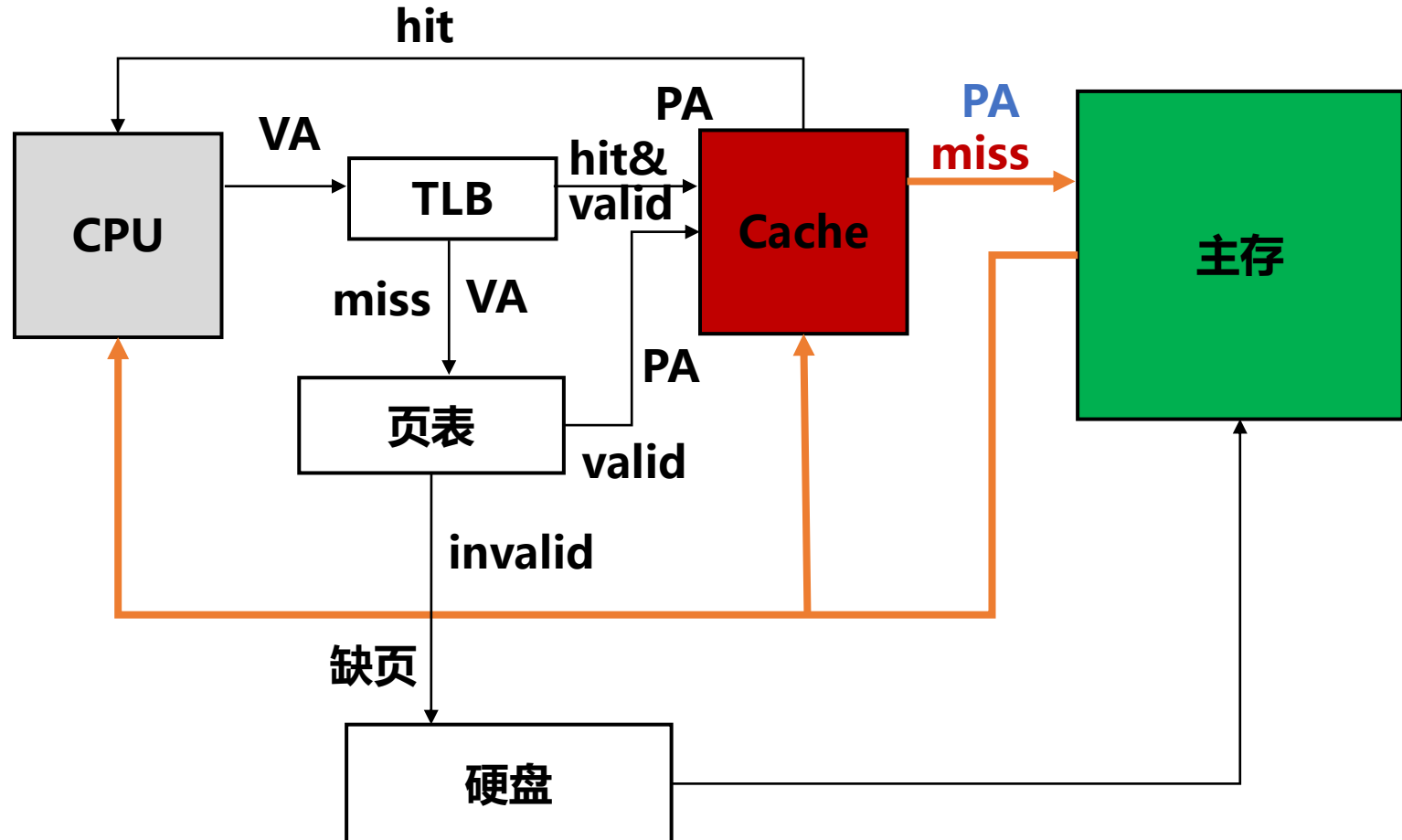
CPU访存过程 (续)



- TLB或页表命中
- Cache命中



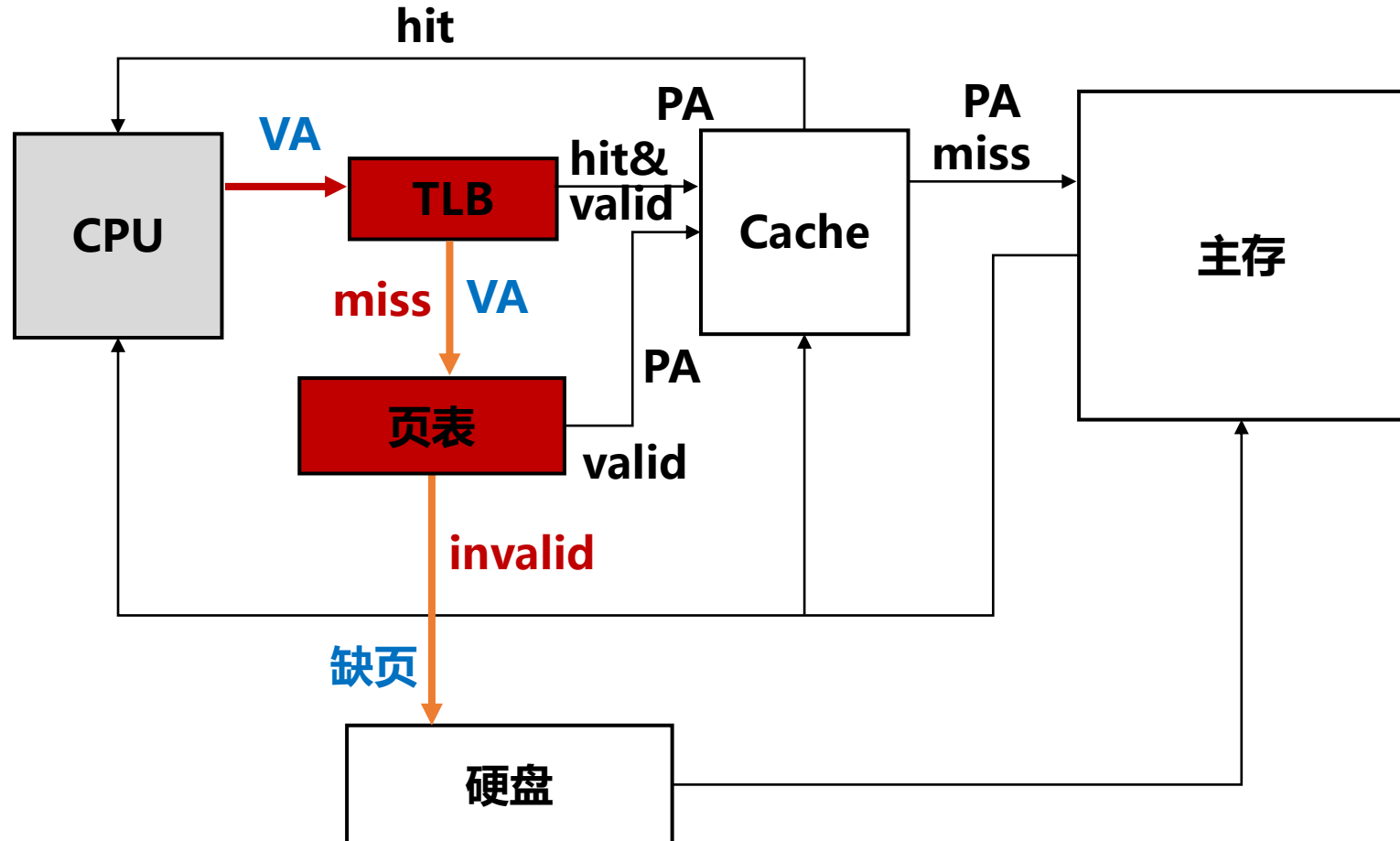
CPU访存过程 (续)



- TLB或页表命中
- Cache未命中



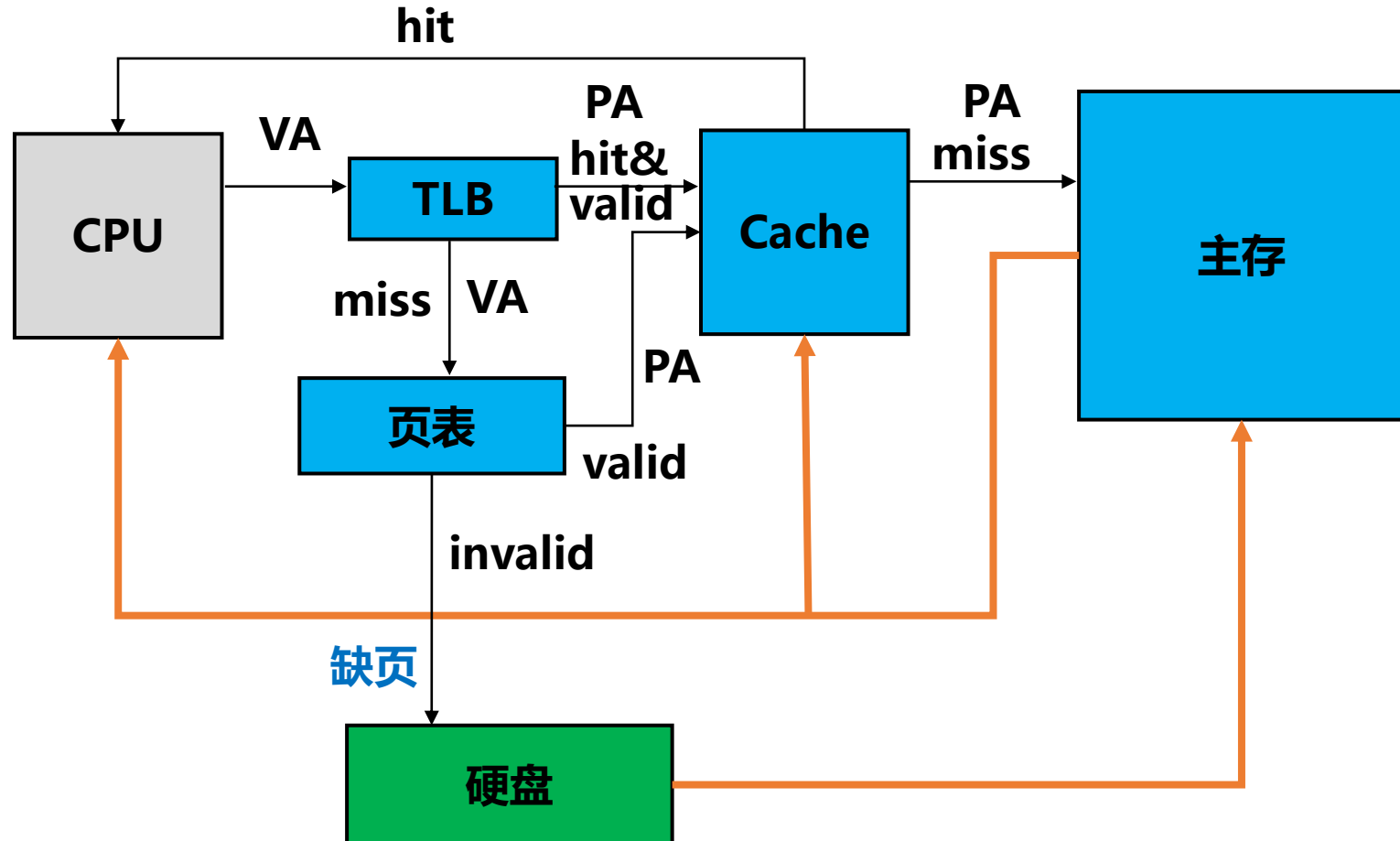
CPU访存过程 (续)



➤ 页表未命中



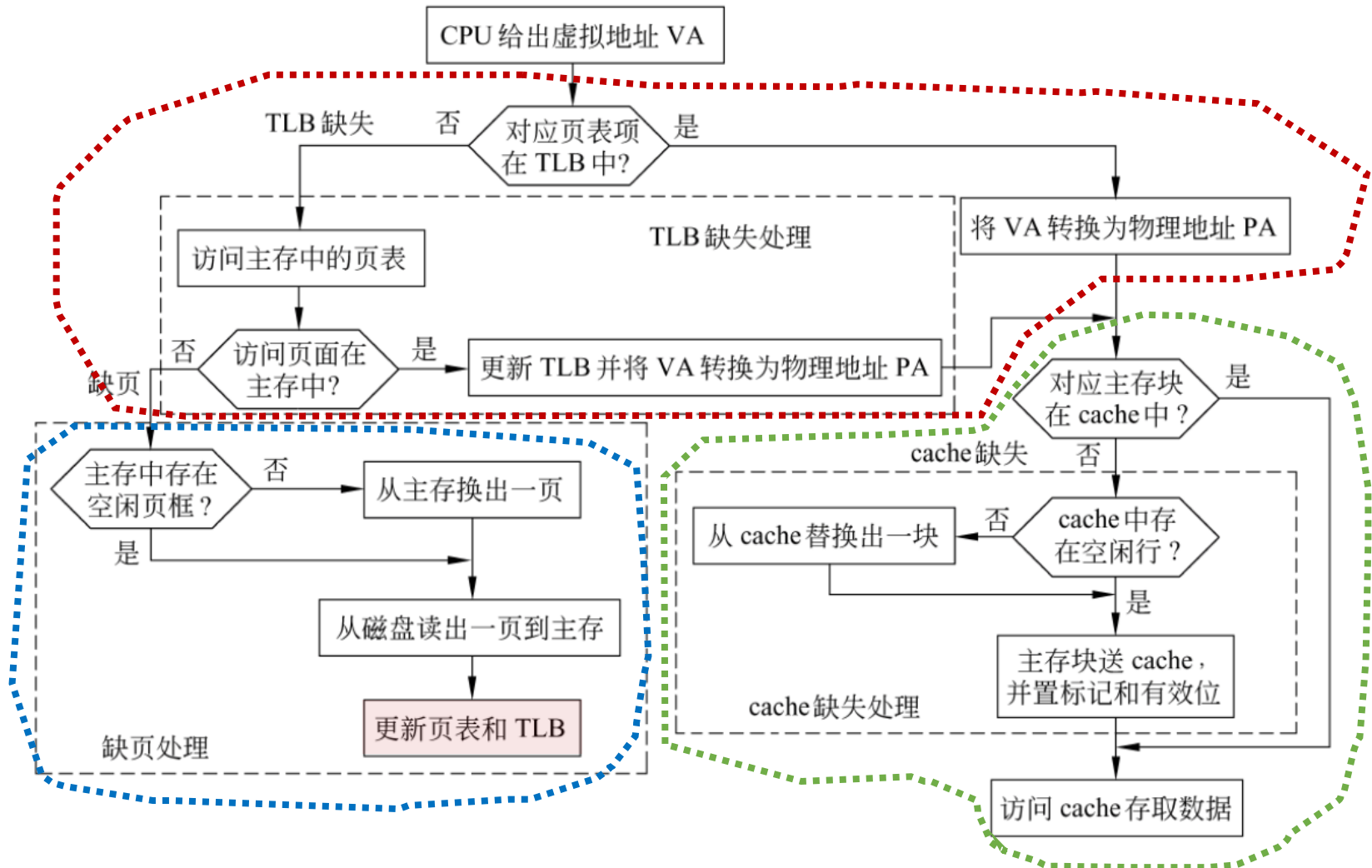
CPU访存过程 (续)



- 页表未命中
- 更新



CPU访存过程：小结



TLB、页表、Cache的缺失组合

序号	TLB	page	cache	说 明
1	hit	hit	hit	可能, TLB 命中则页一定命中, 信息在主存, 就可能在 cache 中
2	hit	hit	miss	可能, TLB 命中则页一定命中, 信息在主存, 但可能不在 cache 中
3	miss	hit	hit	可能, TLB 缺失但页可能命中, 信息在主存, 就可能在 cache 中
4	miss	hit	miss	可能, TLB 缺失但页可能命中, 信息在主存, 但可能不在 cache 中
5	miss	miss	miss	可能, TLB 缺失, 则页也可能缺失, 信息不在主存, 一定也不在 cache
6	hit	miss	miss	不可能, 页缺失, 说明信息不在主存, TLB 中一定没有该页表项
7	hit	miss	hit	不可能, 页缺失, 说明信息不在主存, TLB 中一定没有该页表项
8	miss	miss	hit	不可能, 页缺失, 说明信息不在主存, cache 中一定也没有该信息

最好
好
差
最差

1
2, 3
4
5

无须访问主存
访问1次主存
访问2次主存
访问2次主存, 且需要访问硬盘



分段式虚拟存储器

- 将程序和数据分成不同长度的段，将所需的段加载到主存中
- 虚拟地址：段号 + 段内偏移量
- 与分页式虚拟存储器相比
 - 分页式虚拟存储器（页对程序员不可见）
 - 优点：实现简单、开销少
 - 缺点：一个数据或一条指令可能会分跨在两个页面
 - 分段式虚拟存储器（段对程序员可见）
 - 优点：段的分界与程序的自然分界相对应，易于编译、管理、修改和保护
 - 缺点：段的长度不固定（引起碎片）



段页式虚拟存储器

- 将程序和数据分段，段内再进行分页
 - 每个分段都有一个页表
- 虚拟地址
 - 段号 + 页号 + 页内偏移量
- 优点
 - 程序按段实现共享与保护
- 缺点
 - 需要多次查表



总结

- 存储器管理
- 交换技术
 - 分区
 - 分页
- 虚拟存储器
 - 分页式虚拟存储器
 - 页表，虚拟地址和物理地址的转换
 - CPU访存过程
 - TLB、页表、Cache的缺失组合
 - 分段式虚拟存储器
 - 段页式虚拟存储器



谢谢

bohanliu@nju.edu.cn



南京大學
NANJING UNIVERSITY