

本节内容

# 请求分页管理方式

# 知识总览

请求分页存储管理与基本分页存储管理的主要区别：

在程序执行过程中，当所访问的信息不在内存时，由操作系统负责将所需信息从外存调入内存，然后继续执行程序。

若内存空间不够，由操作系统负责将内存中暂时用不到的信息换出到外存。

操作系统要提供页面置换的功能，将暂时用不到的页面换出外存

操作系统要提供请求调页功能，将缺失页面从外存调入内存

## 请求分页管理方式

页表机制

缺页中断机构

地址变换机构

注意与基本分页存储管理的页表机制、地址变换流程对比学习

# 页表机制



与基本分页管理相比，请求分页管理中，为了实现“请求调页”，操作系统需要知道每个页面是否已经调入内存；如果还没调入，那么也需要知道该页面在外存中存放的位置。

当内存空间不够时，要实现“页面置换”，操作系统需要通过某些指标来决定到底换出哪个页面；有的页面没有被修改过，就不用再浪费时间写回外存。有的页面修改过，就需要将外存中的旧数据覆盖，因此，操作系统也需要记录各个页面是否被修改的信息。

请求页表项增加了四个字段

是否已调入内存

可记录最近被访问过几次，或记录上次访问的时间，供置换算法选择换出页面时参考

页面调入内存后是否被修改过

页面在外存中的存放位置

页号	内存块号
0	a
1	b
2	c

基本分页存储管理的页表

王道24考研交流群：769832062

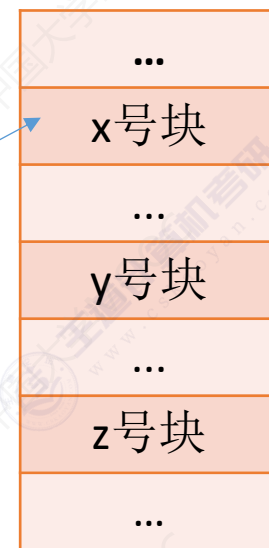
页号	内存块号	状态位	访问字段	修改位	外存地址
0	无	0	0	0	x
1	b	1	10	0	y
2	c	1	6	1	z

请求分页存储管理的页表

王道考研/CSKAOYAN.COM

## 缺页中断机构

页号	内存块号	状态位	访问字段	修改位	外存地址
0	a	1	0	0	x
1	b	1	10	0	y
2	c	1	6	1	z



外存



内存

假设此时要访问逻辑地址=（页号，页内偏移量）=（0, 1024）

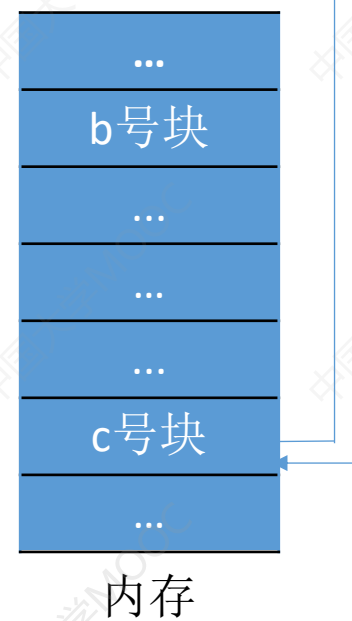
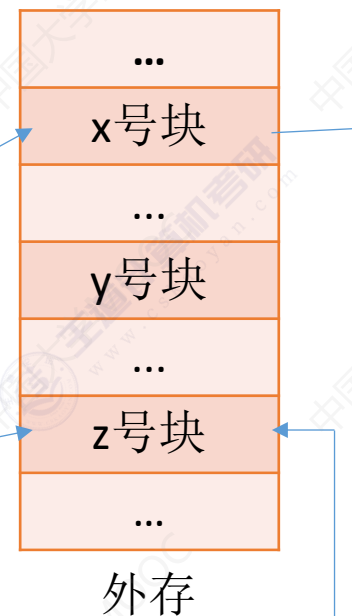
在请求分页系统中，每当要访问的页面不在内存时，便产生一个缺页中断，然后由操作系统的缺页中断处理程序处理中断。

此时缺页的进程阻塞，放入阻塞队列，调页完成后将其唤醒，放回就绪队列。

如果内存中有空闲块，则为进程分配一个空闲块，将所缺页面装入该块，并修改页表中相应的页表项。

## 缺页中断机构

页号	内存块号	状态位	访问字段	修改位	外存地址
0	c	1	0	0	x
1	b	1	10	0	y
2	无	0	0	0	z



假设此时要访问逻辑地址=（页号，页内偏移量）=（0, 1024）

在请求分页系统中，每当要访问的页面不在内存时，便产生一个缺页中断，然后由操作系统的缺页中断处理程序处理中断。

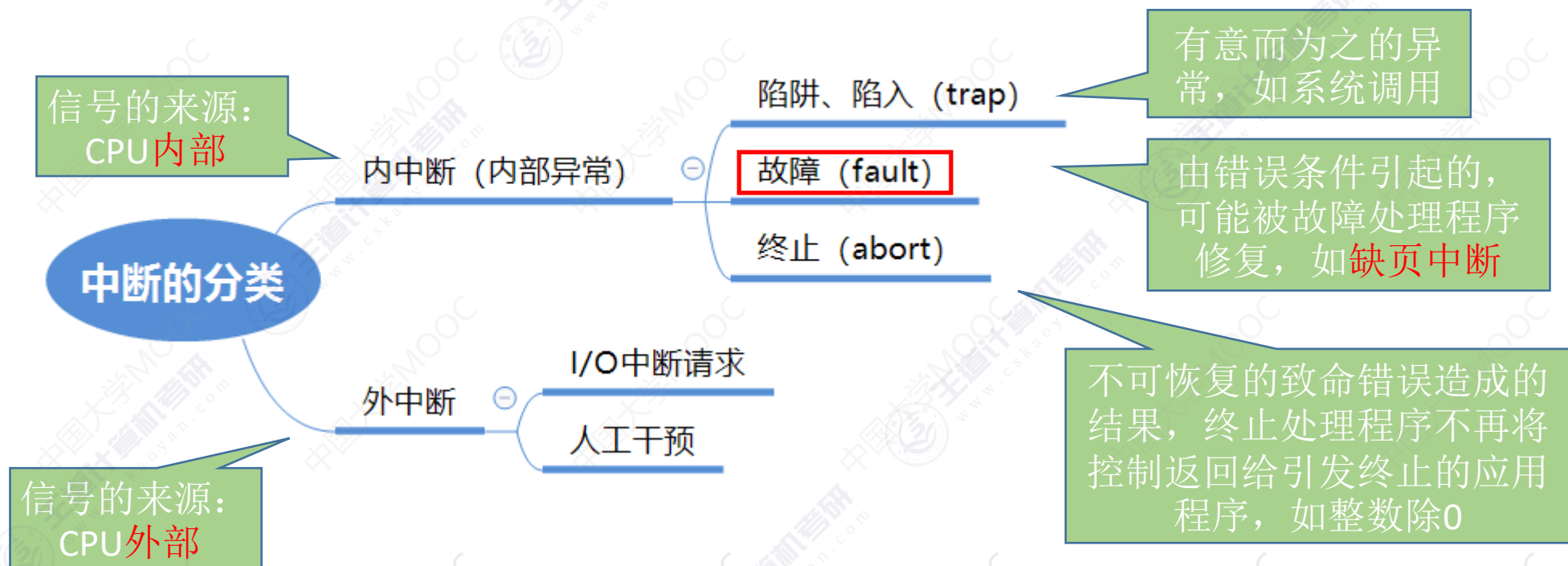
此时缺页的进程阻塞，放入阻塞队列，调页完成后将其唤醒，放回就绪队列。

如果内存中有空闲块，则为进程分配一个空闲块，将所缺页面装入该块，并修改页表中相应的页表项。

如果内存中没有空闲块，则由页面置换算法选择一个页面淘汰，若该页面在内存期间被修改过，则要将其写回外存。未修改过的页面不用写回外存。

# 缺页中断机构

缺页中断是因为当前执行的指令想要访问的目标页面未调入内存而产生的，因此属于内中断。一条指令在执行期间，可能产生多次缺页中断。（如：copy A to B，即将逻辑地址A中的数据复制到逻辑地址B，而A、B属于不同的页面，则有可能产生两次中断）





## 地址变换机构

请求分页存储管理与基本分页存储管理的主要区别：

在程序执行过程中，当所访问的信息不在内存时，由操作系统负责将所需信息从外存调入内存，然后继续执行程序。

若内存空间不够，由操作系统负责将内存中暂时用不到的信息换出到外存。

操作系统要提供页面置换的功能，  
将暂时用不到的页面换出外存

操作系统要提供  
请求调页功能，  
将缺失页面从外  
存调入内存

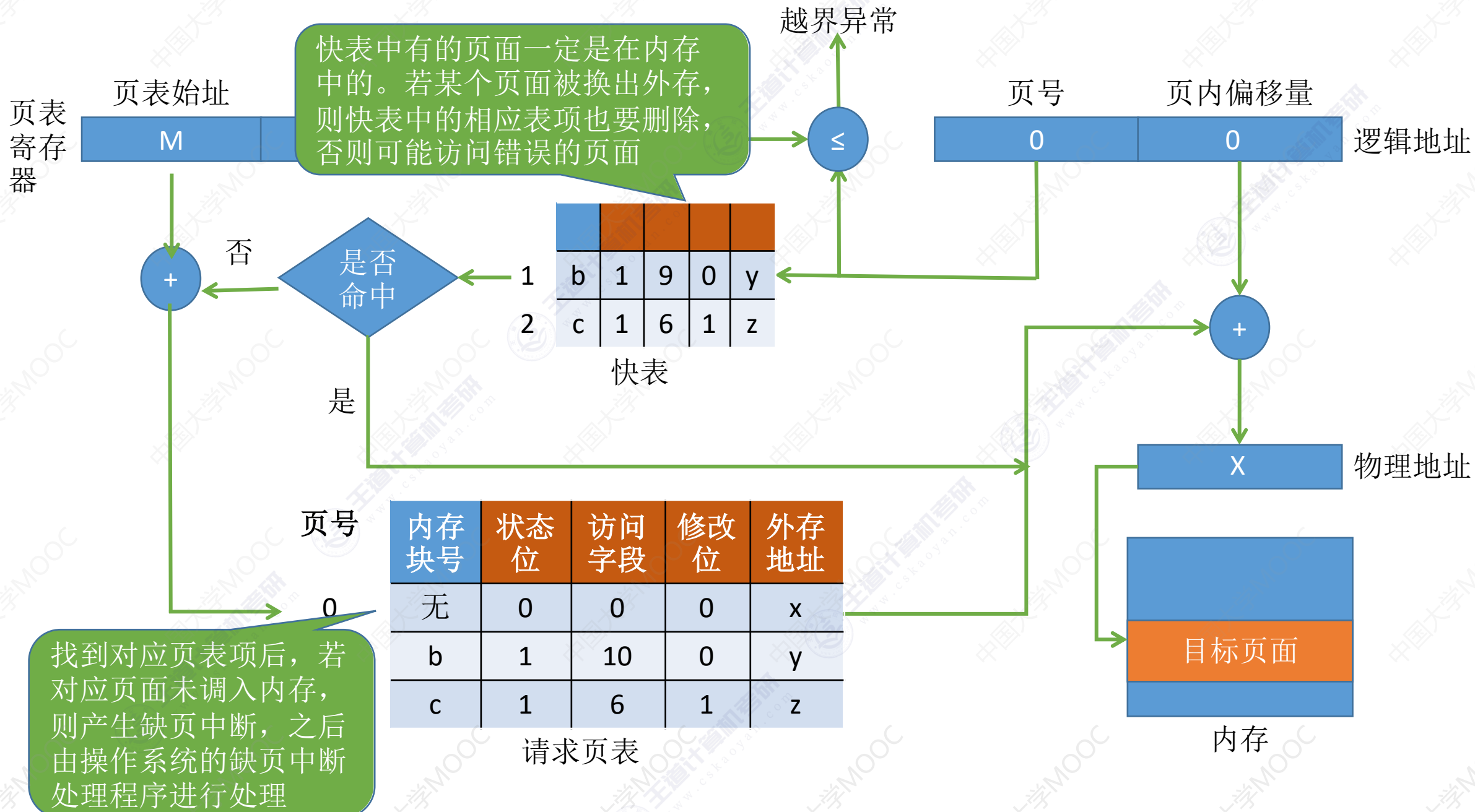
页号	内存块号	状态位	访问字段	修改位	外存地址
0	无	0	0	0	x
1	b	1	10	0	y
2	c	1	6	1	z



新增步骤1：请求调页（查到页表项时进行判断）

新增步骤2：页面置换（需要调入页面，但没有空闲内存块时进行）

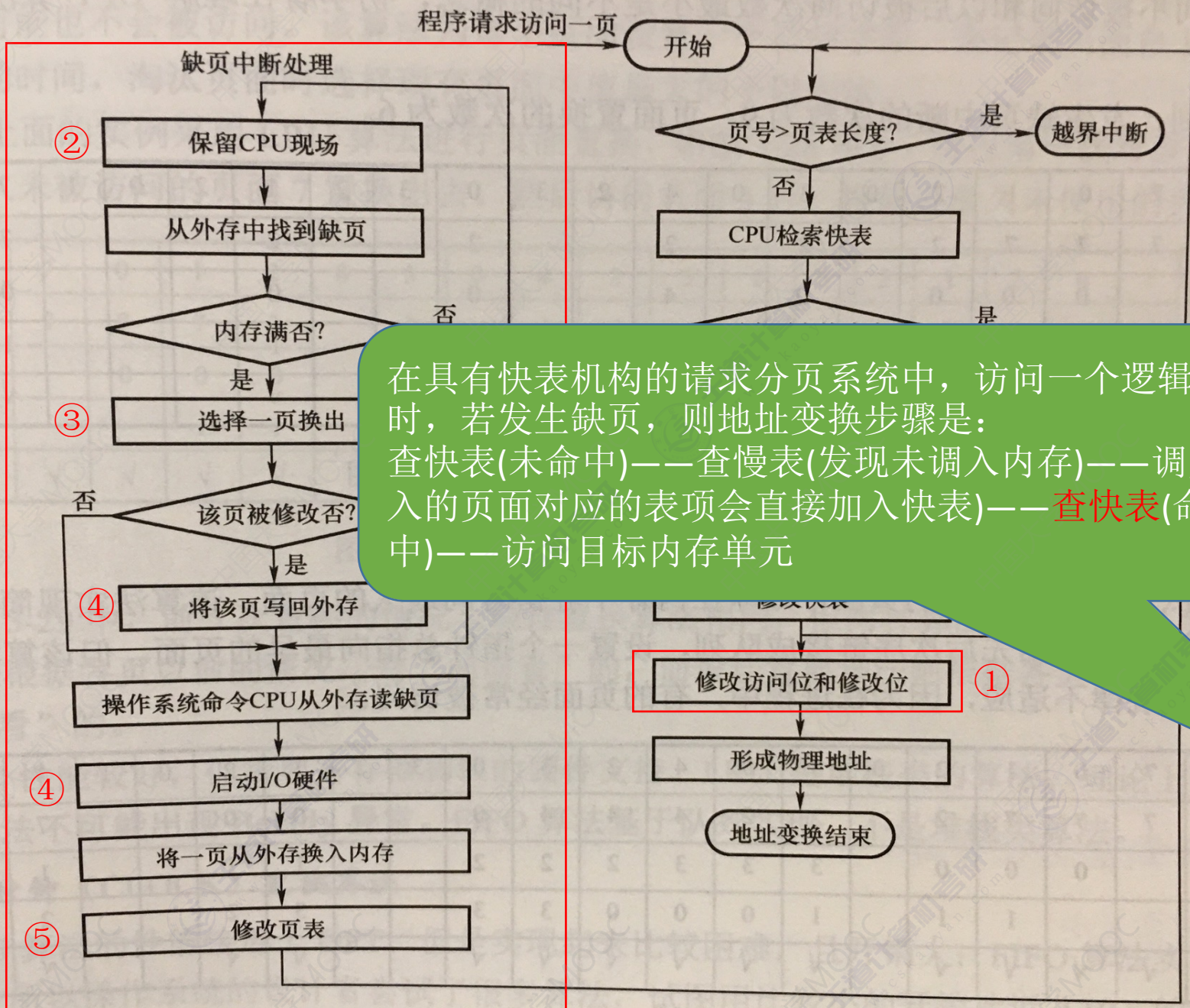
新增步骤3：需要修改请求页表中新增的表项



快表中有的页面一定是在内存中的。若某个页面被换出外存，则快表中的相应表项也要删除，否则可能访问错误的页面

找到对应页表项后，若对应页面未调入内存，则产生缺页中断，之后由操作系统的缺页中断处理程序进行处理





补充细节：

①只有“写指令”才需要修改“修改位”。并且，一般来说只需修改快表中的数据，只有要将快表项删除时才需要写回内存中的慢表。这样可以减少访存次数。

②和普通的中断处理一样，缺页中断处理依然需要保留CPU现场。

③需要用某种“页面置换算法”来决定一个换出页面（下节内容）

④换入/换出页面都需要启动慢速的I/O操作，可见，如果换入/换出太频繁，会有很大的开销。

⑤页面调入内存后，需要修改慢表，同时也需要将表项复制到快表中。



# 知识回顾与重要考点

## 请求分页管理方式

### 页表机制

在基本分页的基础上增加了几个表项

状态位：表示页面是否已在内存中

访问字段：记录最近被访问过几次，或记录上次访问的时间，供置换算法选择换出页面时参考

修改位：表示页面调入内存后是否被修改过，只有修改过的页面才需在置换时写回外存

外存地址：页面在外存中存放的位置

### 缺页中断机构

找到页表项后检查页面是否已在内存，若没在内存，产生缺页中断

缺页中断处理中，需要将目标页面调入内存，有必要时还要换出页面

缺页中断属于内中断，属于内中断中的“故障”，即可能被系统修复的异常

一条指令在执行过程中可能产生多次缺页中断

### 地址变换机构（重点关注与基本分页不同的地方）

找到页表项是需要检查页面是否在内存中

若页面不再内存中，需要请求调页

若内存空间不够，还需换出页面

页面调入内存后，需要修改相应页表项



公众号：王道在线



b站：王道计算机教育



抖音：王道计算机考研