



# 课程主要内容

---

👉 操作系统引论（第1章）

👉 处理机管理（第2-3章）

👉 存储器管理（第4-5章）

👉 设备管理（第6章）

👉 文件管理（第7-8章）

👉 操作系统接口（第9章）



# 第5章 虚拟存储器

---

存储器是计算机系统的重要组成部分，是计算机系统的一种宝贵而紧俏的资源。操作系统中的存储管理是指对内存的管理，它是操作系统的重要功能之一。

存储管理的主要任务是为多道程序的运行提供良好的环境，方便用户使用存储器，提高存储器的利用率以及从逻辑上扩充存储器。为此

**存储管理应具有以下功能：**

- 实现内存的分配和回收
- 地址变换
- “扩充”内存容量
- 进行存储保护



## 5.1 虚拟存储器的基本概念

### □ 常规存储管理方式的共同点：

要求一个作业全部装入内存后方能运行。

### □ 问题：

(1) 有的作业很大, 所需内存空间大于内存总容量, 使作业无法运行。

(2) 有大量作业要求运行, 但内存容量不足以容纳下所有作业, 只能让一部分先运行, 其它在外存等待。

### □ 解决方法 (1) 增加内存容量。

(2) 从逻辑上扩充内存容量

----覆盖

----对换

----虚拟存储器



# 一、虚拟存储器的引入(1)

---

## ❖ 常规存储器管理方式的特征

(1) 一次性:

作业在运行前需一次性地全部装入内存。将导致上述两问题。

(2) 驻留性:

作业装入内存后, 便一直驻留内存, 直至作业运行结束。



# 一、虚拟存储器的引入(2)

## ❖ 局部性原理

指程序在执行时呈现出局部性规律，即在一较短时间内，程序的执行仅限于某个部分，相应地，它所访问的存储空间也局限于某个区域。

局部性又表现为**时间局部性**（由于大量的循环操作，某指令或数据被访问后，则不久可能会被再次访问）和**空间局部性**（如顺序执行，指程序在一段时间内访问的地址，可能集中在一定的范围之内）。



# 虚拟存储器的概念（1）

- ◆ 基于局部性原理，程序在运行之前，没有必要全部装入内存，仅须将当前要运行的页（段）装入内存即可。
- ◆ 运行时，如访问的页（段）在内存中，则继续执行，如访问的页（段）未在内存中（缺页或缺段），则利用OS的请求调页（段）功能，将该页（段）调入内存。
- ◆ 如内存已满，则利用OS的页（段）置换功能，按某种置换算法将内存中的某页（段）调至外存，从而调入需访问的页。



## 虚拟存储器的概念（2）

**虚拟存储器**是指仅把作业的一部分装入内存便可运行作业的存储管理系统，它具有请求调页功能和页面置换功能，能从逻辑上对内存容量进行扩充，其**逻辑容量**由外存容量和内存容量之和决定，其**最大容量**由计算机的地址结构决定，其运行速度接近于内存，成本接近于外存。



## 二、虚拟存储器的实现方法

---

### ◆ 实现虚拟存储器必须解决好以下有关问题：

- ◆ 主存辅存统一管理问题
- ◆ 逻辑地址到物理地址的转换问题
- ◆ 部分装入和部分对换问题

### ◆ 虚拟存储管理主要采用以下技术实现：

- ◆ 请求分页存储管理
- ◆ 请求分段存储管理
- ◆ 请求段页式存储管理





## 二、虚拟存储器的实现方法

### 1、请求分页系统

在分页系统的基础上，增加了请求调页功能、页面置换功能所形成的页式虚拟存储器系统。

它允许只装入若干页的用户程序和数据，便可启动运行，以后在硬件支持下通过调页功能和页面置换功能，陆续将要运行的页面调入内存，同时把暂不运行的页面换到外存上，置换时以页面为单位。

**系统须设置相应的硬件支持和软件：**

(1) 硬件支持：请求分页的页表机制、缺页中断机构和地址变换机构。

(2) 软件：请求调页功能和页面置换功能的软件。



## 二、虚拟存储器的实现方法

### 2、请求分段系统

在分段系统的基础上，增加了请求调段功能及分段置换功能，所形成的段式虚拟存储器系统。

它允许只装入若干段的用户程序和数据，便可启动运行，以后在硬件支持下通过请求调段功能和分段置换功能，陆续将要运行的段调入内存，同时把暂不运行的段换到外存上，置换时以段为单位。

**系统须设置相应的硬件支持和软件：**

(1) 硬件支持：请求分段的段表机制、缺段中断机构和地址变换机构

(2) 软件：请求调段功能和段置换功能的软件



## 三、虚拟存储器的特征

### 1、多次性

多次性是虚拟存储器最重要的特征。指一个作业被分成多次调入内存运行。

### 2、对换性

对换性指允许在作业运行过程中进行换进、换出。换进、换出可提高内存利用率。

### 3、虚拟性

虚拟性是指能够从逻辑上扩充内存容量，使用户所看到的内存容量远大于实际内存容量。虚拟性是虚拟存储器所表现出来的最重要的特征，也是实现虚拟存储器最重要的目标。

**注：**虚拟性以多次性和对换性为基础，而多次性和对换性又是以**离散**分配为基础。

## 5.2 请求分页存储管理方式

### ❖ 虚拟存储器的实现方式

	请求分页系统	请求分段系统
基本单位	页	段
长度	固定	可变
分配方式	固定分配	动态
复杂性	简单	较复杂

❖ **原理**——地址空间的划分与页式存储管理相同；装入页时，装入作业的一部分（即运行所需的）页即可运行。

- 请求分页中的硬件支持
- 请求分页中的内存分配策略和分配算法
- 请求分页中的页面调入策略



# 一、请求分页中的硬件支持

## 1、页表机制

页号	块号	状态位	访问字段	修改位	外存地址
----	----	-----	------	-----	------

- (1) 状态位P：指示该页是否已调入内存。
- (2) 访问字段A：记录本页在一段时间内被访问的次数或最近未被访问的时间。
- (3) 修改位M：表示该页在调入内存后是否被修改过。若修改过，则换出时需重写至外存。
- (4) 外存地址：指出该页在外存上的地址。



# 一、请求分页中的硬件支持

## 2、缺页中断机构

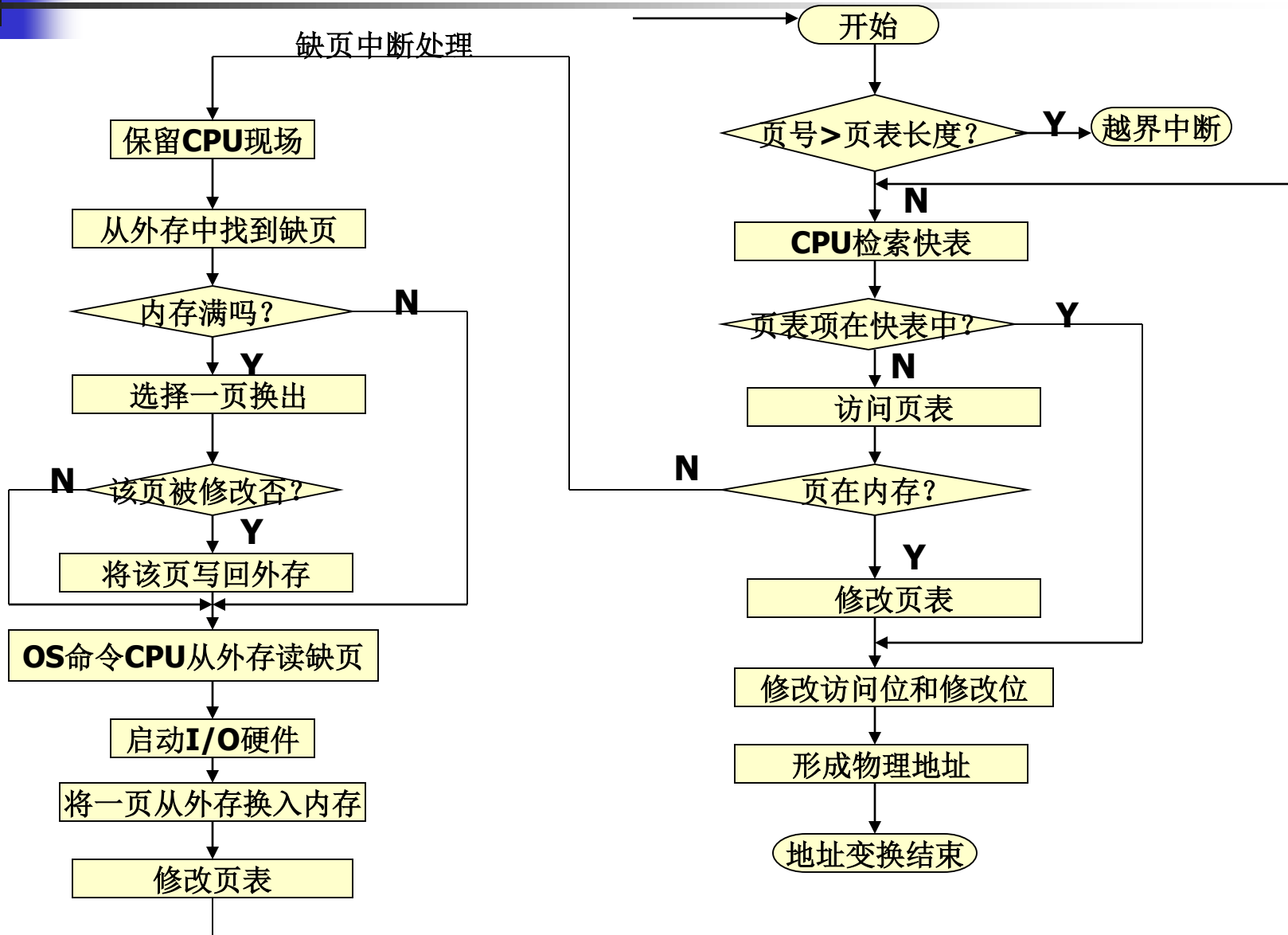
在请求分页系统中，当访问的页不在内存，便产生一缺页中断，请求OS将所缺页调入内存空闲块，若无空闲块，则需置换某一页，同时修改相应页表表目。

### 缺页中断与一般中断的区别：

- (1) 在指令执行期间产生和处理中断信号
- (2) 一条指令在执行期间，可能产生多次缺页中断

# 3、地址变换机构

程序请求访问一页



## 二、请求分页中的内存分配策略和分配算法

在请求分页系统中，为进程分配内存时，将涉及以下三个问题：

### 1、最小物理块数的确定

最小物理块数指能保证进程正常运行所需的最小的物理块数，与计算机的硬件结构有关，取决于指令的格式、功能和寻址方式。

### 2、物理块的分配策略

### 3、物理块分配算法





## 2、物理块的分配策略

(1) 固定分配局部置换：为每个进程分配**固定数目 $n$ 的物理块**，在整个运行中都不改变。如出现缺页，则从中置换一页。

(2) 可变分配全局置换：分配**固定数目的物理块**，但OS自留一空闲块队列，若发现缺页，则从空闲块队列中分配一空闲块与该进程，并调入缺页于其中。当空闲块队列用完时，OS才从内存中选择一页置换。

(3) 可变分配局部置换：分配一定数目的物理块，若发现缺页，则从**该进程**的页面中置换一页，根据该进程缺页率高低，则**可增加或减少物理块**。



### 3、物理块分配算法

---

在采用固定分配策略时，将系统中可供分配的所有物理块分配给各个进程，可采用以下几种算法：

- (1) 平均分配算法：平均分配给各个进程。
- (2) 按比例分配算法：根据进程的大小按比例分配给各个进程。
- (3) 考虑优先权的分配算法：将系统提供的物理块一部分根据进程大小先按比例分配给各个进程，另一部分再根据各进程的优先权适当增加物理块数。



### 三、请求分页中的页面调入策略

调入策略决定什么时候将一个页面由外存调入内存，从何处将页面调入内存。

#### □何时调入页面（1）

❖预调页策略：将那些**预计**在不久便被访问的页预先调入内存。这种调入策略提高了调页的效率，减少了I/O次数。但由于这是一种基于局部性原理的预测，若预调入的页面在以后很少被访问，则造成浪费，故这种方式常用于程序的首次调入。



## 三、请求分页中的页面调入策略

### □何时调入页面（2）

❖**请求调页策略**：当进程运行中访问的页不在内存时，则发出缺页中断，提出请求调页，由OS将所需页调入内存。这种策略实现简单，应用于目前的虚拟存储器中，但易产生较多的缺页中断，且每次调一页，系统开销较大，容易产生抖动现象。



### 三、请求分页中的页面调入策略

#### □ 从何处调入页面（1）

在请求分页系统中，通常将外存分成了文件区和对换区，文件区按离散分配方式存放文件，对换区按连续分配方式存放对换页。

❖ **对换区**：系统有足够的对换区空间，运行前可将与进程相关的文件从文件区复制至对换区，以后缺页时，**全部从对换区调页**。



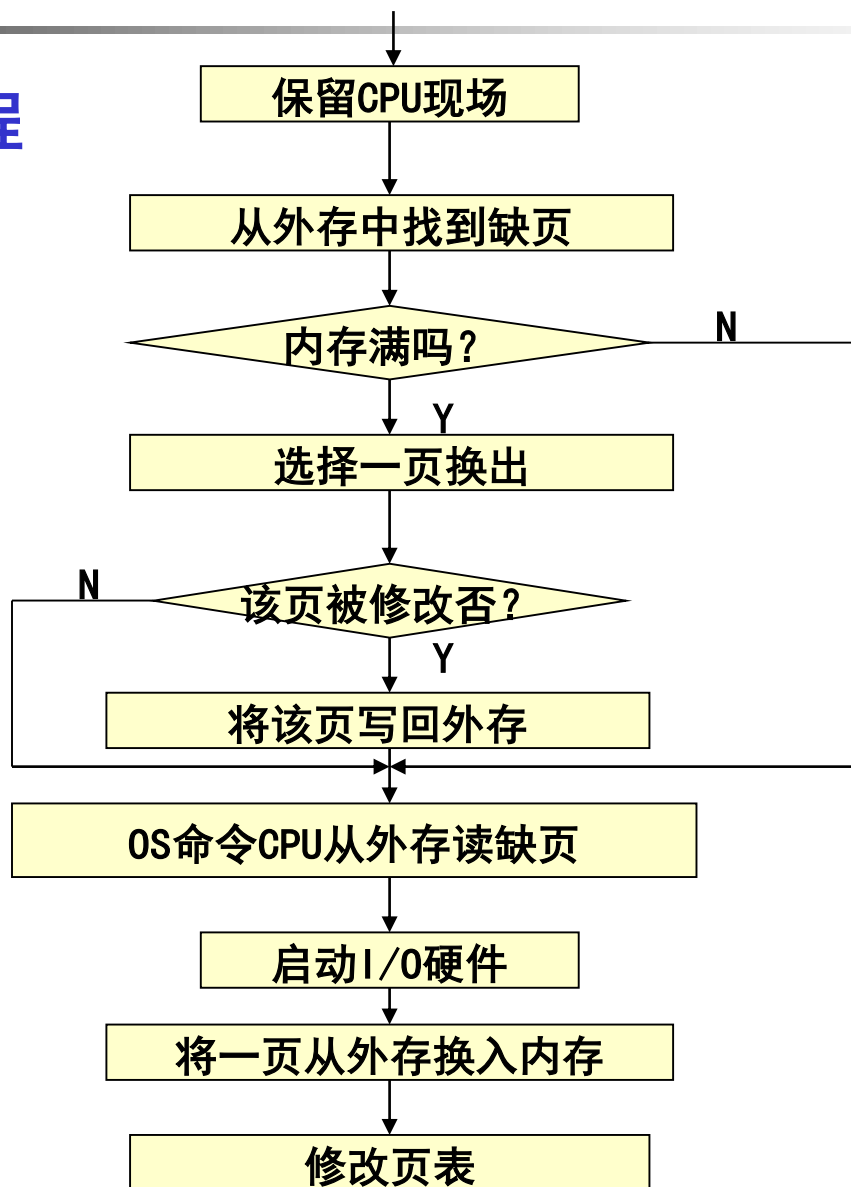
## 三、请求分页中的页面调入策略

### □ 从何处调入页面（2）

- ❖ **文件区**：系统没有足够的对换区空间，凡是不会被修改的文件，每次都直接从文件区调页，换出时不必换出。
- ❖ **文件区、对换区**：系统没有足够的对换区空间，对可能会修改的文件第一次调页直接从文件区，换出时换至对换区，以后从对换区调页。
- ❖ **UNIX方式**：凡未运行过的页面均从文件区调页，运行过的页面和换出的页面均从对换区调页。

### 三、请求分页中的页面调入策略

#### □ 页面调入过程





## 5.3 请求分页中的页面置换算法

页面置换算法也称为页面淘汰算法，是用来选择换出页面的算法。页面置换算法的优劣直接影响到系统的效率，若选择不合适，可能会出现以下现象：

刚被淘汰出内存的页面，过后不久又要访问它，需要再次将其调入，而该页调入内存后不久又再次被淘汰出内存，然后又要访问它，如此反复，使得系统把大部分时间用在了页面的调进换出上，而几乎不能完成任何有效的工作，这种现象称为**抖动**（又称**颠簸**）。





## 5.3 请求分页中的页面置换算法

常用的页面置换算法：

- ❖ 最佳置换算法：选择永远不再需要的页面或最长时间以后才需要访问的页面予以淘汰。
- ❖ 先进先出置换算法FIFO：选择先进入内存的页面予以淘汰。
- ❖ 最近最久未使用置换算法LRU：选择最近一段时间最长时间没有被访问过的页面予以淘汰。
- ❖ Clock置换算法
- ❖ \*其它算法

# 最佳置换算法例

假定系统为某进程分配了3个物理块，进程运行时的页面走向为 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5，开始时3个物理块均为空，计算采用**最佳置换**页面淘汰算法时的缺页率？

页面走向	1	2	3	4	1	2	5	1	2	3	4	5
物理块1	1	1	1	1			1			3	3	
物理块2		2	2	2			2			2	4	
物理块3			3	4			5			5	5	
缺页	缺	缺	缺	缺			缺			缺	缺	

缺页率=7/12

**注：**实际上这种算法无法实现，因页面访问的未来顺序很难精确预测，但可用该算法评价其它算法的优劣。



# 先进先出置换算法例题

1、假定系统为某进程分配了3个物理块，进程运行时的页面走向为 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5，开始时3个物理块均为空，计算采用先进先出页面淘汰算法时的缺页率？

页面走向	1	2	3	4	1	2	5	1	2	3	4	5
物理块1	1	1	1	4	4	4	5			5	5	
物理块2		2	2	2	1	1	1			3	3	
物理块3			3	3	3	2	2			2	4	
缺页	缺	缺	缺	缺	缺	缺	缺			缺	缺	

缺页率=9/12

## 先进先出置换算法例题

2、假定系统为某进程分配了4个物理块，进程运行时的页面走向为 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5，开始时4个物理块均为空，计算采用先进先出页面淘汰算法时的缺页率？

页面走向	1	2	3	4	1	2	5	1	2	3	4	5
物理块1	1	1	1	1			5	5	5	5	4	4
物理块2		2	2	2			2	1	1	1	1	5
物理块3			3	3			3	3	2	2	2	2
物理块4				4			4	4	4	3	3	3
缺页	缺	缺	缺	缺			缺	缺	缺	缺	缺	缺

缺页率=10/12

## 先进先出置换算法例题

3、假定系统为某进程分配了5个物理块，进程运行时的页面走向为 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5，开始时5个物理块均为空，计算采用先进先出页面淘汰算法时的缺页率？

页面走向	1	2	3	4	1	2	5	1	2	3	4	5
物理块1	1	1	1	1			1					
物理块2		2	2	2			2					
物理块3			3	3			3					
物理块4				4			4					
物理块5							5					
缺页	缺	缺	缺	缺			缺					

缺页率=5/12

# 先进先出置换算法\_注（1）：

- 1、该算法的出发点是最早调入内存的页面不再被访问的可能性会大一些。
- 2、该算法实现比较简单，对具有线性顺序访问的程序比较合适，而对其他情况效率不高。因为经常被访问的页面，往往在内存中停留最久，结果这些常用的页面却因变老而被淘汰。

页面走向	1	2	3	4	1	2	5	1	2	3	4	5
物理块1	1	1	1	4	4	4	5			5	5	
物理块2		2	2	2	1	1	1			3	3	
物理块3			3	3	3	2	2			2	4	
缺页	缺	缺	缺	缺	缺	缺	缺			缺	缺	



## 先进先出置换算法\_注（2）：

**3、**先进先出算法存在一种异常现象，即在某些情况下会出现分配给的进程物理块数增多，缺页次数有时增加，有时减少的奇怪现象，这种现象称为**Belady现象**。如上几例：

物理块数	3	4	5
缺页次数	9	10	5



## 最近最久未使用算法例

假定系统为某进程分配了3个物理块，进程运行时的页面走向为 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5，开始时3个物理块均为空，计算采用最近最久未使用页面淘汰算法时的缺页率？

页面走向	1	2	3	4	1	2	5	1	2	3	4	5
物理块1	1	1	1	4	4	4	5			3	3	3
物理块2		2	2	2	1	1	1			1	4	4
物理块3			3	3	3	2	2			2	2	5
缺页	缺	缺	缺	缺	缺	缺	缺			缺	缺	缺

缺页率=10/12





## 最近最久未使用算法\_注 (1)

❖ **该算法的出发点**：如果某个页面被访问了，则它可能马上还要被访问。反之，如果很长时间未被访问，则它在最近一段时间也不会被访问。

❖ **该算法的性能接近于最佳算法，但实现起来较困难**。因为要找出最近最久未使用的页面，必须为每一页设置相关记录项，用于记录页面的访问情况，并且每访问一次页面都须更新该信息。这将使系统的开销加大，所以在实际系统中**往往使用该算法的近似算法**。



## 5.4 请求分段式存储管理方式

- ❖ 请求分段存储管理系统也与请求分页存储管理系统一样，为用户提供了一个比内存空间大得多的**虚拟存储器**。虚拟存储器的最大容量由计算机的地址结构确定。
- ❖ 在请求分段存储管理系统中，作业运行之前，只要求将当前需要的若干个分段装入内存，便可启动作业运行。在作业运行过程中，如果要访问的分段不在内存中，则通过**调段功能**将其调入，同时还可以通过**置换功能**将暂时不用的分段换出到外存，以便腾出内存空间。



## 5.4 请求分段式存储管理方式

---

### ❖ 请求分段中的硬件支持

- 段表机制
- 缺段中断机构
- 地址变换机构

### ❖ 分段共享与保护

- 共享段表
- 共享段的分配与回收
- 分段保护（越界检查、存取控制检查、环境保护机构）



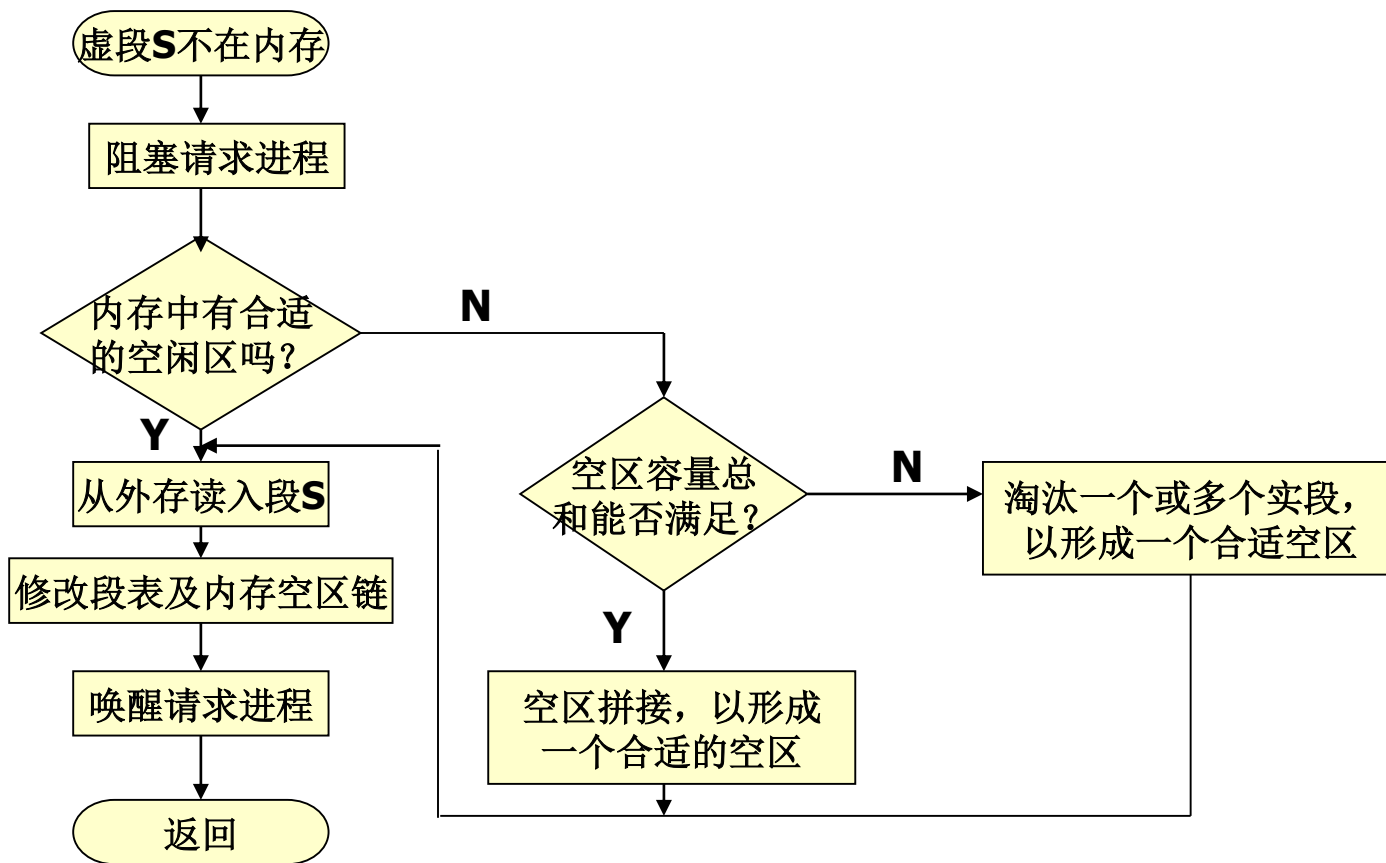
# 段表机制

段名	段长	段的基址	存取方式	访问字段 <b>A</b>	修改位 <b>M</b>	存在位 <b>P</b>	增补位	外存地址
----	----	------	------	------------------	-----------------	-----------------	-----	------

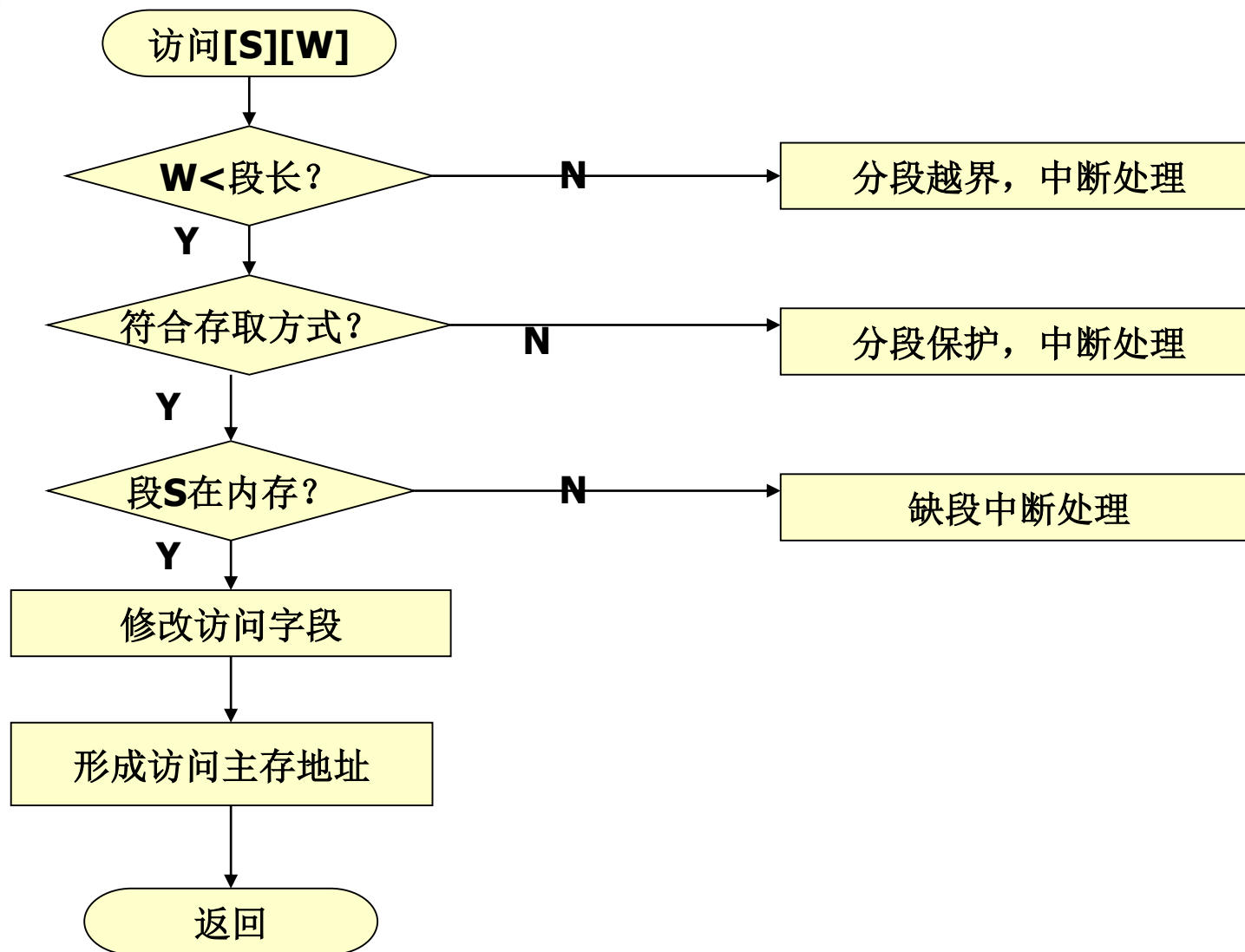
- ❖ 存取方式：存取属性（执行、只读、允许读/写）
- ❖ 访问字段A：记录该段被访问的频繁程度
- ❖ 修改位M：表示该段在进入内存后，是否被修改过。
- ❖ 存在位P：表示该段是否在内存中。
- ❖ 增补位：表示在运行过程中，该段是否做过动态增长。
- ❖ 外存地址：表示该段在外存中的起始地址。

# 缺段中断机构

❖ 当被访问的段不在内存中时，将产生一缺段中断信号。其缺段中断的处理过程如图：



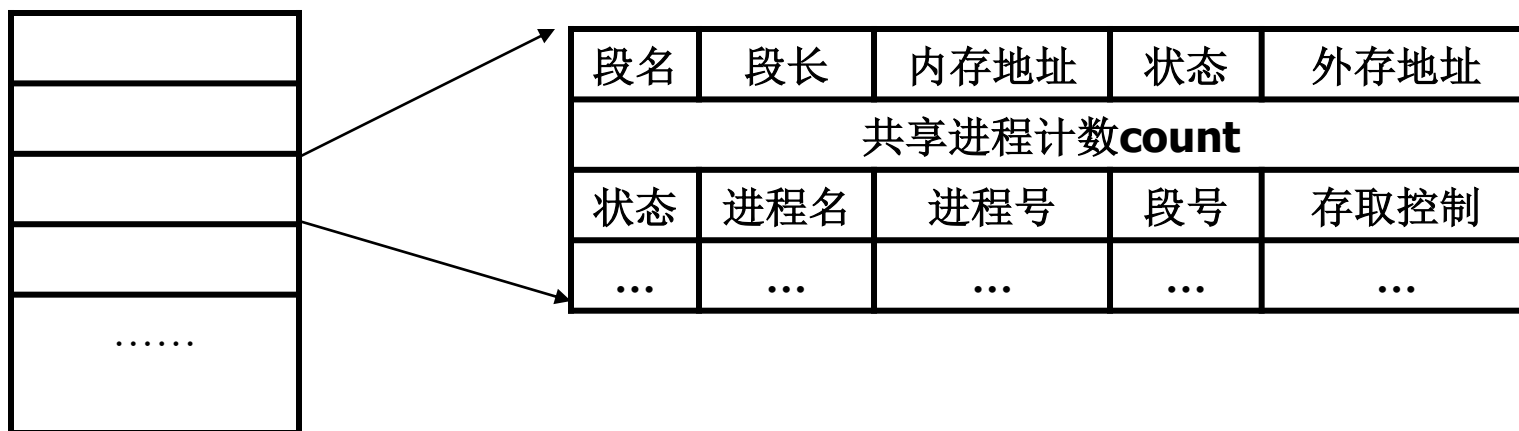
# 地址变换机构



# 分段的共享与保护

为实现分段的共享，应配置相应的数据结构：

## ❖ 共享段表

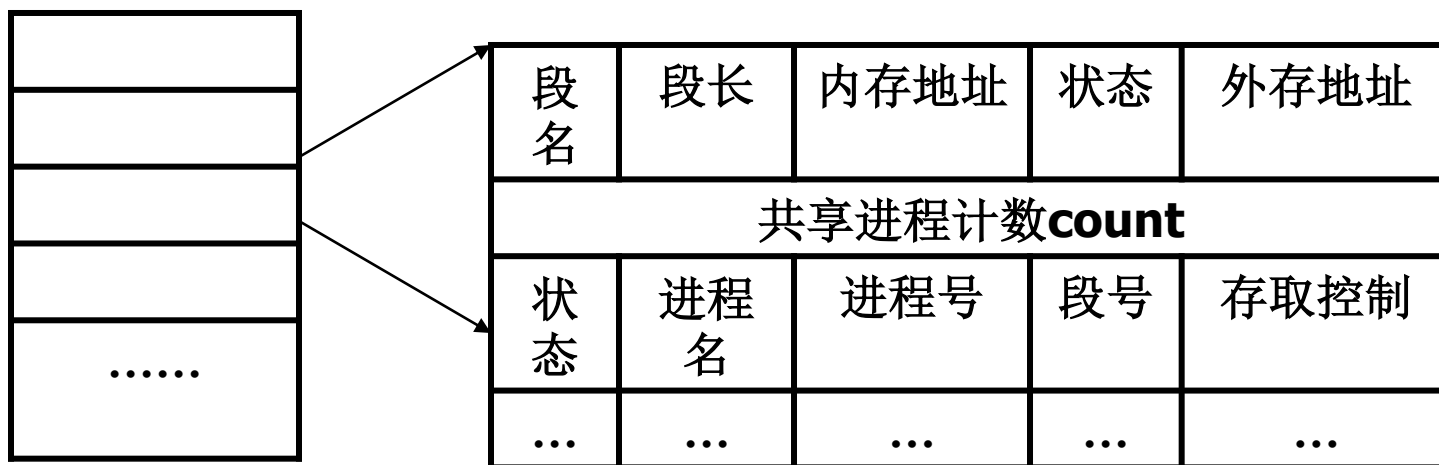


共享段表

# 共享段的分配与回收 (1)

## ❖ 共享段的分配

当第一个使用共享段的进程提出请求时，由系统为该共享段分配一物理区，并调入该共享段，同时修改相应的段表（该段的内存地址）和共享段表。当其它进程需要调用此段时，不需调入，只需修改相应的段表和共享段表。



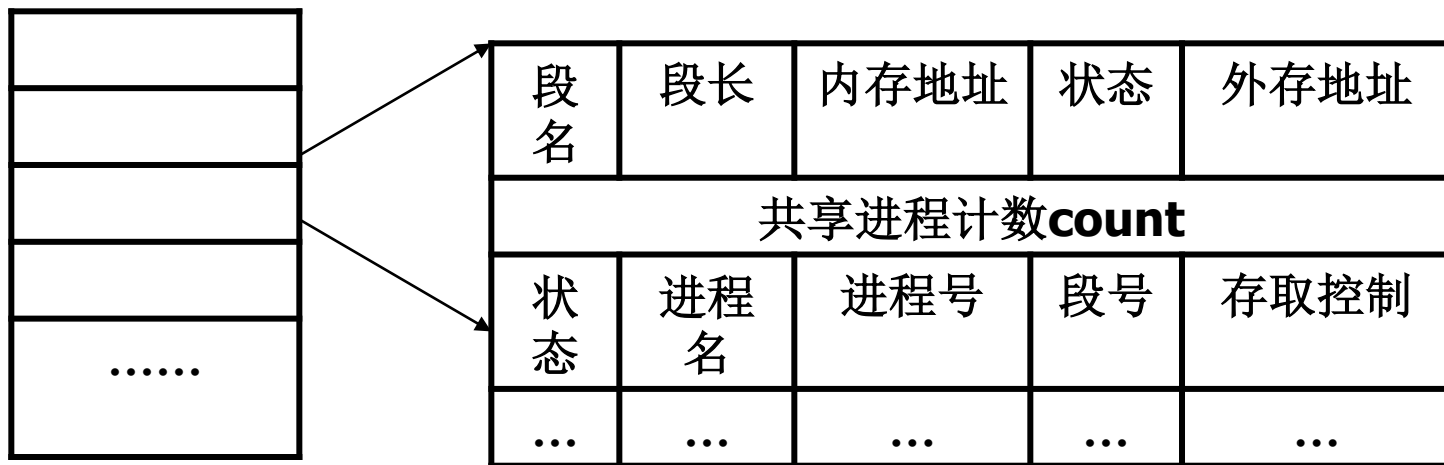
共享段表



## 共享段的分配与回收（2）

### ❖ 共享段的回收

当共享共享段的某进程不再使用该共享段时，修改相应的段表和共享段表。当最后一个共享此段的进程也不再需要此段时，则系统回收此共享段的物理区，同时修改共享段表（删除该表项）。



共享段表



# 分段保护（1）

在分段系统中，由于每个段在逻辑上是独立，因而比较容易实现信息保护。目前分段管理的保护主要有三种：

## ❖ 地址越界保护

先利用段表寄存器中的段表长度与逻辑地址中的段号比较，若段号超界则产生越界中断；再利用段表项中的段长与逻辑地址中的段内位移进行比较，若段内位移大于段长，也会产生越界中断。注：在允许段动态增长的系统中，允许段内位移大于段长。



## 分段保护（2）

---

### ❖ 访问控制保护（存取控制保护）

在段表中设置了一个存取控制字段，用于规定对该段的访问方式。

### ❖ 环保护机构

在环系统中，程序的访问和调用应遵循一定的规则：

（1）一个程序可以访问同环或较低特权环中的数据；

（2）一个程序可以调用同环或较高特权环中的服务；



# 本章练习

---

- 1、什么是虚拟存储器？
- 2、假定系统为某进程分配了3个物理块，进程运行时的页面走向为7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0, 1, 7, 0, 1, 开始时3个物理块均为空，给出下列置换算法时页面置换情况，并计算出该算法的缺页率？
  - (1) 最佳置换淘汰算法
  - (2) 先进先出淘汰算法
  - (3) 最近最久未使用淘汰算法