



# 操作系统

## 第十七讲

张涛

# Review

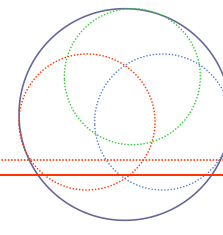
## 分段存储管理

### 分段存储管理基本思想

### 段地址映射

### 分段的共享与保护

### 分段管理方案评价



# Today we focus on...

段页式存储管理

段页式存储管理基本思想

地址映射

虚拟存储管理其它问题

Windows NT中的存储管理

# 4.8.1 段页式存储管理基本思想

## combined paging and segmentation

### ■ 用户程序划分

按段式划分（对用户来讲，按段的逻辑关系进行划分；对系统讲，按页划分每一段）

### ■ 内存划分

按页式存储管理方案

### ■ 内存分配

以页为单位进行分配

### ■ 逻辑地址



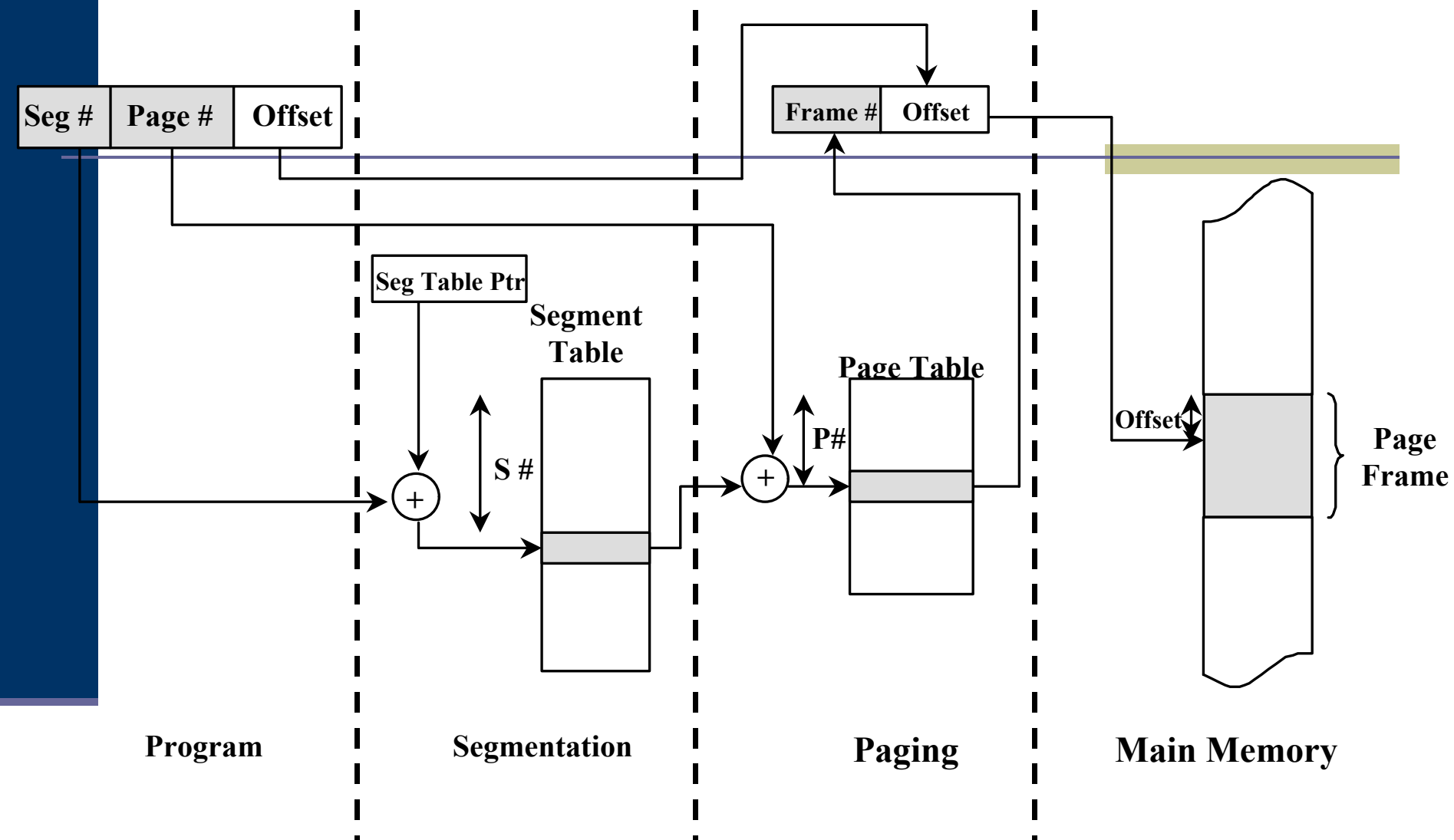
## 4.8.2 地址映射

- **段表**：记录了每一段的页表始址和页表长度
- **页表**：记录了逻辑页号与内存块号的对应关系（每一段有一个，一个程序可能有多个页表）
- **内存分配管理**：同页式管理

- 存储管理的分配单位：段，页
- 逻辑地址的组成：段号，页号，页内偏移地址。
- 地址变换：先查段表，再查该段的页表。
- 缺段中断和缺页中断。

### Virtual Address

Segment Number	Page Number	Offset
----------------	-------------	--------



## 段页式地址变换

## 4.9 虚拟存储管理的其它问题

- 调入策略
- 分配策略
- 清除策略
- 常驻集和工作集策略
- 缺页率算法
- 负载控制策略



## 4.9.1 虚拟存储的调入策略 ( fetch policy )

- 请求调页(demand paging): 只调入发生缺页时所需的页面。
  - 优点: 容易实现。
  - 缺点: 对外存I/O次数多, 开销较大
- 预调页(prepaging): 在发生缺页需要调入某页时, 一次调入该页以及相邻的几个页。
  - 优点: 提高调页的I/O效率。
  - 缺点: 基于预测, 若调入的页在以后很少被访问, 则效率低。常用于程序装入时的调页。

## 4.9.2 虚拟存储的分配策略 (assignment policy)

---

- 在虚拟段式管理中，如何对物理内存进行分配，可采用最佳适应、最先适应等。
- 在虚拟页式和段页式管理中，地址变换最后通过页表进行，因此不必考虑分配策略。

## 4.9.3 清除策略(cleaning policy)

- **请求清除(demand cleaning)**：该页被置换时才调出，把清除推迟到最后一刻。
  - 缺点：调入所缺页面之前还要调出已修改页面，缺页进程的等待时间较长
- **预清除(precleaning)**：该页被置换之前就调出，因而可以成批调出多个页面。
  - 缺点：可能形成不必要的开销。

## 4.9.4 常驻集和工作集策略

### ■ 常驻集(resident set)

- 常驻集指虚拟页式管理中给进程分配的物理页面数目。

### ■ 常驻集与缺页率的关系：

- 每个进程的常驻集越小，则同时驻留内存的进程就越多，可以提高并行度和处理器利用率；另一方面，进程的缺页率上升，使调页的开销增大。
- 进程的常驻集达到某个数目之后，再给它分配更多页面，缺页率不再明显下降。该数目是"缺页率－常驻集大小"曲线上的拐点(curve)。

## ■ 常驻集大小的确定方式：

- **固定分配(fixed-allocation)**：常驻集大小固定。各进程平均分配，根据程序大小按比例分配，优先权
- **可变分配(variable-allocation)**：常驻集大小可变，按照缺页率动态调整（高或低 → 增大或减小常驻集），性能较好。增加算法运行的开销。

## ■ 置换范围(replacement scope)：被置换的页面局限在本进程，或允许在其他进程。

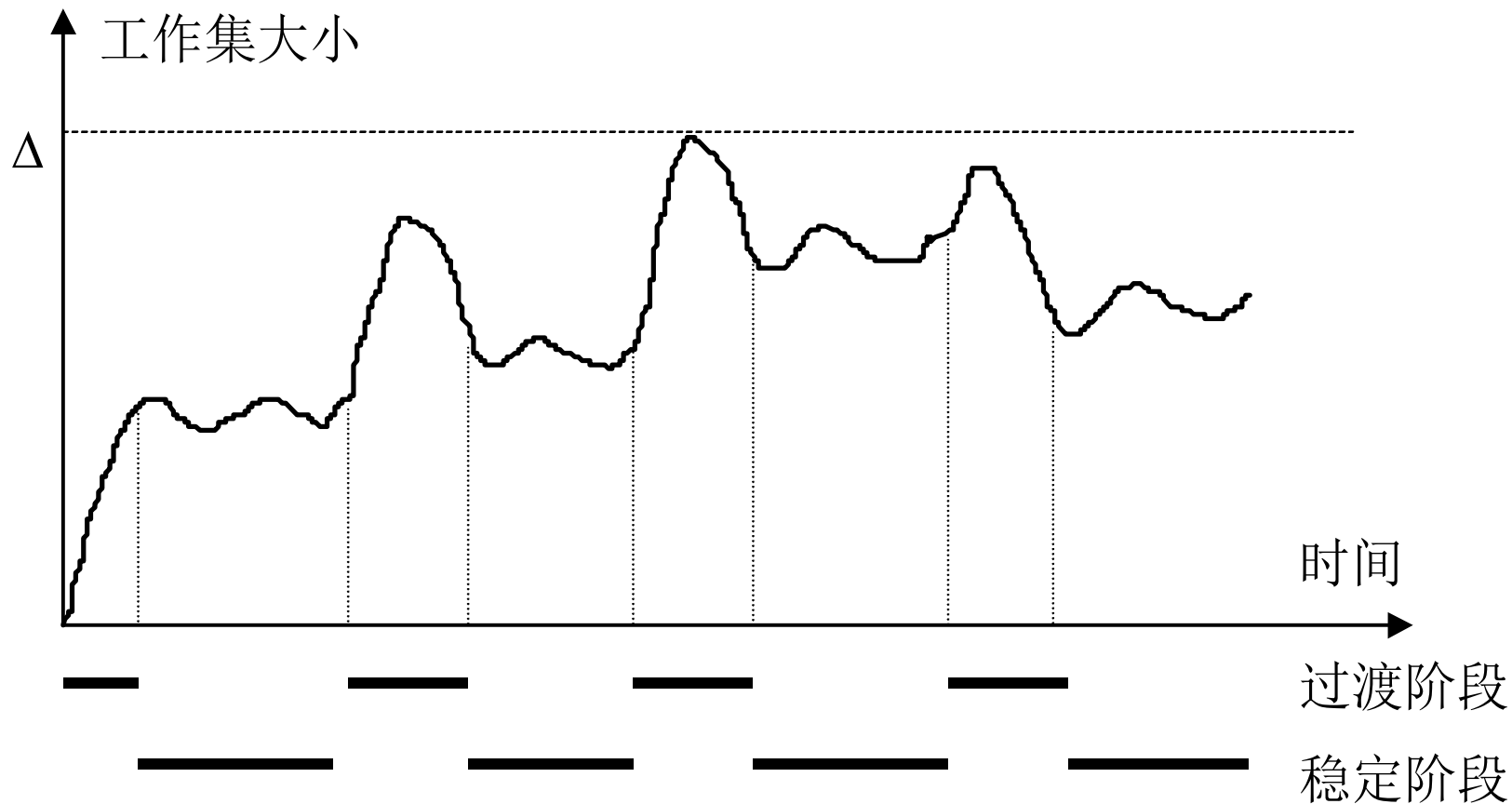
- **局部置换(local replacement)**：容易进行性能分析
- **全局置换(global replacement)**：更为简单，容易实现，运行开销小

## ■ 常驻集大小和置换范围的配合：三种策略

- **固定分配+局部置换**：进程开始前要依据进程类型决定分配多少页面。多了会影响并发水平，少了会使缺页率过高。
- **可变分配+局部置换**：局部转换但在大尺度上进行可变分配。
- **可变分配+全局置换**：这时OS会一直维持一定数目的空闲页面，以快速置换；主要问题是置换策略的选择，如何决定哪个进程的页面将被调出。较好的选择是页面缓冲算法。
- **不包括“固定分配全局置换”**（因为对各进程进行固定分配时不可能进行全局置换）。

# 工作集策略(working set strategy)

- 常驻集大小的动态调整策略----引入工作集概念。
- 工作集是一个进程执行过程中所访问页面的集合，可用一个二元函数 $W(t, \Delta)$ 表示
  - $t$ 是执行时刻；
  - $\Delta$ 是一个虚拟时间段，称为窗口大小(window size)，它采用"虚拟时间"单位(即阻塞时不计时)，大致可以用执行的指令数目，或处理器执行时间来计算；
  - 工作集是在 $[t - \Delta, t]$ 时间段内所访问的页面的集合；
  - $|W(t, \Delta)|$  指工作集大小即页面数目；



**工作集大小的变化：**进程开始执行后，随着访问新页面逐步建立较稳定的工作集。当内存访问的局部性区域的位置大致稳定时，工作集大小也大致稳定；局部性区域的位置改变时，工作集快速扩张和收缩过渡到下一个稳定值。



## ■ 工作集的性质：

- 随 $\Delta$ 单调递增： $W(t, \Delta) \subseteq W(t, \Delta + a)$ ，其中 $a > 0$ ；

## ■ 利用工作集进行常驻集调整的策略：

- 记录一个进程的工作集变化；
- 定期从常驻集中删除不在工作集中的页面；
- 总是让常驻集包含工作集；

## ■ 困难：

- 工作集的过去变化未必能够预示工作集的将来大小或组成页面的变化；
- 记录工作集变化要求开销太大；
- 对工作集窗口大小 $\Delta$ 的取值难以优化，而且通常该值是不断变化的；

## ■ 主要缺点：

- 当局部性区域的位置改变时，工作集的变化处于过渡阶段，其快速扩张使较多新页面添加到进程的常驻集中；
- 其中较少使用的页面至少还要经过一段虚拟时间才会被淘汰，因而带来较多不必要的调页开销。

## 4.9.5 缺页率算法 (PFF, page fault frequency)

- 页面被访问时的处理：每个页面设立使用位(use bit)，在该页被访问时设置use bit=1；
- 缺页时的处理：每次缺页时，由操作系统计算与上次缺页的"虚拟时间"间隔 $t$ ，如处理器执行时间
- 缺页时对常驻集的调整：定义一个"虚拟时间"间隔的阈值(threshold) $F$ 。依据 $t$ 和 $F$ 来修改常驻集。
  - 如果 $t$ 小于 $F$ ，则所缺页添加到常驻集中；
  - 否则，将所有use bit=0的页面从物理内存清除并缩小常驻集；随后，对常驻集中的所有页面设置use bit=0；
- 可变采样间隔算法：VSWS, variable-interval sampled working set

## 4.9.6 虚拟存储中的负载控制

### ■ 改善时间性能的途径

- **降低缺页率**：缺页率越低，虚存平均访问时间延长得越小；
- **提高外存的访问速度**：外存和内存的访问时间比值越大，则达到同样的平均访问时间，所要求的缺页率就越低；

### ■ 抖动问题(thrashing)

- 随着驻留内存的进程数目增加，或者说进程并发水平的上升，处理器利用率先是上升，然后下降。
- 处理器利用率下降的原因通常称为虚拟存储器发生“**抖动**”，

- **决定驻留内存的进程数目**，在避免出现抖动的前题下，尽可能提供进程并发水平。

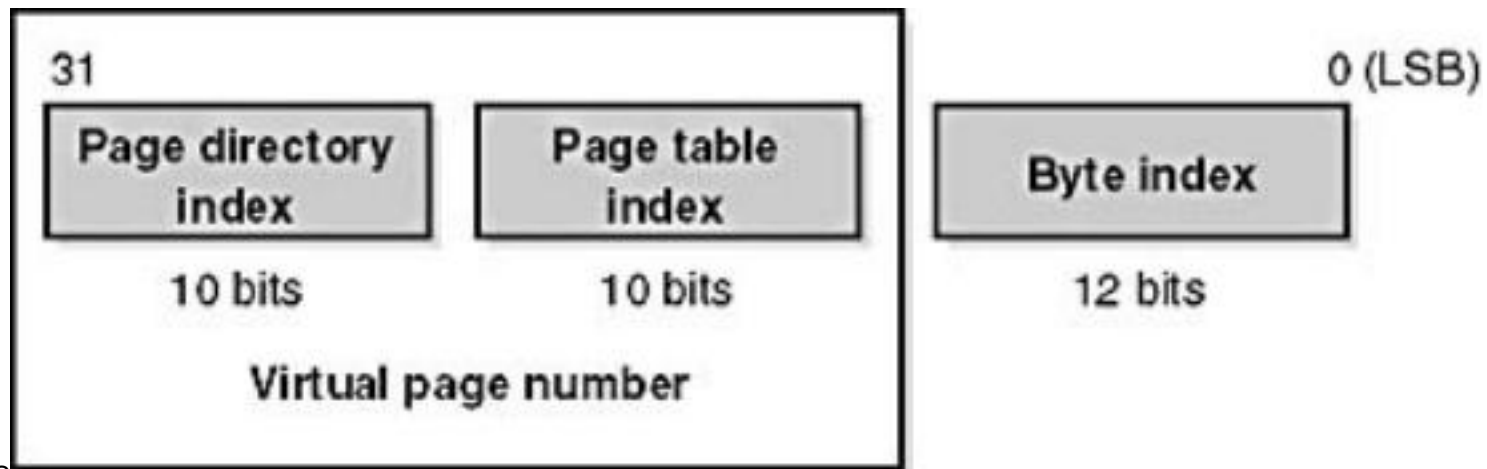
## 4.10 Windows NT中的存储管理

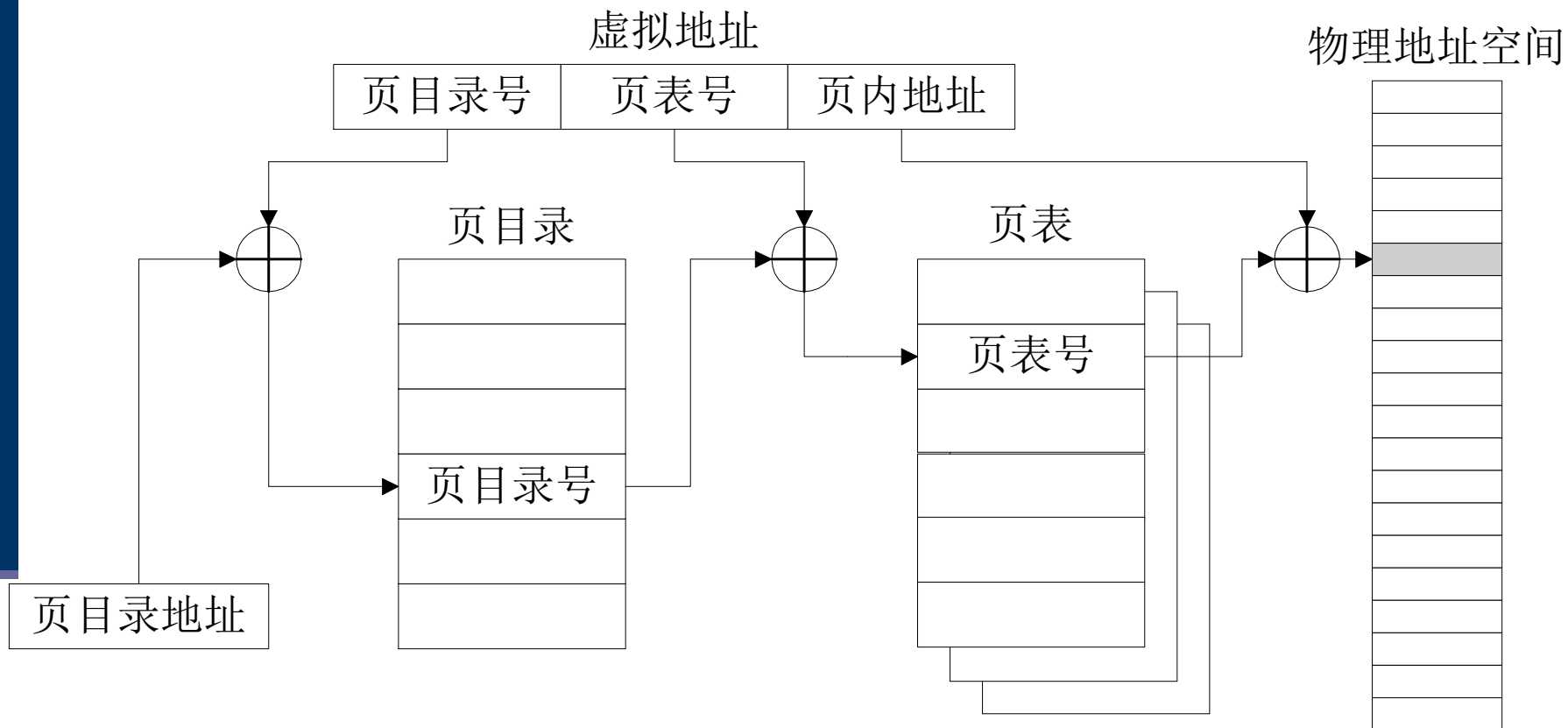
- 页面大小为4KB ( $2^{12}$ )。每个进程地址空间为4GB ( $2^{32}$ )
- **用户存储区**：在用户态和核心态都可访问的用户存储区为2GB；用户存储区为页交换区，可对换到外存；包括：
  - **专用进程地址空间**：用户代码、数据和堆栈
  - **线程环境块 (TEB)**：用户态代码可修改的线程控制信息
  - **进程环境块 (PEB)**：用户态代码可修改的进程控制信息
  - **共享用户数据页**：系统存储区映像，为用户态可访问的系统空间，目的在于避免用户态与核心态的频繁切换；如：系统时间。

- **系统存储区**：在核心态可访问的系统存储区为 2GB；按交换特征，系统存储区可分为：
  - **固定页面区**：永不被换出内存的页面；如：HAL 特定的数据结构；
  - **页交换区**：非常驻内存的系统代码和数据；如：进程页表和页目录；
  - **直接映射区**：常驻内存且寻址由硬件直接变换的页面，访问速度最快；用于存放内核中频繁使用且要求快速响应的代码。

# 地址转换机构

NT使用2级页表结构转换虚拟地址，第一级称为**页目录**（每个进程一个页目录），第二级称为**页表**。每个页目录或页表有1024( $2^{10}$ )个表项，每个表项为4字节。由于每个页面为4KB，每个进程的地址空间可为4GB ( $2^{10} * 2^{10} * 2^{12}$ )。







# 页面调度策略

- **取页策略：**NT采用按进程需要进行的请求取页和按集群方法进行的提前取页。
  - **集群方法：**在发生缺页时，不仅装入所需的页，而且装入该页附近的一些页。
- **置页策略：**在线性存储结构中，简单地把装入的页放在未分配的物理页面即可。
- **淘汰策略：**采用局部FIFO置换算法。在本进程范围内进行局部置换，利用FIFO算法把驻留时间最长的页面淘汰出去。

# 工作集策略

- NT根据内存负荷和进程缺页情况自动调整工作集。
  - 进程创建时，指定一个最小工作集（可以用SetProcessWorkingSetSize函数指定）；
  - 当内存负荷不太大时，允许进程拥有尽可能多的页面；
  - 系统通过自动调整保证内存中有一定的空闲页面存在；

# What you need to do?

---

- 复习课本4.6节的内容
- 课后作业：习题20

See you next time!