

# 第35 - 36讲 虚拟存储管理的 软件策略概述



# 虚拟存储管理技术中的软件策略

## ➤ OS 存储管理子系统的设计要素

- 是否使用虚拟存储管理技术
- 选用分页技术、分段技术还是段页技术
- 采用什么样的存储管理算法

其中，

- 前两个问题属于 CPU 硬件设计范畴；从 OS 设计者的角度看，它们相当于“选用什么样的硬件平台作为 OS 的运行平台”。
- 最后一个问题属于 OS 软件设计范畴；该问题是后面讨论的重点内容



# 虚拟存储管理技术中的软件策略

- 除了一些老的个人计算机 OS（比如 MS-DOS）以及专用 OS 外，绝大多数 OS 均使用了虚拟存储管理技术。
- 单纯的分段技术现在已经很少使用；绝大多数 OS 选用了分页技术或者段页技术。
- 因此，我们主要讨论**虚拟存储分页技术中的软件策略**



# 虚拟存储管理技术中的软件策略

- Resident Set Management( 驻留集管理 )
- Placement Policy( 放置策略 )
- Fetch Policy( 获取策略 )
- Replacement Policy( 置换策略 )
- Cleaning Policy( 清除策略 )
- Load Control( 负载控制 )



# Resident Set Management

解决的问题：

系统应当为每个活动进程分配多少个 **frame** ？



# Resident Set

指进程驻留在内存中的页面所组成的集合。

## 注 释

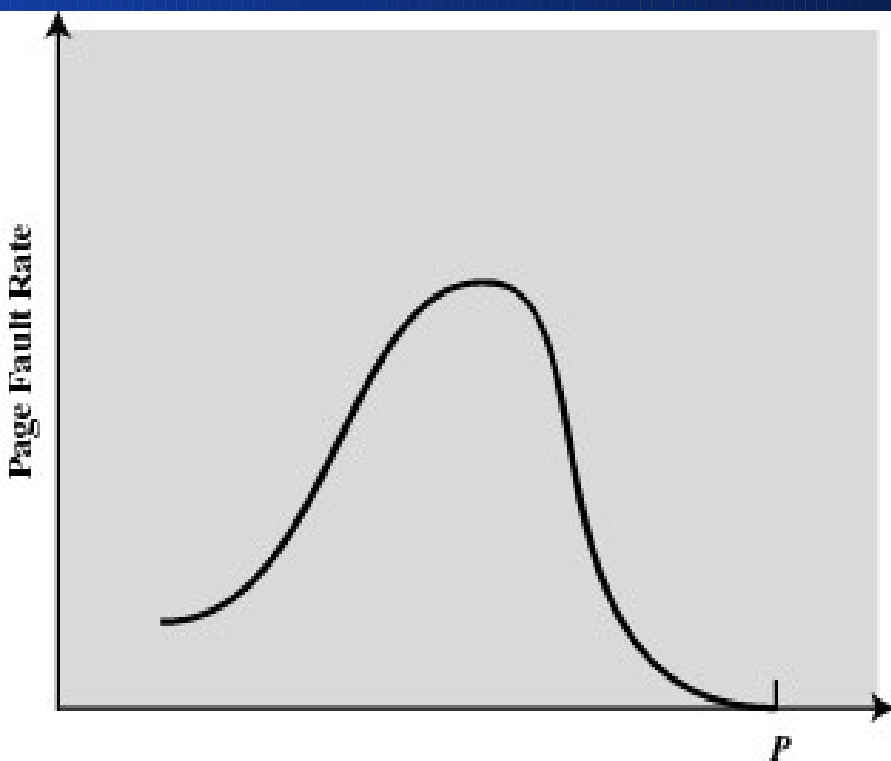
解决的基本问题即：每个活动进程驻留集的尺寸应当为多大？



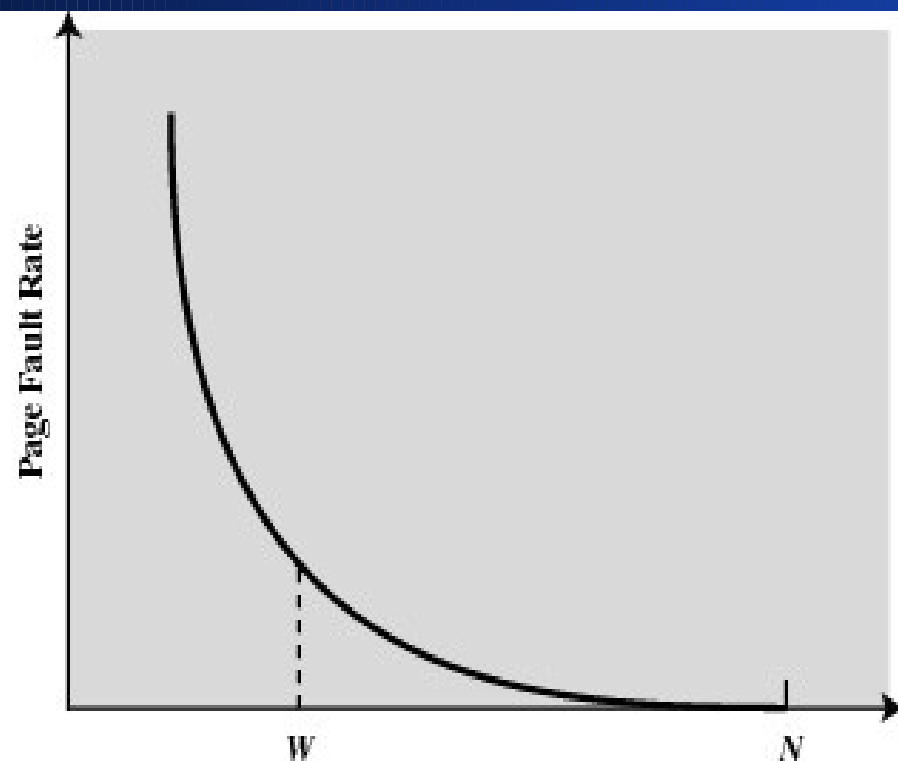
# 影响 Frame 分配的因素

- 分配给每个活动进程的 frame 数越少，在内存中驻留的活动进程数就越多，进程调度程序能调度就绪进程的概率就越大。
- 分配给每个活动进程的 frame 数越少，每个活动进程在其运行过程中发生缺页中断的概率就越大。
- 分配给每个活动进程的 frame 数超过一定限度时，多余的 frame 并不能显著地降低每个活动进程的缺页中断率





(a) Page Size



(b) Number of Page Frames Allocated

$P$  = size of entire process

$W$  = working set size

$N$  = total number of pages in process

Figure 8.11 Typical Paging Behavior of a Program





# Resident Set Size

## ➤ Fixed-allocation

- gives a process a fixed number of pages within which to execute
- when a page fault occurs, one of the pages of that process must be replaced

## ➤ Variable-allocation

- number of pages allocated to a process varies over the lifetime of the process



# 调整驻留集尺寸的方法（工作集方法：1）

- 一个进程的工作集  $w(t, \Delta)$  是指，在时间  $t$  之前的一段时间  $\Delta$  内该进程所引用的全部页面的集合。
- 工作集  $w(t, \Delta)$  是时间  $t$  的函数。一般说来， $t$  不同， $w(t, \Delta)$  的尺寸和 / 或元素也不同。
- 通常人们把  $\Delta$  称为工作集  $w(t, \Delta)$  的窗口尺寸。显然，工作集  $w(t, \Delta)$  也是窗口尺寸  $\Delta$  的函数。一般说来，窗口尺寸  $\Delta$  越大，工作集尺寸  $|w(t, \Delta)|$  也就越大。
- 要保证一个进程有效地运行，系统就必须保证 **任何时候** 该进程的工作集都驻留在内存中。



# 调整驻留集尺寸的方法（工作集方法：2）

- 从理论上讲，系统可以通过监控一个进程的工作集来调整其驻留集。
- 然而，实践中，监控各个进程的工作集将导致很大的系统开销。
- 而且，监控到的工作集信息是进程的历史；因此，据此调整各个进程的驻留集未必能够获得好的系统性能。
- 此外，窗口尺寸的最佳值也是无法确定的。

# 调整驻留集尺寸的方法 (PFF 算法: 1)

- 基于工作集方法的基本原理，人们在实践中提出了缺页频率算法 (Page-Fault Frequency Algorithm，简称 PFF)。
- 使用 PFF 算法的系统通过监控各个进程的页面访问失败率来调整各个进程的驻留集尺寸。



# 调整驻留集尺寸的方法 (PFF 算法： 2)

## ➤ PFF 算法将按照以下规则调整各进程的驻留集尺寸：

- 如果一个进程的缺页率**低于**某个最小阈值，那么系统将**减小**该进程的驻留集尺寸；
- 如果一个进程的缺页率**高于**某个最大阈值，那么系统将**增加**该进程的驻留集尺寸。

## ➤ 因为缺页率与缺页时间间隔成反比，因此，可以通过测量缺页时间间隔来监控缺页率。

# Placement Policy

解决的基本问题：

系统应当在内存的什么位置为活动进程分配

**frame ?**

# Placement Policy

- 绝大多数使用虚拟存储分页技术的系统可以在内存中的任何位置为活动进程分配 **frame** 。
- 但是，有些系统需要专门技术解决内存中页面的放置问题。



# Fetch Policy

**Determines when a page should be brought into memory ?**



# Fetch Policy

- **Demand paging (请求调页)** only brings pages into main memory when a reference is made to a location on the page.
  - Many page faults when process first started.
- **Prepaging (预调页)** brings in more pages than needed.
  - More efficient to bring in pages that reside contiguously on the disk.



# Replacement Policy （置换策略）

- Which page is replaced?
- 当一个页面欲装入内存时，系统应当在什么范围内判断已经没有任何空闲 **frame** 分配给新的页面？
- 当系统在指定的范围内发现已经没有任何空闲 **frame** 分配给新的页面时，系统应当从指定的范围内选择哪个页面移出内存？



# Variable Allocation, Global Scope( 可变分配, 全局置换 )

- Easiest to implement.
- Adopted by many operating systems.
- Operating system keeps list of free frames.
- Free frame is added to resident set of process when a page fault occurs.
- If no free frame, replaces one from another process.



# Variable Allocation, Global Scope( 可变分配, 全局置换 )

- When new process added, allocate number of page frames based on application type, program request, or other criteria.
- When page fault occurs, select page from among the resident set of the process that suffers the fault.
- Reevaluate allocation from time to time.



# 驻留集尺寸与置换范围

- 固定分配策略将导致系统必须使用局部置换策略；反之，全局置换策略将导致系统必须使用可变分配策略
- 可变分配策略与局部置换策略也可组合，即系统可根据性能的需要增加或减少分配给每个活动进程的 frame 数；当进程的 frame 全部用完，而需要装入一个新的页面时，系统将在该进程的当前驻留集中选择一个页面移出内存



# Replacement Policy （置换注意事项）

## ➤ Frame Locking

- If frame is locked, it may not be replaced.
- Kernel of the operating system.
- Control structures.
- I/O buffers.
- Associate a *lock bit* with each frame.



# Replacement Policy

- **Page removed should be the page least likely to be referenced in the near future.**
- **Most policies predict the future behavior on the basis of past behavior.**

