

# 内存管理

## 内存管理概念

① 内存管理是?

1

② 内存管理的目的是?

1

③ 内存管理的功能是?

10

④ 内存管理有哪些类别?

4

⑤ 程序的链接与接入过程

17

⑥ 逻辑地址与物理地址

6

⑦ 进程的内存映像

12

● 内存保护

由OS和硬件一起完成

7

● 内存共享

8

内存管理的目的  
1. 方便用户  
2. 提高内存利用率

## 内存管理方法

① 连续分配

28

② 分段

18

满足程序员/用户  
1. 方便编程  
2. 分段共享  
3. 分段保护  
4. 动态链接  
5. 动态增长

③ 分页

28

面向计算机

④ 段页

10

为了在物理内存有限的情况下扩充内存；  
选择在逻辑上扩充内存即虚拟内存管理

## 虚拟内存管理

① 传统存储管理方式的特征

5

② 局部性原理

5

③ 虚拟存储器

6

④ 虚拟内存的实现

10

⑤ 请求分页管理方式

16

⑥ 页框分配

29

⑦ 页面置换算法 (书P206)

10

选择调出页面的算法

① 抖动/颠簸

6

② 工作集

7

③ 内存映射文件

6

④ 虚拟存储器性能影响因素

10

⑤ 地址翻译 (书P211)

1

## 虚拟存储器 其他知识点

内存管理概念

1 内存管理是?

✓ 内存管理 = 操作系统对内存的划分和动态分配

2 内存管理的目的是?

为了更好地支持多道程序并发执行

3 内存管理的功能是?

- ✓ 1. 内存空间的分配与回收 由OS完成主存储器空间的分配和管理
- ✓ 2. 地址转换 存储管理将逻辑地址转换为物理地址
- ✓ 3. 内存空间的扩充 利用虚拟存储技术/自动覆盖技术, 从逻辑上扩充内存
- ✓ 4. 内存共享 允许多个进程访问内存的同一部分
- ✓ 5. 存储保护 保证各道作业在各自的存储空间运行, 互不干扰

4 内存管理有哪些类别?

- ✓ 连续分配 单一连续分配——【单道发展到多道OS】→ 固定分区分配——【为了适应大小不同的程序】→ 动态分区分配
- ✓ 不连续分配 分段存储管理 → 分页存储管理 → 段页

5 程序的链接与装入过程

- ✓ 1. 编译 编译 = 由编译程序将用户源代码编程成目标模块
- ✓ 2. 链接 链接 = 由链接程序将目标模块和库函数链接, 形成完整的装入模块
  - 类别 (书P166) 1. 静态链接 2. 装入时动态链接 3. 运行时动态链接
- 装入 = 由装入程序将装入模块装入内存运行
- ✓ 3. 装入
  - 类别 (书P167) 1. 静态装入 在编程时把物理地址计算好
  - 2. 可重定位装入 转入时把逻辑地址转换为物理地址, 但装入后不能改变
  - 3. 动态重定位装入 执行时再决定装入的地址并装入, 装入后有可能会换出

6 逻辑地址与物理地址

- ✓ 逻辑地址 1. 编译后, 目标模块从0号单元开始编址, 称为该目标模块的逻辑地址
- 地址重定位 2. 不同进程可以有相同的逻辑地址, 这些相同的逻辑地址可以映射到主存的不同位置
- ✓ 物理地址 3. 进程运行时, 看到和使用的地址都是逻辑地址
- 物理地址空间是指内存中物理单元的集合, 是地址转换的最终地址

7 进程的内存映像

- ✓ 当一个程序调入内存运行时, 就构成了进程的内存映像
- 组成要素
  - 代码段 代码段是只读的, 可以被多个进程共享
  - 数据段 程序运行时加工处理的对象, 包括全局变量和静态变量
  - 进程控制块PCB 存放在系统区, OS通过PCB控制和管理进程
  - 堆 用来存放动态分配的变量【动态的】
  - 栈 用来实现函数调用【动态的】

内存保护

由OS和硬件一起完成

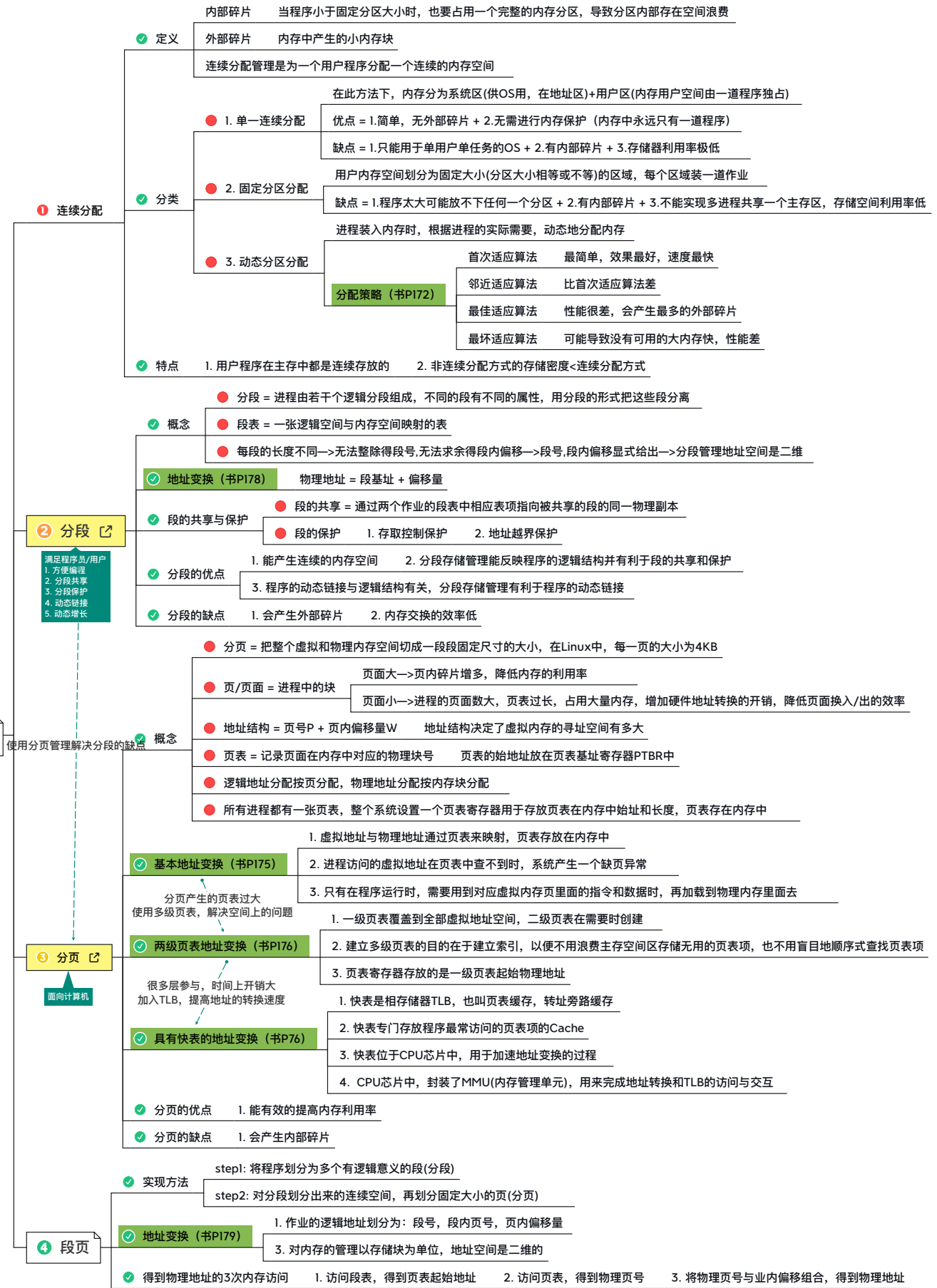
- ✓ 内存保护的目 = 确保每个进程都有一个单独的内存空间
- ✓ 内存保护的方法
  - 1. 在CPU中设置一对上, 下限寄存器, 判断CPU访问的地址是否越界
  - 只有OS内核才能加载这两个寄存器 重定位寄存器(基址寄存器)含最小的物理地址值【用于“加”】
  - 2. 使用重定位寄存器和界址寄存器 界址寄存器含逻辑地址的最大值【用于“比”】
  - 逻辑地址 + 重定位寄存器的值 = 物理地址

内存共享

- ✓ 概念
  - 1. 只有只读区域的进程内存空间可以共享
  - 2. 纯代码/可重入代码 = 不能修改的代码, 不属于临界资源
  - 3. 可重入程序通过减少交换数量来改善系统性能
- ✓ 实现方式 (书P169) 1. 段的共享 2. 基于共享内存的进程通信 3. 内存映射文件

内存管理方法

内存管理的目的  
1. 方便用户  
2. 提高内存利用率



虚拟内存管理



## 虚拟存储器 其他知识点

### ① 抖动/颠簸

✔ 定义 在页面置换时，出现频繁的页面调度行为

系统中同时运行的进程太多→分配给每个进程的物理块太少→进程在运行时频繁出现缺页→频繁的调页

✔ 产生原因 主要原因是因为页面置换算法不合理

对换区大小和进程优先级都与抖动无关

### ② 工作集

✔ 定义 在某段时间间隔内，进程要访问的页面集合

✔ 如何确定工作集？ 基于局部性原理，用最近访问过的页面来确定

✔ 有什么作用？

1. 工作集反映了进程在接下来一段时间内很可能频繁访问的页面集合

2. 为了防止抖动现象，要使分配给进程的物理块数>工作集大小

防止抖动方法

### ③ 内存映射文件

✔ 定义 与虚拟内存有些相似，将磁盘文件的全部或部分内容与进程虚拟地址空间的某区域建立映射关系

✔ 作用 可以之间访问被映射的文件，而不必执行文件I/O操作，也无序对文件内容进行缓存处理

✔ 优点 适合用来管理大尺寸文件

### ④ 虚拟存储器性能影响因素

1. 页面较大→缺页率较低→可以减少页表长度，但似得页内碎片增大

2. 页面较小→缺页率较高

→可以减少内存碎片，提高内存利用率

→使得页表过长，占用大量内存

3. 分配给进程的物理块数越多，缺页率就越低

4. 分配给进程的物理块数超过某个值时，对缺页率的改善并不明显

5. 好的页面置换算法可以使进程在运行过程中具有较低的缺页率

6. LRU，CLOCK将未来可能要用到的进程保存在内存中，可以提高页面的访问速度

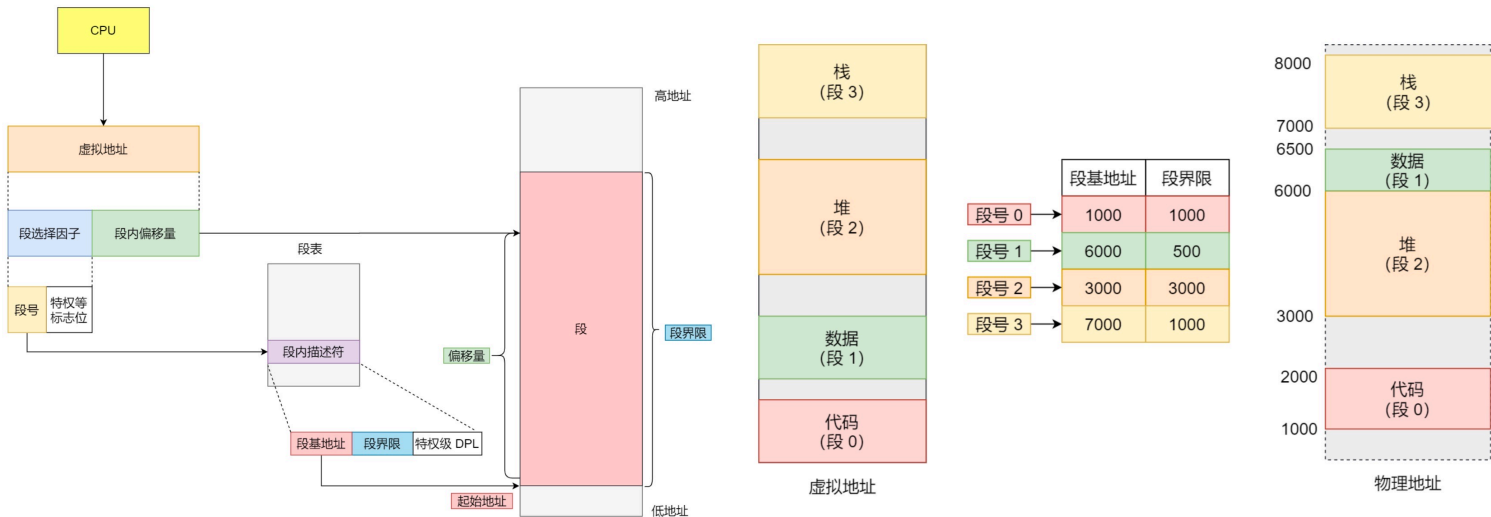
7. 编写程序的局部化程度越高，执行时的缺页率越低

8. 存储和访问尽量使用系统的访问方式（如都按行存储就按行访问）

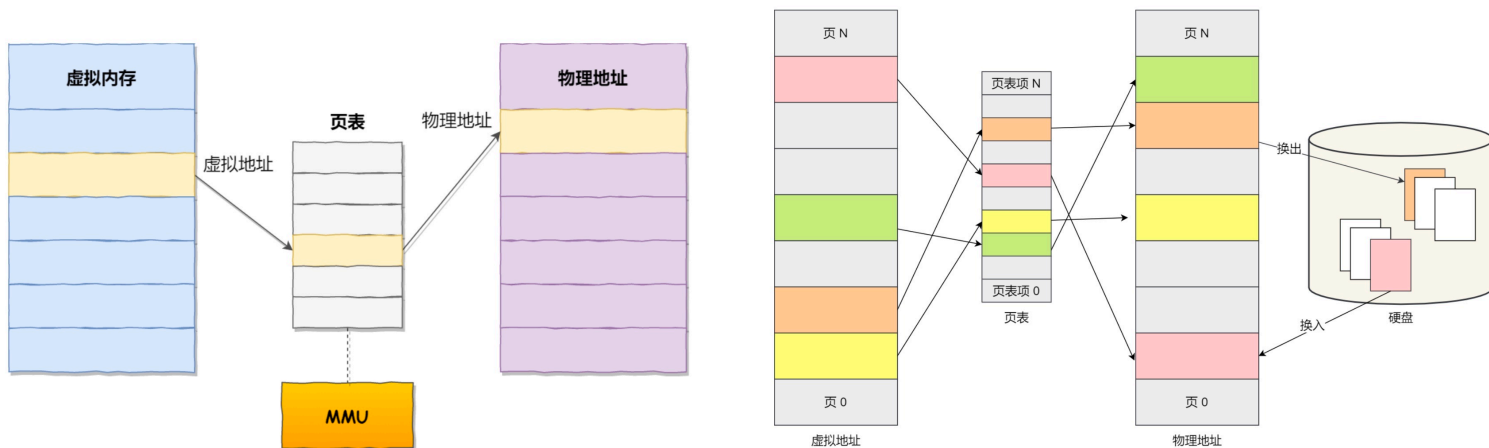
### ⑤ 地址翻译（书P211）

结合计组食用

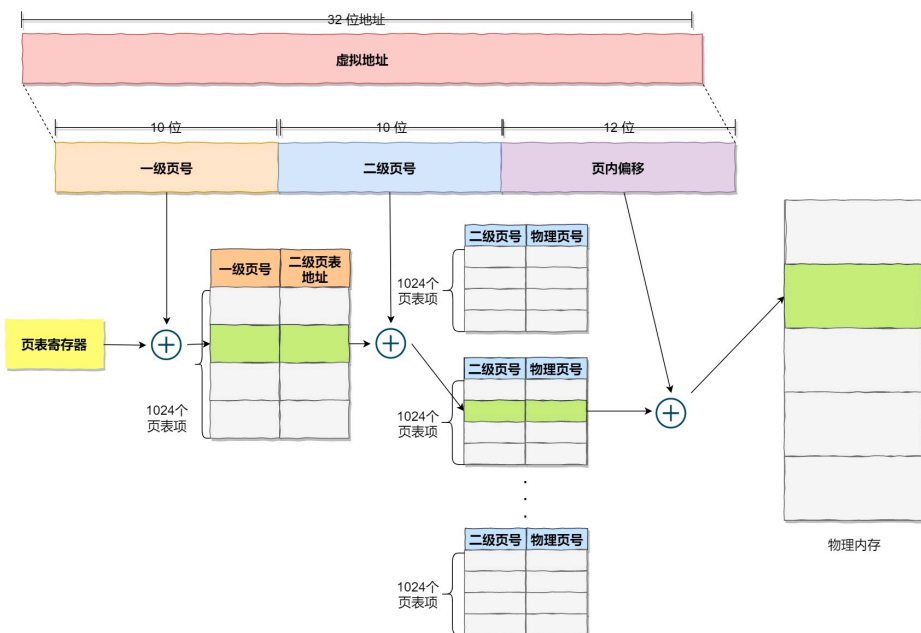
# 分段



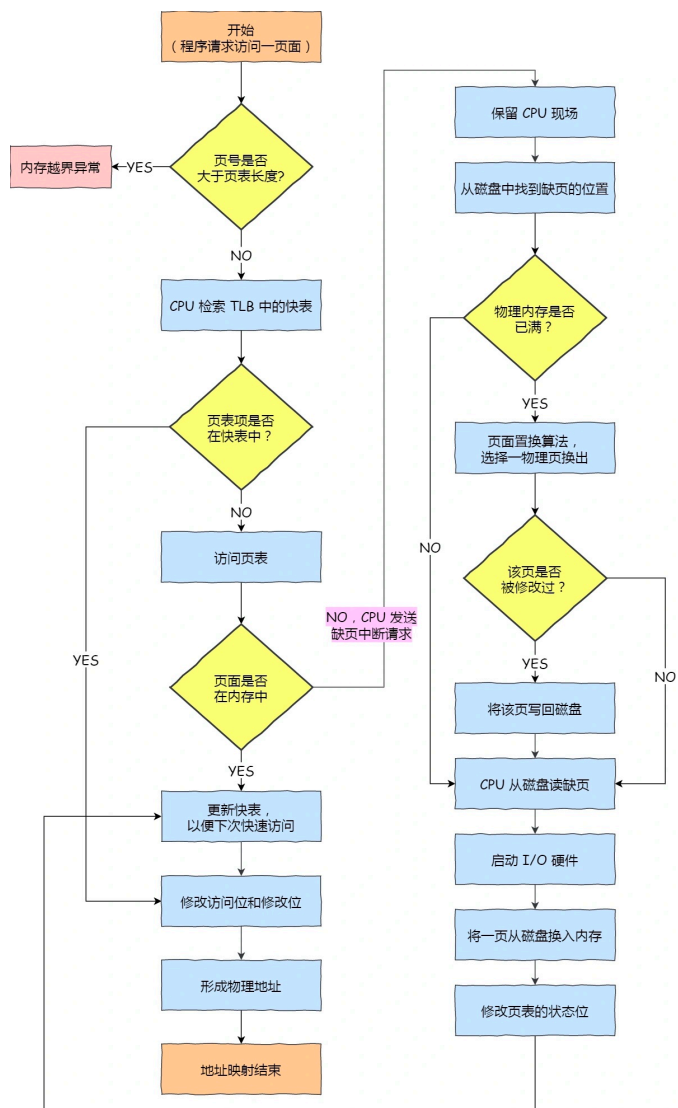
# 分页



# 二级分页



## 如何调入页面？



## 缺页中断的处理

