4内存管理

- 基于分页的虚拟存储系统
  - 请求调入功能和置换功能，能从逻辑上对内存容量进行扩充的一种存储器系统
  - 好处：提高内存的利用率
  - 离散性，多次性，对换性，虚拟性
- 2程序的链接和装入
  - 它为操作系统提供可装入的程序模块. 链接程序要解决的问题是将编译后的目标模块装配成一个可执行的程序.
  - 动态链接 和 静态链接
  - 绝对装入方式
  - 可重定位装入方式
    - 程序装入时，对目标程序中的指令和数据地址的修改过程成为可重定位
    - 物理地址 = 逻辑地址 + 程序在内存中的起始位置
  - 动态运行时装入(动态重定位)
    - 重新定位寄存器，当用获得CPU进程在内存中的起始地址更新重新定位寄存器
- 3连续分配存储管理方式
  - 单一连续分配内存
    - 适用单用户，单任务的系统
  - 固定分区分配方式
    - 将内存用户区划为若干固定大小的区域，每个区域中驻留一道程序
  - 动态分区分配方式
    - 系统动态的采用内存划分，根据进程需要的空间大小分配，内存中的分区的大小和数量是变化的。动态分区比静态分区提高内存利用率
  - 动态分区分配原理
    - 空闲分区表
    - 空闲分区链
      - 动态的为每个空闲分区建立一个节点，每个节点包括分区大小，分区起始位置，指向前一个空闲分区的指针，以及后一个空闲分区的节点指针
  - 动态分区分配算法
    - 首次适应算法
      - 按照开始的位置递增
      - 外部碎片 和 内部碎片
    - 循环适应算法
      - 优点： 空间区分布均匀, 查找开锁较小
      - 缺点：容易使分区缺少大空闲区
    - 最佳适应算法
      - 每次进程分配前对空闲区按空闲区的大小递增
      - 优点:避免了大材小用, 能提高内存利用率
      - 缺点：容易留下难以利用的小空闲区
  - 动态分区分配的流程
    - 仅回收区的前面有相邻的空闲分区
    - 仅回收区的后面有相邻的空闲分区
    - 回收区的前后都有相邻的空闲分区
- 1存储器的层次结构
  - 从高层到低屋(L0-L5), 数据越大, 层次越低, 存储设备速度越慢, 容量更大, 价格更便宜
  - CPU寄存器：保存最常用的数据 L0层
  - 芯片内的L1高速缓存（SRAM）L1
  - 芯片内的L2高速缓存 (SRAM) L2
  - 主存储器(DRAM) 也就是内存条 L3
  - 本地二级存储，本地硬盘， L4
  - 远程二级存储 FTP服务器等 L5
- 4 基本分页存储管理方式
  - 把进程离散的存储在内存中物理地址不连续的区域中，这种方式成为离散内存管理方式
  - 分页存储管理，分段存储管理，段页式存储管理
  - 页：将一个进程的逻辑地址空间分成若干个大小相等的片，称为页
  - 页框:将物理内存空间分成与页大小相同的若干个存储快，称为页框或者页帧
  - 分页储存：为进程分配内存的时候，以页框为单位将进程中的若干个页分别装入多个不相连的页框中
  - 页内碎片: 进程的最后一页装不满一个页框，而形成了不可利用的碎片，称为页内碎片，是一种部分碎片
  - 页表: 是系统为进程建立的数据结构, 页表的作用是实现从页号到页框号的映射
  - 若A为逻辑地址, L为页大小, P为页号, W为页内偏移量, 则有以下计算关系
    P = INT(A/L)
    W = MOD(A/L)
  - 分页地址变换
    - 页号对应的页表项起始地址=页表起始地址+页表项长度 x 页号
    - 物理地址=页框大小 x 页框号+页内偏移量
  - 块表
    - 转换后援缓冲(TLB)；是为了提高CPU访问速度而采用的专用缓存，用来存放最近被访问过的页表项
  - 两级页表
    - 两级页表是将页表再进行分页，使每个页表分页的大小与页框一样
  - 反页设置
    - 反置页表
    - 反置页的地址映射