- 虚拟内存
  - 概念
    - 程序部分进入内存，暂时用不到的去外存
    - 如果要用到的部分不在内存，去外存调入
    - 如果内存不够，把暂时用不到的放到外存
    - 使用MMU来实现虚拟地址到物理地址的转变
    - 易混淆
      - 虚存最大容量是CPU寻址范围绝定
        - 32位则最大4GB(2^32)
      - 虚存实际容量 = min(最大容量，内存外存之和)
  - 特征
    - 多次性
      - 程序可以分多次调入内存
    - 对换性
      - 程序运行期间可以将程序换入换出
    - 虚拟性
      - 逻辑扩大
    - 基于离散分配
  - 实现
    - 请求分页存储管理
      - 跟基本分页差不多
      - 页表
        - 内存块号
        - 状态位/有效位
          - 是否进入内存
        - 访问字段
          - 上次访问时间/被访问几次
        - 修改位
          - 进入内存后是否被修改
            - 被修改则需要回外存
        - 外存地址
          - 在外存的位置
      - 缺页中断
        - 将缺页进程阻塞，调页完成再唤醒
        - 属于内中断的故障(fault)
        - 一条指令可产生多次缺页中断
        - 页面换入换出进行缓慢IO操作，应减少，追求更少缺页率，否则开销大
      - 快表(TLB)
        - 若其中页被调出外存，则需删除
        - 页表调入内存须同时修改快表
    - 请求分段存储管理
    - 请求段页式存储管理
  - 页面分配策略
    - 驻留集
      - 给进程分配物理页的集合(帧/分区)
        - 驻留集太小导致缺页率大
        - 太大资源利用率低，多道程序并发度下降
      - 固定分配
        - OS给每个进程分配固定大小的物理页
          - 只可进行局部置换
      - 可变分配
        - 驻留集大小可变
          - 局部/全局置换都可
    - 页面分配、置换策略
      - 局部置换
        - 缺页时只选择进程对应的物理页置换
        - 固定分配
          - 缺点:难确定开始时为进程分配多少物理页
        - 可变分配
          - 开始先分配物理页，如果缺页缺多了，就多分点物理页
          - 系统根据缺页频率动态分配页
          - 可能是最佳组合
      - 全局置换
        - 可以直接分配空闲物理页给缺页进程
        - 可以将其他进程的物理页调到外存，再给缺页进程
        - 可变分配
          - 开始为进程分配物理页并保持空闲物理页，进程缺页时从空闲区拿一个，如果空闲区空，选择未锁定的页面换到外存，并将其物理页分给缺页的。
            - 锁定：有重要数据不能换出内存的页
          - 缺点:调出外存的页面可能导致缺页
          - 最易实现
    - 读取策略/调入页面时机
      - 预调入策略
        - 根据局部性原理，预测进程可能被使用的页并调入
        - 预测成功率50%左右
          - 运行前常用
      - 请求调页策略
        - 运行期间缺页调入缺页部分
        - 因为每次调入一个，且都要IO，开销大
    - 放置策略
      - 分页/段页式
      - 纯分段
    - 从何处调页
      - 对换区足够大
        - 页调入调出都在对换区
        - 进程运行前先将数据从文件区复制到对换区
      - 对换区不够
        - 不会被修改的数据直接从文件区调入
        - 会修改则在对换区
      - UNIX
        - 运行前全都在文件区，被用过则在对换区
    - 抖动(颠簸/振荡)现象
      - 刚换出的页马上换入内存
      - 刚换入的页马上换出外存
      - 处理器大部分时间用于交换数据块
        - 降低CPU利用率
      - 原因
        - 分配给进程的物理页不够(或进程频繁访问的页数高于可用物理页数)
    - 工作集
      - 一定时间内，进程实际访问页面集合
      - 工具集一般小于窗口尺寸
        - 窗口尺寸:用于计算衡量工作集大小
        - 驻留集不能小于工作集，否则频繁缺页
      - 局部性的近似
  - 页表置换算法
    - 页面置换实现物理内存和虚拟内存间的分离
    - 最佳置换(OPT)
      - 每次被淘汰的页是长时间不再使用或永不使用的
      - 理想化，实际无法实现
    - 先进先出置换(FIFO)
      - 每次淘汰最早进入内存的页面
      - 按照内存块长度形成列表排序
      - Belady异常:当为进程分配的物理块越多，缺页次数不减反增
      - 实现简单，但算法性能差
    - 最近最久未使用置换(LRU)
      - 淘汰最近最久没被访问的
      - 页表会记录时钟
      - 性能好，但实现困难，开销大
    - 时钟置换(CLOCK)/最近未用(NRU)
      - 页表设置访问位，访问到设1，对页循环检查找到访问位为0的置换，如果为1设0不换出，如果全为1则将第一个从1变0的换出
      - 最多2次扫描
        - 均衡性能和开销
    - 改进型的时钟置换
      - 只有被淘汰页面修改，需要写回外存
      - 新增修改位
      - 首先找到访问位修改位都是0的修改，否则第二次扫描，将扫描到的访问位1置0，0置1，直到找到访问0修改1，否则第三次扫描找两个都为0的，若没找到，第四次扫描找访问0修改1
        - 只有第2次扫描会改访问位
      - 最多4次扫描
  - 传统存储管理缺点
    - 一次性
      - 作业需一次性全部装入内存
      - 作业很大不能全入内存，大作业不能运行
      - 大量作业要求运行，内存无法全部装进去，多道程序并发度下降
    - 驻留性
      - 程序被装入内存会一直驻留，直到程序结束
      - 哪怕该程序只要访问小部分，浪费资源，当时内存利用率不高
    - 时间局部性
      - 可用快表(cache)，但是造价高
    - 空间局部性
  - 外存(磁盘)
    - 对换区
      - 读写速度快
      - 连续分配方式
    - 文件区
      - 读写速度慢
      - 离散分配方式