虚拟内存

1. 背景
   1. 程序运行
      1. 基本要求：进程必须全部放入内存后方可运行
      2. 问题：进程所需内存大于实际内存的容量或内存中同时运行多个进程，可能导致进程无法运行
      3. 试解决：覆盖和动态加载，但是增加了程序员的负担
      4. 解决：①从物理上扩充内存容量 ②从逻辑上扩充内存容量
   2. 常规存储器

特征：①一次性（作业在运行前需要一次性的全部装入内存）②驻留性（作业装入内存后，一直驻留在内存中，直到作业结束）

问题：由于这两个特性，程序中暂时不用的数据占用了大量的内存空间，可能导致需要运行的作业无法装入内存

* 1. 程序的局部性
     1. 含义：在一段时间内，程序的执行仅局限于某个部分；相应的，它所访问的存储空间也局限于某个区域内
     2. 分类：

1. 时间局部性：
   1. 含义：如果程序中的某条指令一旦执行（/存储单元被访问），则不久的将来该指令可能再次被执行（/存储单元再次被访问）
   2. 典型原因：程序中存在着大量的循环
2. 空间局部性
   1. 含义：一旦程序访问了某个存储单元，则在不久的将来，其附近的存储单元也最有可能被访问。
   2. 典型原因：程序是顺序执行的
3. 含义：一种允许进程部分装入内存就可以执行的技术
4. 原理：局部性原理
5. 特点：
   1. 离散性：离散地分配内存
   2. 多次性：一个作业被分成多次调入内存运行，只有运行的部分程序需要在内存中
   3. 对换性：作业运行过程中，页面能够在内存和外存之间被换入和换出
   4. 虚拟性：从逻辑上扩充内存容量，使用户所看到的内存容量远大于实际内存容量
6. 考虑问题：
   1. 程序部分运行可以吗？

=>取页：将所需部分装入内存

* 1. 发现程序不在内存时，如何将其装入后继续运行？

=>请求调页：缺页时，产生缺页中断，将外存上的页调入内存

* 1. 内存无空间时怎么办？

=>页面置换：内存不足时，将某些页换出内存

1. 请求调页：
   1. 预调页策略：预测将要访问的程序或数据所在的页面，预先调入内存，主要用于进程的首次调入
   2. 定义：当进程运行过程中发生缺页时，由系统将缺的页调入内存，目前虚拟存储器大多采用此策略
   3. 硬件支持：在采用分页式内存分配方法的系统中，比起没有采用虚拟内存，支持虚拟内存的系统的页表项除了包括“页号”和“物理帧号”，还应有“状态位”“访问字段”“修改位”“外存地址”。
      1. 状态位：指示该页是否在内存中
      2. 访问字段：记录本页被访问的情况，以供页面置换算法参考
      3. 修改位：表示该页调入内存后是否被修改，用来确定该页被换出时是否需要重写外存上的此页
      4. 外存地址：指出该页在外存上的地址，通常是物理块号，供调入该页时使用
   4. 缺页中断：
      1. 定义：请求分页系统中，当所要访问的页面不在内存时，便要产生缺页中断，请求操作系统将所缺页调入内存
      2. 与一般中断的区别：
         * 1. 在指令执行期间产生和处理中断信号（一般中断在一条指令执行完后检查和处理中断信号）