1. 背景
   1. 基本要求：进程必须全部放入内存后方可运行
   2. 如果 进程大于内存的容量/内存中同时运行多个进程，可能导致进程无法运行
      * 覆盖和动态加载（增加了程序员的负担）
   3. 解决方法：
2. 从物理上扩充内存容量
3. 从逻辑上扩充内存容量
4. 常规存储器
   1. 特征：

一次性：作业在运行前需要一次性的全部装入内存

驻留性：作业装入内存后，一直驻留在内存中，直到作业结束

* + - 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. 但是，

程序通常有处理异常错误的代码，很少执行

数组、链表和表通常分配了比实际需要更多的内存

程序的某些选项或特点可能很少使用

即使需要完整的程序，也并不是同时需要所有的程序

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：

时间局部性

1. 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

空间局部性

* 1. 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

1. 虚拟内存
   1. 含义：
   2. 特点：
2. 离散性：离散地分配内存
3. 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
4. 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
5. 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
   1. 请求调页