内存映射
& 动态内存分配概念
VM System & Malloc Concepts

课 程 名 ： 计算机系统
主 讲 人 ： 孟文龙

本课内容

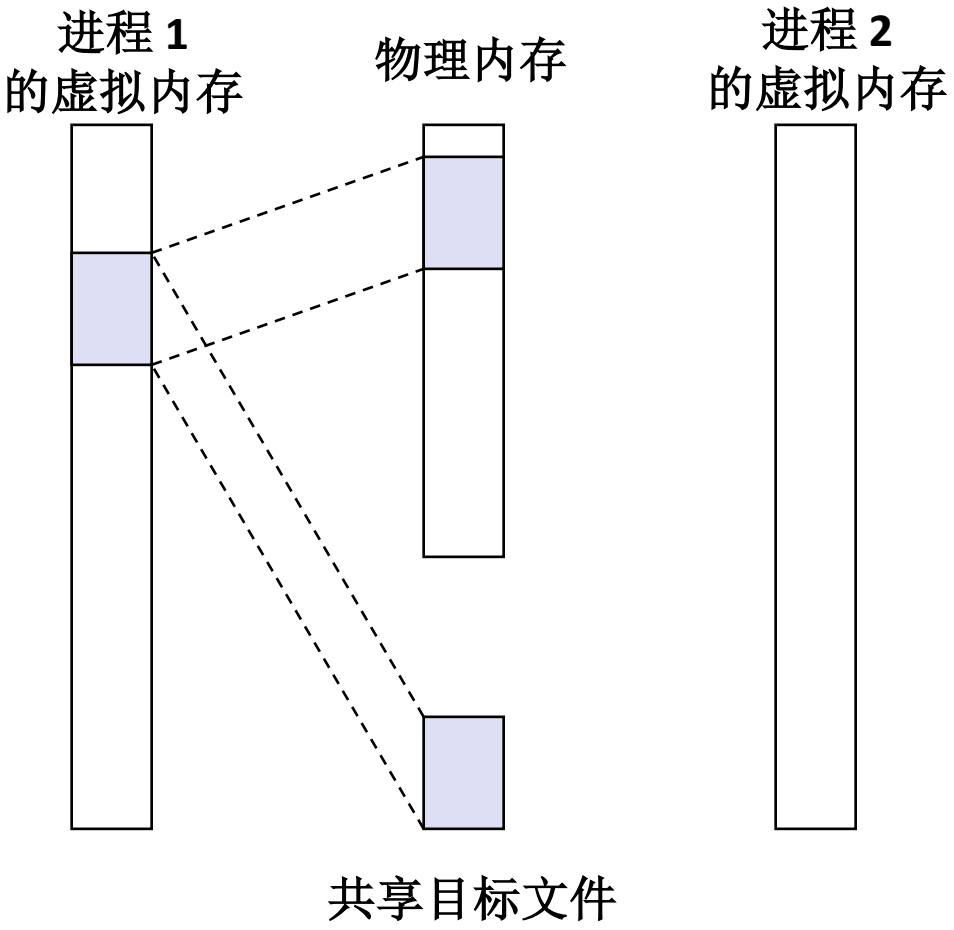
- **内存映射**
- 动态内存分配
  - 基本概念
  - 隐式空闲链表

内存映射

- Linux 通过将虚拟内存区域与磁盘上的对象相关联，以初始化这个虚拟内存区域的内容
  - 这个过程称为内存映射

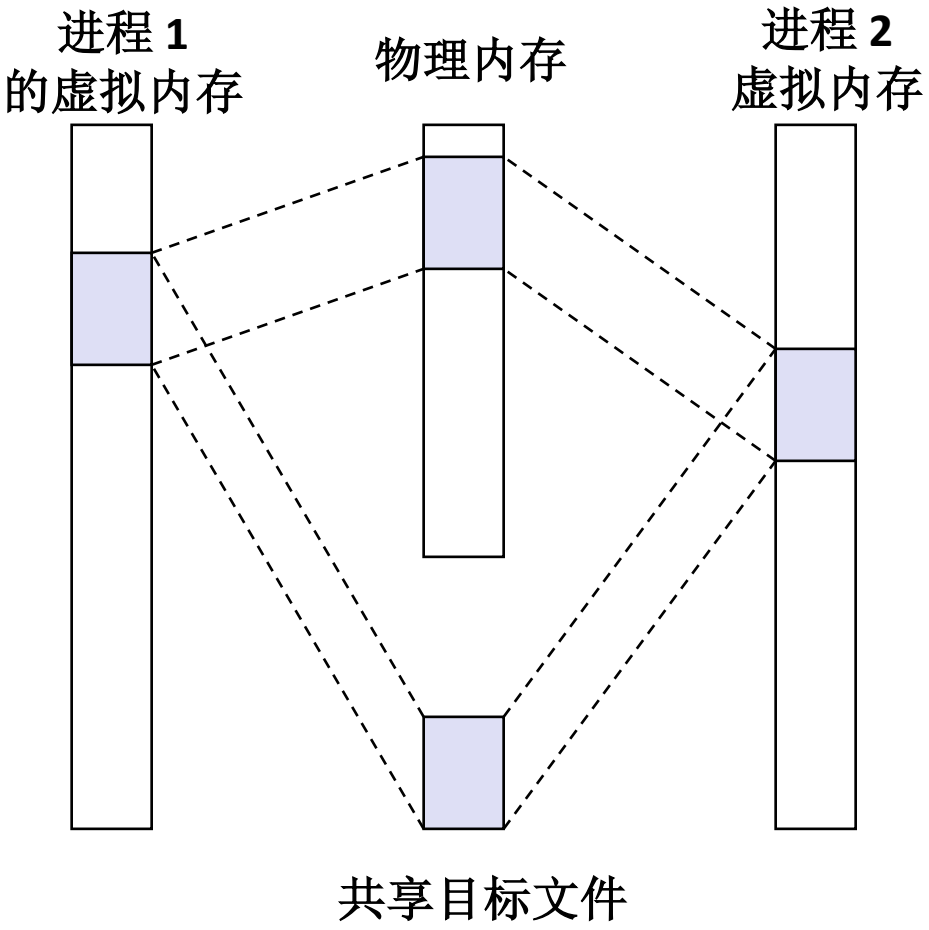
- 虚拟内存区域可以映射的对象（根据初始值的不同来源分）
  - 磁盘上的普通文件（例如一个可执行目标文件）
    - 将文件的节按页面大小分片，对虚拟页面初始化
  - 匿名文件（内核创建，内容全为零）
    - 假设你用malloc等方式，向操作系统申请了一块新的内存区域（比如 4KB），这块区域在虚拟地址空间里有了对应的“虚拟页面”。
    - 第一次引用该区域内的虚拟页面时分配一个内容全是零的物理页
- 初始化后的页面在内存和交换文件空间之间来回交换

再看共享对象shared object



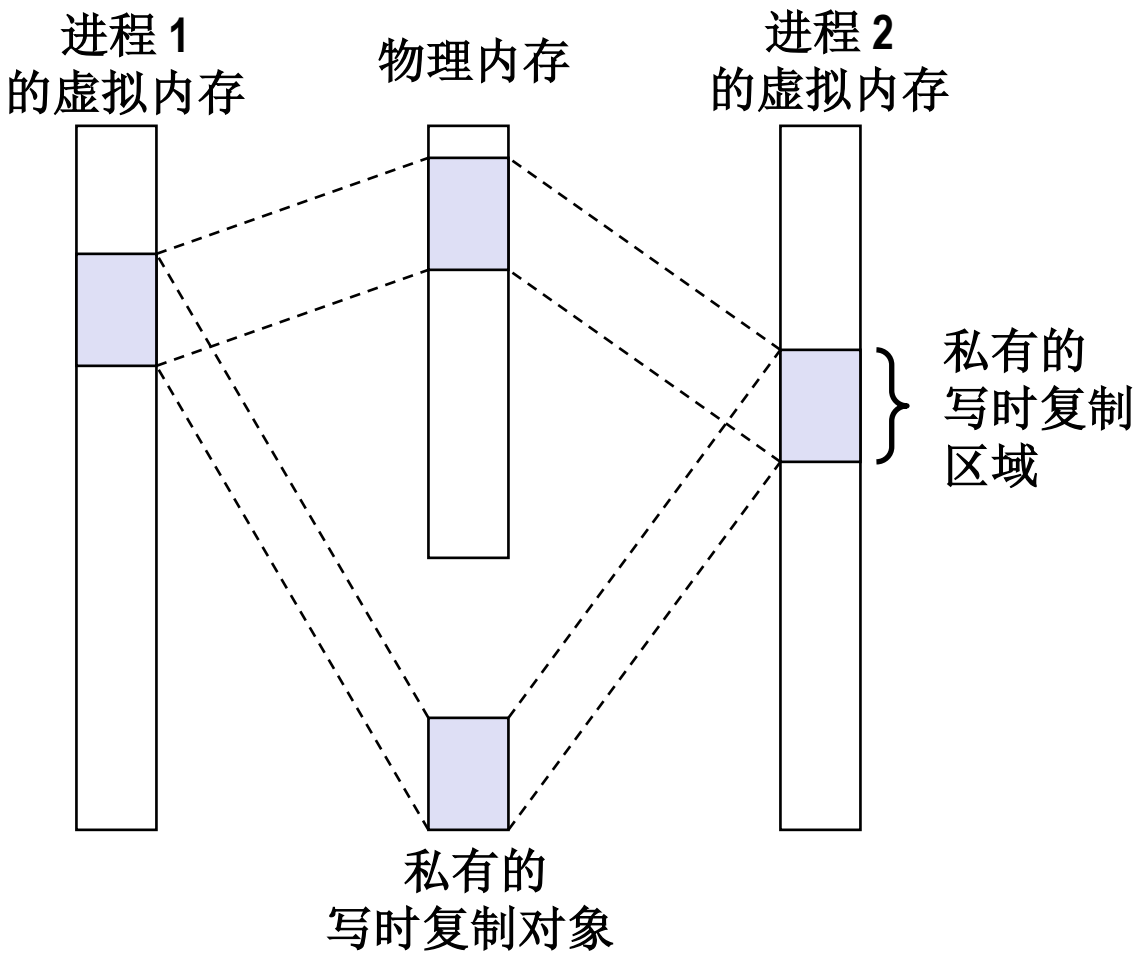
- 被映射至虚拟内存某区域的对象分为共享对象和私有对象两类
- 进程 1 映射了共享对象

再看共享对象



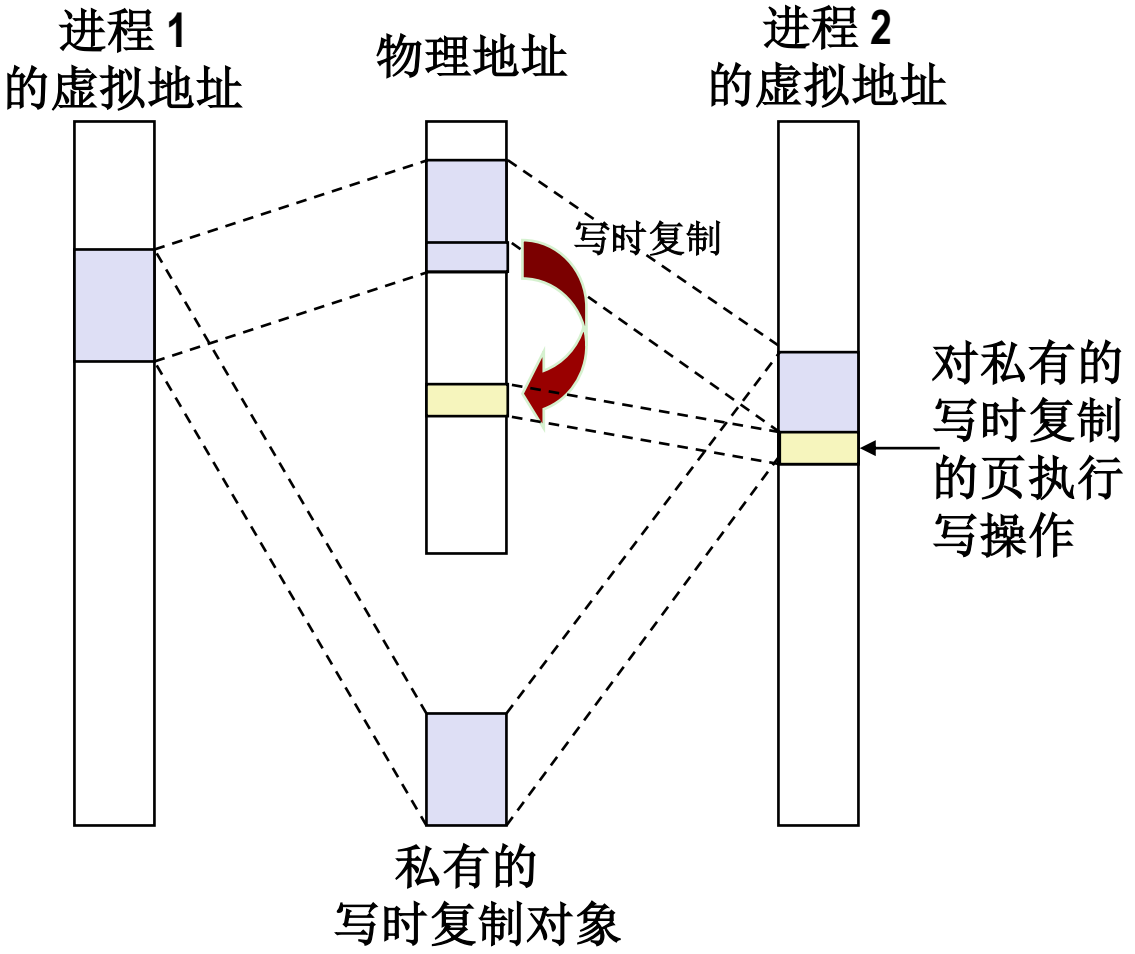
- 进程 2 映射了同一文件
- 注意该目标文件在两个进程虚拟内存空间中的地址可以不同

私有的写时复制(Copy-on-write)对象



- Copy-on-write:最初被多个进程共享、但只要有进程尝试写入时，就会为该进程单独分配物理内存副本的内存区域。
- 两个进程都映射了私有的写时复制 *COW* 对象
- 私有区域相应的页表条目 PTE 全部标记为只读

私有的写时复制Copy-on-write对象



- 写私有页的指令触发保护故障
- 故障处理程序创建这个页面的一个 R/W 副本，更新 PTE 条目
- 故障处理程序返回重新执行写指令

再看 fork 函数

- 虚拟内存和内存映射机制解释了 fork 函数如何为每个进程提供私有的地址空间
- 内核为新进程赋予一个唯一的 PID
- 为新进程创建虚拟内存
  - 创建当前进程的 mm_struct、vm_area_struct 和全部页表的原样副本
  - 将两个进程涉及到的所有页面标记为只读
  - 将两个进程的所有 vm_area_struct 都标记为 COW
- fork 返回时，新进程拥有了与原进程完全相同的虚拟内存
- 随后的写操作通过写时复制机制创建新页面