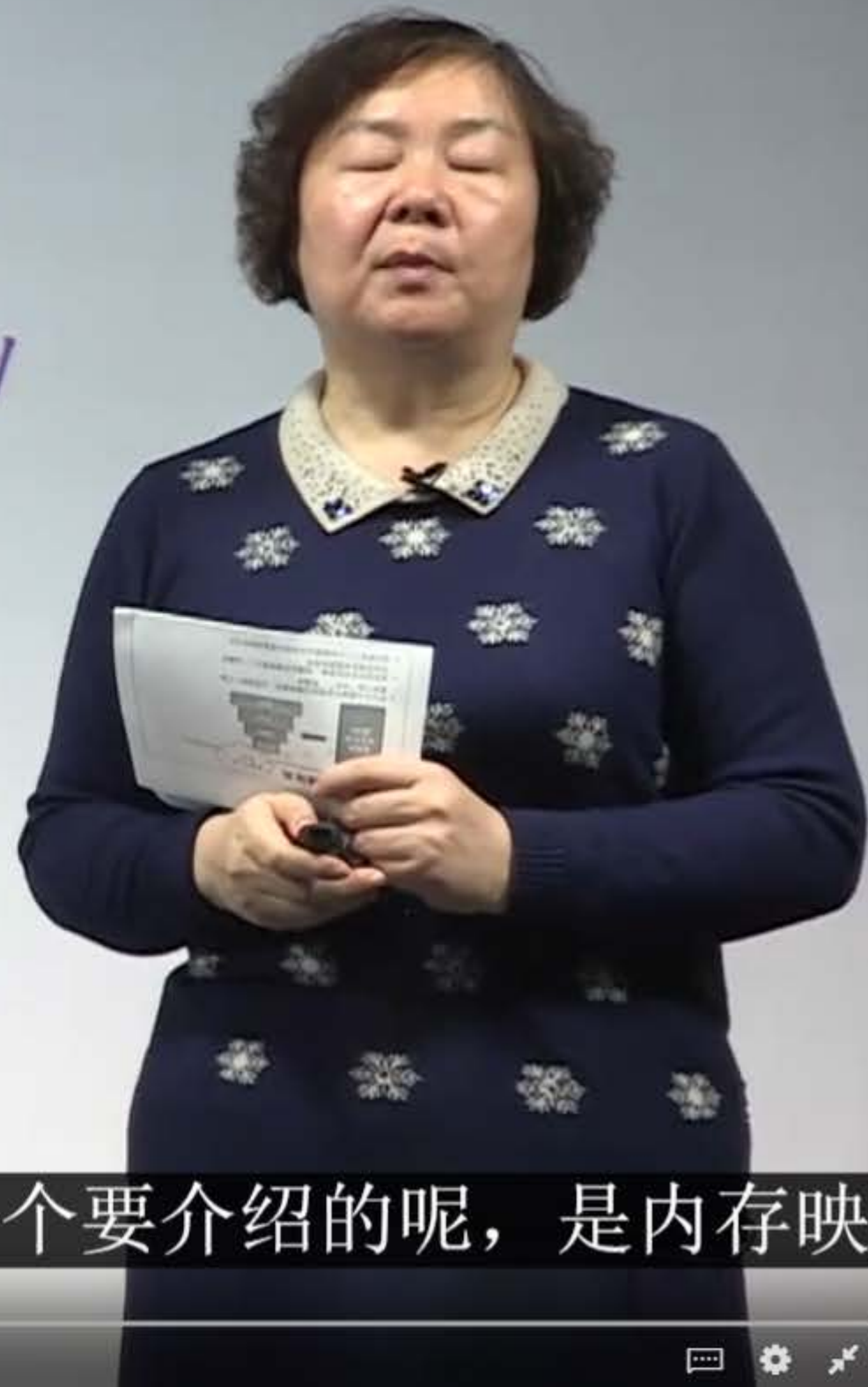


内存映射技术、写时复制

其他相关技术



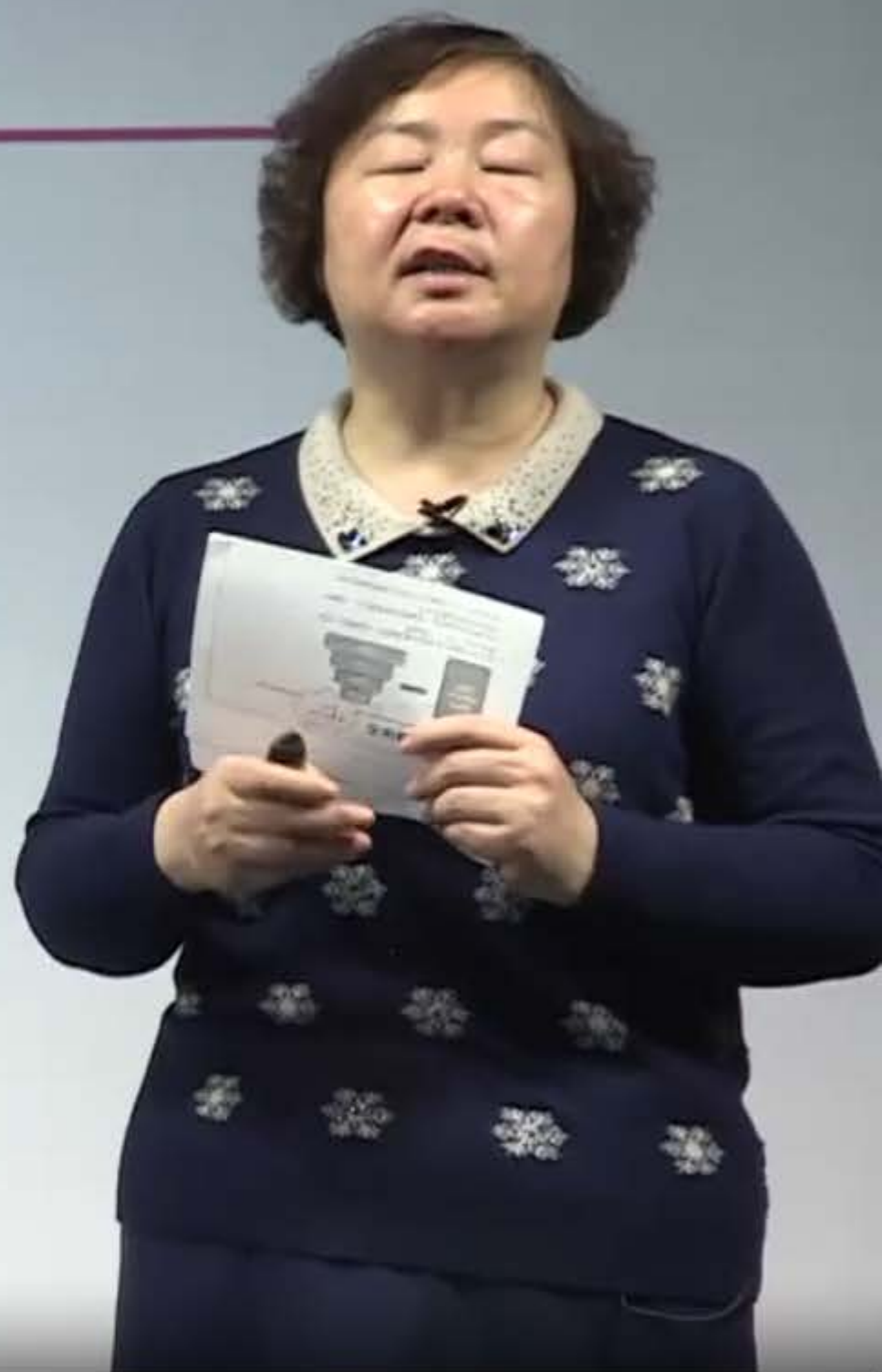
下面呢，我们介绍与存储管理相关的一些软件技术 第一个要介绍的呢，是内存映射文件

内存映射文件

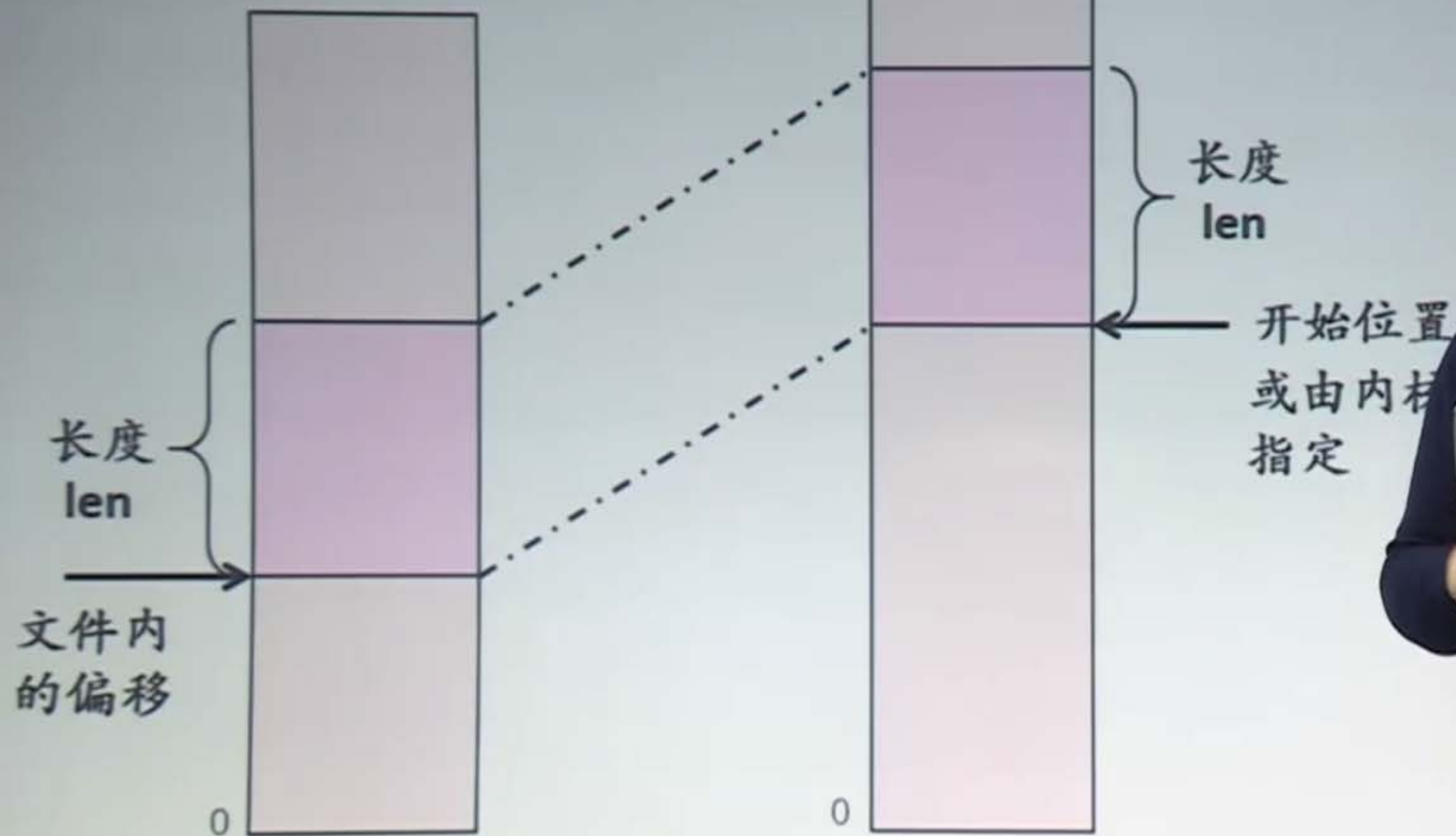
◎ 基本思想

进程通过一个系统调用(**mmap**)将一个文件(或部分)映射到其虚拟地址空间的一部分，访问这个文件就象访问内存中的一个大数据组，而不是对文件进行读写

- ◎ 在多数实现中，在映射共享的页面时不会实际读入页面的内容，而是在访问页面时，页面才会被每次一页的读入，磁盘文件则被当作后备存储
- ◎ 当进程退出或显式地解除文件映射时，所有被修改页面会写回文件



磁盘上的一个文件

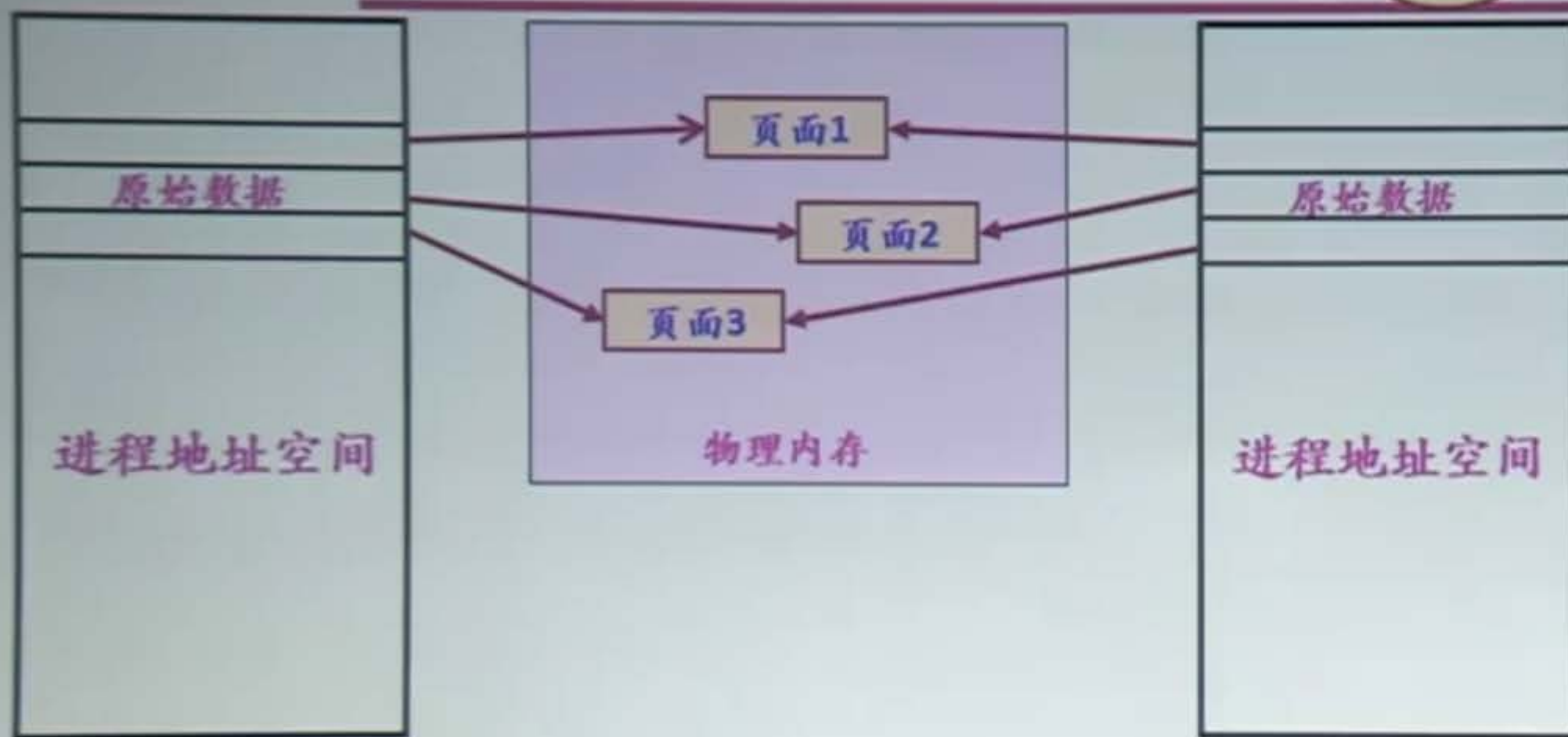


进程的虚拟地址空间



支持写时复制技术

进程试图改变
页面2的数据后



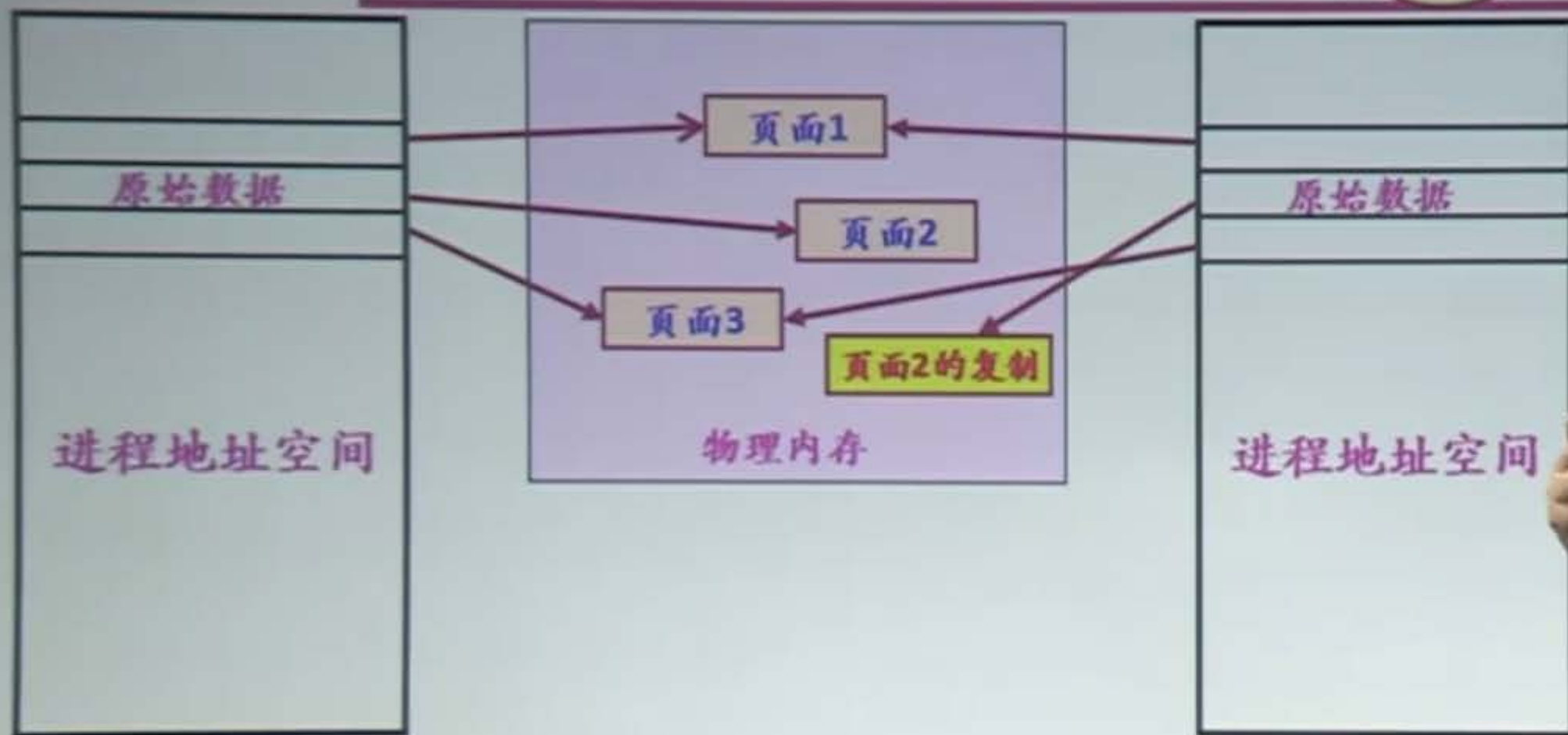
例如：两个进程共享三个页，每页都标志成写时复制

和只读这样一个标志呢，就产生了冲突 就会产生 Page Fault



支持写时复制技术

进程试图改变
页面2的数据后



例如：两个进程共享三个页，每页都标志成写时复制

新复制的页面对执行写操作的进程是私有的，对其他共享写时复制页面的进程是不可见的

本讲重点

- 掌握虚拟存储技术的相关概念
- 掌握虚拟页式存储管理方案的实现
多级页表、反转页表、页表项、地址转换、**MMU**、快表(**TLB**)、页错误(**Page Fault**)、缺页异常处理
- 掌握相关软件策略
驻留集、置换范围、清除策略
置换算法：**OPT**、**FIFO**、第二次机会、时钟算法、**LRU**、老化、工作集
- 了解虚存相关的软件技术
内存映射文件、写时复制



本周要求

重点阅读教材

第3章相关内容: 3.3、3.4、3.5.1、3.5.7、3.5.8、3.6.1、...

重点概念

虚拟存储技术: 虚拟内存 (虚存) 虚拟地址空间

存储体系 存储保护 内存管理单元MMU

快表TLB 页错误 缺页异常处理

页表 页表项 多级页表 反转页表

驻留集 置换范围 清除策略 页缓冲技术

页面置换算法: 最佳、先进先出、第二次机会、时钟

最少使用、老化、工作集 模型

内存映射文件 写时复制技术



下面呢，我们介绍与存储管理相关的一些软件技术 第一个要介绍的呢，是内存映射文件 通常一个进程在运行过程中，如果要使用文件，它要通过打开文件、读文件、写文件、关闭文件这样一些操作 而通过文件系统来完成这个文件的处理过程 而内存映射文件呢，它的基本思想是 进程可以通过一个系统调用，比如说 Linux 里头可以通过 `mmap` 这样一个系统调用，将一个文件，或将文件的某些部分 映射到了我的虚拟地址空间的一部分 因为我们知道虚拟地址空间，还有很大的空间，没有内容，所以我们可以把文件整个文件或者文件的一部分，映射到这个虚拟地址空间的其中的一部分，然后 对文件的访问，就如同访问内存中的一个大数组一样，而不是 用前面我们说的读文件、写文件、判断文件是否结束，这样的系统调用了 所以这就是基本的思想，那么在多数实现的时候呢，在 映射共享的页面时呢，不会实际读入这个页面 就是把文件映射到了内存的这个虚拟地址空间，那么 实际上呢 是不会读入页面的内容，而是要真正访问到这个页面的时候，才会被 从磁盘读入内存，也就是换句话说，是利用了虚拟存储机制当中的 缺页异常这块，来把相应文件内容读入内存的，所以就是利用了虚存 来做这件事情。那么当然 用完这个文件之后呢，退出了，进程退出，或者用完这个文件的时候呢 再需要用一个系统调用，把这种映射关系解除 同时，把相应的页面的内容写回到磁盘 这就是内存映射文件，我们来看一张图，这是我的进程的地址空间 地址空间有很多内容，但是也有很多空闲的地方对吧，那然后这时候我们可以把 这个文件，这是文件的整体内容，我们可以通过把文件的一部分，长度为 `L Len` 的内容，把它映射到了我的虚拟地址空间的一部分 当然这个部分映射到什么位置，可以由用户来指定，也可以由内核来指定，这就是 内存映射文件的技术，然后对文件的各种操作，其实就是对内存 空间的，相当于你的读写操作就完成了，赋值语句就可以做到这一点了 就不需要用读写这样一个显示的文件操作 下面我们介绍另一种写时复制技术 在讲进程创建的时候 我们在谈到 Linux 实现进程创建时呢，采用了一种写时复制技术 而这种技术的实现、支持呢 是在存储管理里头完成的 下面我们来看一下，那么这有两个进程 这两个进程呢，实际上呢 共享了三个页框 就是共享了物理内存的三个页框 那么这三个页框是由于有两个，或两个以上的进程共享 所以每一个页框呢，实际上都标记成写时复制 写时复制的标记呢，是由一个 只读标记，再加上另外一个标记，这个组合成的一个标记 那么当某一个进程 试图要改变其中一个页面的内容的时候 那么这个时候呢，它要做一个写操作，那么 和只读这样一个标志呢，就产生了冲突 就会产生 Page Fault 进入了操作系统 操作系统

呢 检查出它是一个写时复制的这样一个操作，于是呢就会在 内存里头另开辟一个页面 把相应的内容写到这个页面里头 而新复制的这个页面呢，对于执行写操作的这个进程来讲呢，它是一个私有的 对于其他的进程，它们是共享 这个写时复制页面的进程的，它是看不到这样一个结果的 这是一个写时复制技术的一个基本的实现 那么我们在进程创建的时候呢，父进程创建了子进程 在一开始的时候呢，父子进程实际上是共享一块 虚拟地址空间，而这个虚拟地址空间是标记成写时复制的 一旦子进程要用自己的代码和数据，然后替换 从父进程那里继承的代码和数据的时候，那么这个时候呢它要做写操作 那么就要把写时复制这个权限 只读的权限冲突，那么通过操作系统 会给予进程分配新的空间，来把内容 加载到内存，这就是 Linux 里头来实现这个 Fork 这样一个创建进程的这个过程当中，利用的写时复制技术 本讲的内容呢非常多，啊，我们这里可以看一下虚拟存储技术的相关概念 虚拟页式存储管理方案的实现，那这里头呢有多级页表、反转页表、页表项的设计，还有硬件完成的地址转换 快表，以及产生了 Page Fault 之后，软件要做的处理 操作系统为了支持虚拟页是存储管理方案，还提出了相应的策略 比如说分配给一个进程多少页框 如果内存没有空闲页框了 需要进行置换，置换的范围和置换的算法 如何回收分配给进程的页框 清除策略，并且利用这种清除策略带来的一些 页缓冲技术。置换算法有若干，另外呢还有 虚存相关的一些软件技术、内存映射文件和写时复制技术 那么这是本讲之后大家要阅读的教材 我们也可以看到列出了很多的概念，希望大家能够重点去掌握这些 概念。好，今天的课程呢我们就上到这里，谢谢大家！