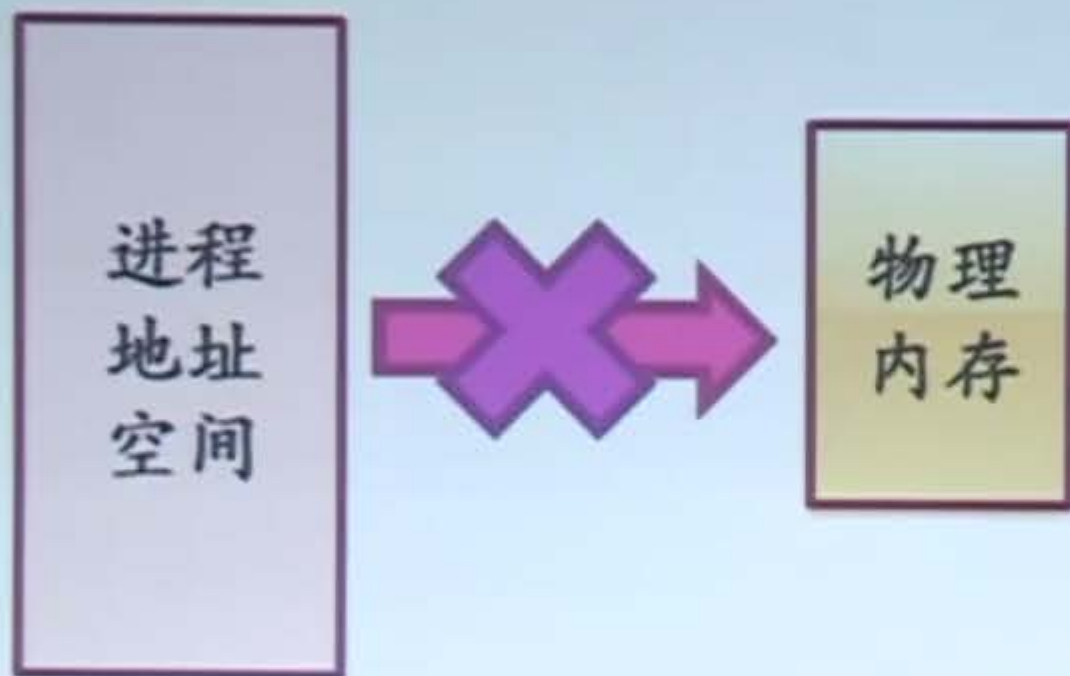


交换技术 SWAPPING



内存不足时如何管理？



怎么办呢？



内存“扩充”技术

- ◎ 内存紧缩技术（例如：可变分区）
- ◎ 覆盖技术 **overlaying**
- ◎ 交换技术 **swapping**
- ◎ 虚拟存储技术 **virtual memory**

如何解决在较小的
内存空间运行较大
的进程呢？



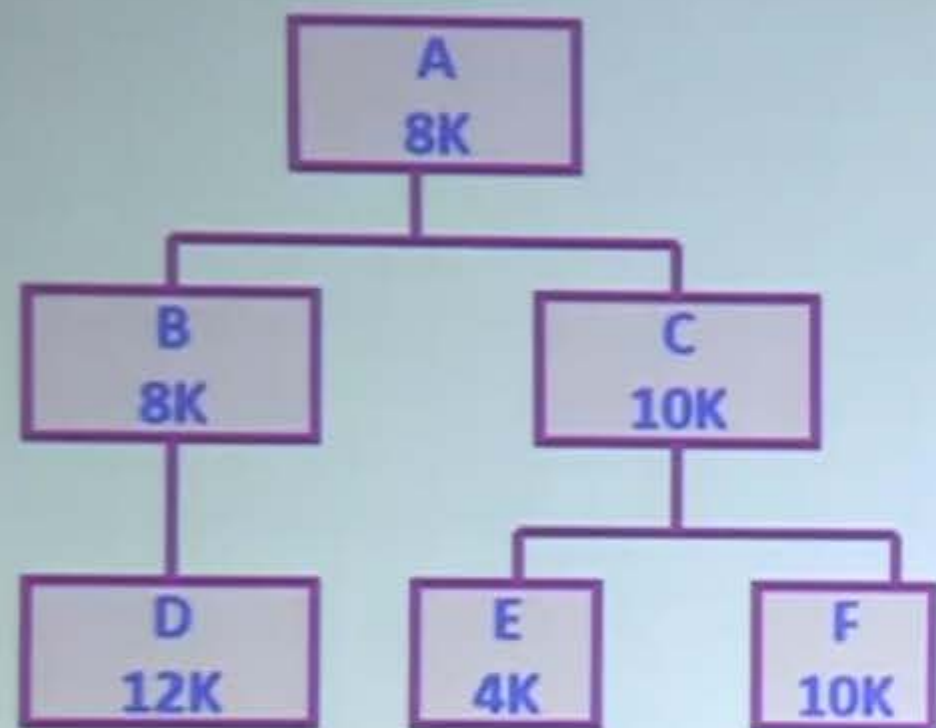
覆盖技术(OVERLAYING)

- ◉ 解决的问题 → 程序大小超过物理内存总和
- ◉ 程序执行过程中，程序的不同部分在内存中相互替代
 - 按照其自身的逻辑结构，将那些不会同时执行的程序段共享同一块内存区域
 - 要求程序各模块之间有明确的调用结构
- ◉ 程序员声明覆盖结构，操作系统完成自动覆盖

主要用于早期的操作系统

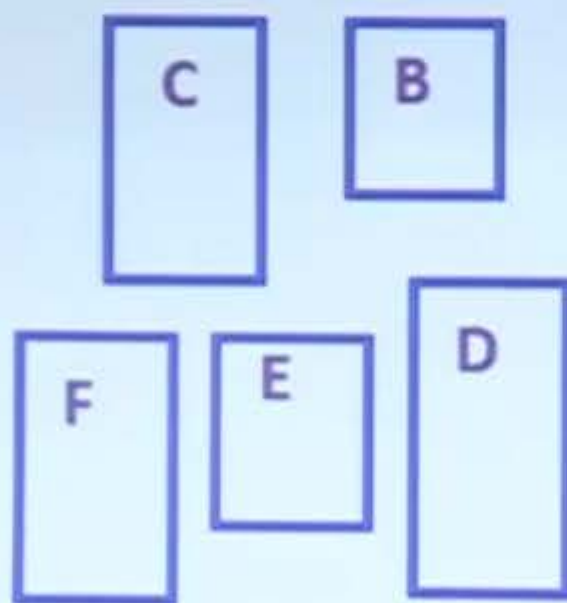


覆盖技术示例



程序X的调用结构

缺点：对用户
不透明，增加了
用户负担



物理内存

交换技术(SWAPPING)

◎ 设计思想

内存空间紧张时，系统将内存中某些进程暂时移到外存，把外存中某些进程换进内存，占据前者所占用的区域（进程在内存与磁盘之间的动态调度）

◎ 讨论：实现时遇到的问题

- ✓ 进程的哪些内容要交换到磁盘？会遇到什么困难？
- ✓ 在磁盘的什么位置保存被换出的进程？
- ✓ 交换时机？
- ✓ 如何选择被换出的进程？
- ✓ 如何处理进程空间增长？



关于讨论的问题

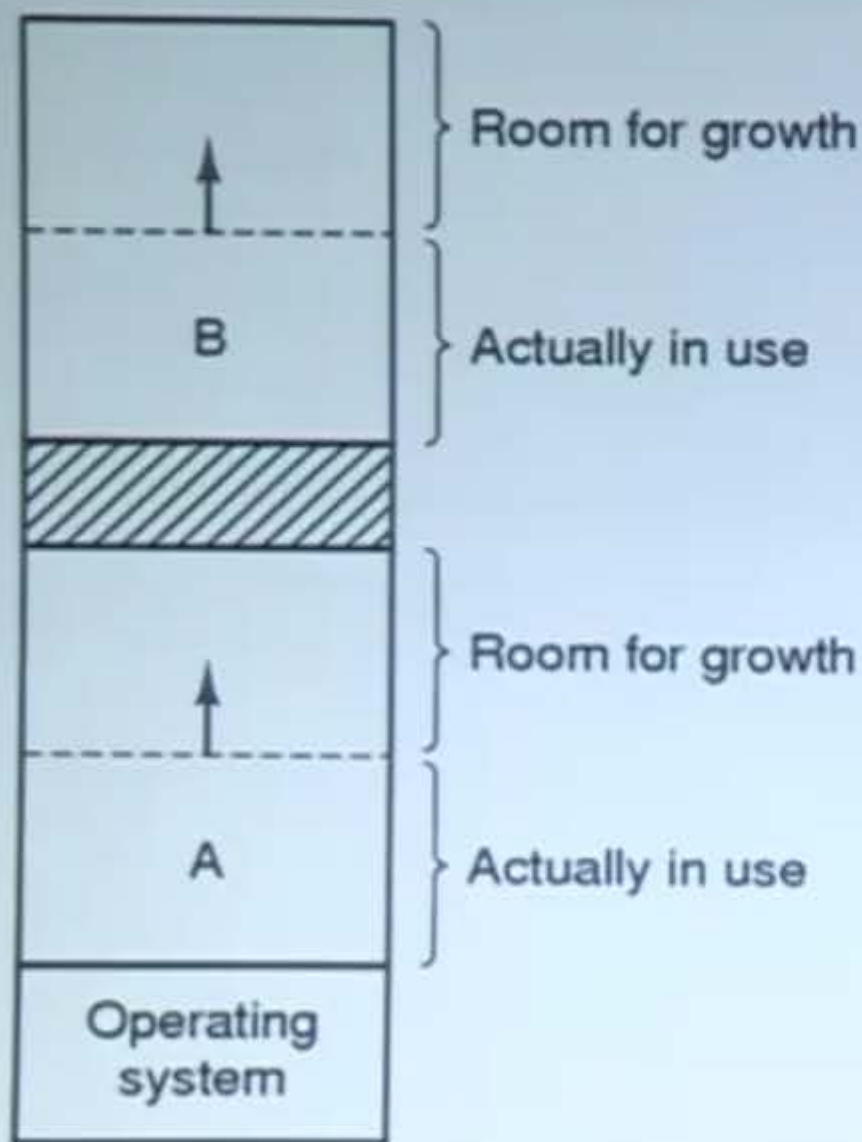
- 运行时创建或修改的内容：栈和堆
- 交换区：一般系统会指定一块特殊的磁盘区域作为交换空间（**swap space**），包含连续的磁道，操作系统可以使用底层的磁盘读写操作对其高效访问
- 何时需发生交换？
只要不用就换出（很少再用）；内存空间有不够的危险时换出
与调度器结合使用
- 考虑进程的各种属性；不应换出处于等待I/O状态的进程

swapper

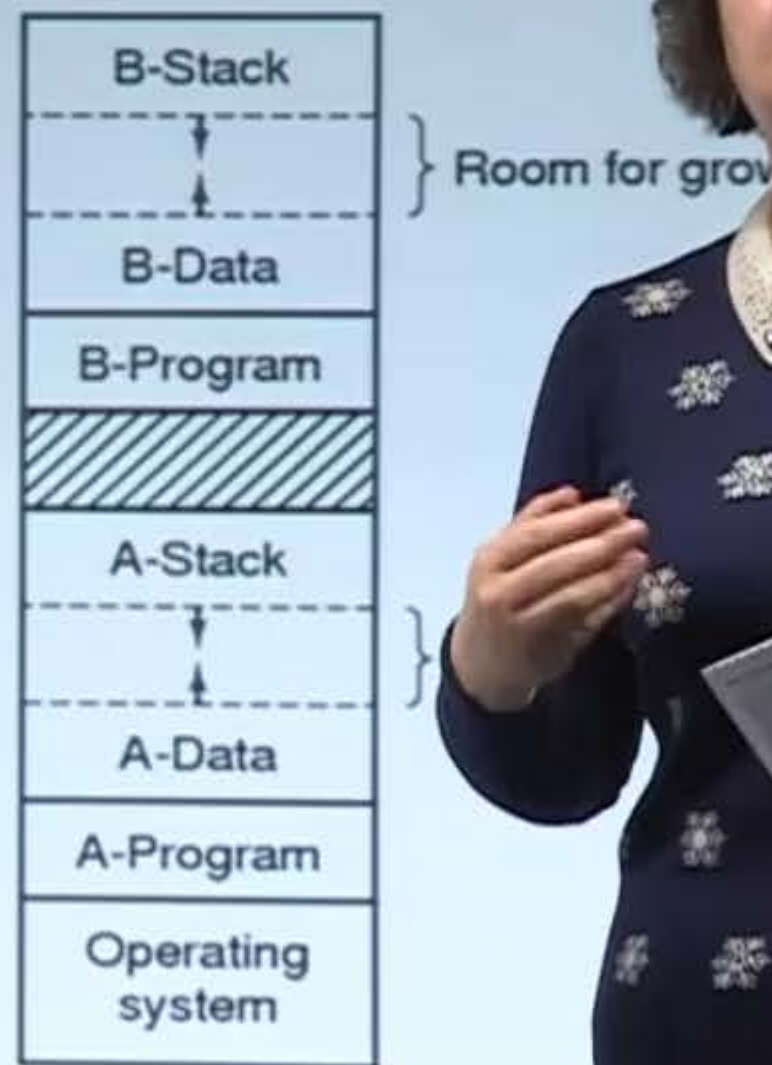


进程空间增长的困难及解决

- 数据段
- 栈段



(a)



(b)



本讲重点

- ◎ 掌握地址重定位的概念
- ◎ 掌握基本内存管理方案
- ◎ 掌握物理内存管理技术
- ◎ 了解交换技术基本思想



最后呢，我们介绍了交换技术 主要是为了解决内存空间不足的这个问题。

本周要求

- 重点阅读教材

第3章相关内容：3.1、3.2

- 重点概念

地址重定位 地址转换 地址映射 地址翻译 地
逻辑地址/物理地址 虚拟地址/实地址 相对地
单一连续区 固定分区 可变分区
页式 段式 段页式
物理内存管理（位图、空闲区表、空闲区链表）
内存分配算法（首先适配、最佳适配、最差适配）
内存回收算法（上相邻、下相邻、上下相邻、上
碎片（内碎片、外碎片） 紧缩技术 交换技术



下面我们介绍交换技术 交换技术的引入呢是面临了这样一个问题 一个大的地址空间怎么样来装入一个小的 可用的物理内存？也就是内存不足的时候，我怎么样 来管理呢？那么一个大的地址空间是不可能装到一个 小的可用的物理内存里头的，因此我们该怎么办呢？[声音] 交换技术的引入，实际上就是为了解决刚才我们描述的这种场景 总结一下就是，如何解决在一个较小的啊内存空间里头 运行一个较大的进程这样一个问题 主要可以采用的技术呢我们叫做内存"扩充"技术 当然了 这个扩充加了个引号，不是买一个新买一个内存条然后把内存 物理内存扩大，而是说采用软件技术或者软硬件结合 来使得一个大的进程在一个较小的 物理内存空间上运行。有一种情况就是我们前面所介绍的可变分区存储管理方案当中 会出现的碎片问题。那么由于系统中有很多的碎片 可能会导致一个进程 进入内存不可以，因为碎片虽然很多，碎片的总和能够满足进程的需求 但是由于碎片都分散在不同的位置，所以呢这个进程还是进不了内存 当然了如果是这种情况我们只需要采用内存 紧缩技术就可以了，所以内存紧缩技术也是一种把内存 扩展的这么一个思路 那我们里头呢主要介绍的呢是覆盖技术 还有交换技术。另外一种内存扩充技术就是我们下一讲 非常重要的就是要介绍的虚拟存储技术 我们里头呢只介绍覆盖技术和交换技术 所谓覆盖技术呢它要解决的问题呢就是当 一个程序的大小超过了物理内存总和 那这种情况下，那么没有别的办法我们只能采用覆盖技术 具体的做法是，我们考虑到 进程在执行的过程当中它的不同的部分呢，实际上在 内存当中不用同时出现，也就是可以互相啊替代 那么我们可以按照 进程自身的逻辑结构，把那些不会在同时执行的这个程序段 让它可以共享同一块内存区域，因此呢 要求程序员，因为他对自己的程序了如指掌，所以要求程序员 能够给出来自己程序的各个模块之间的一个调用结构 啊调用结构，把这个调用结构给出来之后呢，然后呢 通过相应的覆盖技术呢，能够完成这个不同的 啊程序段共享同一块物理内存的目的 那么这个工作呢是 程序员自己完成的，也就是程序员要自己把这个调用结构 找出来，并且呢声明覆盖结构告诉操作系统 操作系统只是完成一个覆盖的过程 当然了 这个技术呢说实在的啊已经 是很久很久以前的一个技术了。主要的是 用于早期的操作系统，那么那个时候 如果程序大于物理内存，那就只能通过这种覆盖技术来解决问题了 我们有一个小小的示意啊 这是一个啊程序，这个程序有一个调用的关系 那么我们假设啊 A 调用了 B 或者 C，那么调用 B 就走 B 这个分支的话就 C 就不执行了 那么同样道理如果是走 C 这个分支，C 再往下走呢可能执行 E 也可能执行 F 那么我们可以看到在这样一个结构当中我们可以 建立这

么一个由程序员来建立这么一个覆盖结构，这个覆盖结构就等于说 把内存啊跟这个进程相关的分成三个区域，第一个区域放 A 这就啊 A 占据的常驻这个区域，然后 涉及的两个覆盖区，一个覆盖呢是 0 一个覆盖是 1。那么这个区域大小呢是根据了 B 和 C，比如说覆盖 0 是根据 B 和 C 中哪个长度大就按长度大的那个设计 那么覆盖 1 呢，是 D、E、F 这三个啊 段落啊程序段哪个大就按那个来设计，所以我们来看 那么我们只需要这么一大块内存就够了 啊，所以我们需要 8K、10K、12K 一共需要 30K 就可以了 那么我们知道 B 和 C 不能同时，不可能 同时执行，所以呢我们可以来看一下 B 可能先占据着些区域 然后呢 C 呢再占据着区域，它们两不同时所以都可以占据一个区域 同样道理 D、E、F 也 不可能同时执行，所以呢 D 呢先占用然后呢 E 占用或者是 F 占用，根据情况谁执行谁占用这个区域 这样的话呢本来一个需要很大的一个区域呢我们只需要一个很小的区域就可以解这个问题了 当然了这个缺点是啊用户是对用户是不透明的，程序员要自己 要完成这个覆盖的关系的啊结构的描述 操作系统只是来做这个执行。这是覆盖技术当然这个现在很少用到了 下面我们介绍一下交换技术 那么交换技术呢实际上是我们现在虚存技术的一个最初的一个设计思想，那现在演化到现在就变成了虚拟存储技术 那么所谓交换技术当初为什么引入呢？只要是当内存空间紧张的时候 那么系统呢会把在内存当中的一些进程 暂时移动到啊移动到磁盘上外存 那么然后呢再把外存的一些进程把它换进内存，然后占据前者所占用的 区域，也就是说进程会在内存磁盘之间呢动态的来调度 最早交换技术是为了解决这样一个问题引入的。那么我们来看一下 在实现交换技术的时候呢我们要考虑的几个问题 第一个问题就是哪些进程的内容，进程有很多啊内容，哪些进程的内容要把它交换到磁盘上保存在磁盘上 那么会遇到一些什么困难？那么第二个问题 在磁盘上什么位置来保存被置换出去的这个进程 第三个问题交换时机，什么时候做这样一个啊这个 工作？第四个问题如何选择 被换出的进程？第五个问题如何处理进程空间的生长？好！那我们来看看 Swapper 这个交换啊交换技术的解决刚才我们说的几个问题 那么第一个问题那么交换 的内容是什么？其实交换的内容往往是在运行过程中 动态创建或者是已经修改过的内容，那么这里头主要是栈 和堆啊这里头的内容，像代码呀和一些 静态的数据没有改变不需要再保存到磁盘上，因为 在可执行文件里这些东西都有，不需要保存，所以呢这是 哪些内容要交换到磁盘上呢？是这些内容 然后我们来看看那么交换到磁盘上什么位置呢？通常情

况下呢，在磁盘上会建立一个区域叫做交换区啊交换区 那么系统会把一块特殊的磁盘空间呢 把它设计出来，然后作为这个交换空间 包含了一个一部分的啊磁道 那么这块区域呢一般呢是不经过文件系统 由操作系统用底层的一些磁盘读写操作来对它进行访问，这样效率 比较高，因为你要经过文件系统的话可能呢 会耗费一些文件查找这样一些工作 那么这块区域在磁盘这块区域呢操作系统直接管起来。啊这样效率比较高，那么这个区域呢就称之为交换区 那当然了不同的操作系统对这个区域的名称 叫法不一样，比如说在 Windows 里头呢，它就不叫交换区 它起了一个名字叫做页文件啊 Pagefile 当然这是不同的系统对这个区域的叫法不一样 那么什么时候需要发生这个交换的这个过程呢？ 其实主要的原因就是，一种情况是，我这个进程 很少使用了，不用了，就把它换出去，不要占据内存 第二种情况呢，是更多的，就是内存空间不够了，或者是 有不够的这种潜在的危险的时候，就立刻启动这个 交换技术，然后把它换出去，就启动这个程序，把它换出去 当然这个过程呢，往往是和调度器，调度程序结合起来使用的 那么在交换的过程中呢，我们还要考虑一个问题 就是说，哪些进程，不能把它交换到磁盘上 要考虑到进程的各种属性。就像我们前面讲到紧缩技术一样 有一些进程，在内存里是不能随意搬家的，同样道理 这些进程，也不能随意把它 交换到磁盘上。因为，你正在等待 I/O，你把它交换到磁盘上，就会影响 I/O 的结果 那么关于进程空间增长的这个困难呢 以及解决方案呢，我们知道数据段和 栈段，它总是会增长的，堆，它会不断地增长 栈也不断地会增长。那我们怎么样来解决这个增长的这个 预留空间的问题呢？ 那么一种方案我们的书上给出了两种方案。一种方案呢，就是我们会 这是进程，比如说进程 A，进程 B。那么进程 A 呢占据这一块儿空间，进程 B 呢 这个地方预留了一部分。这一部分呢，就是 考虑到这个进程可能会增长，会长长，那么把这个空间留出来 那么进程 A 呢，实际用了这么多，然后留了一部分空间去增长，那么这个还留了一点 那么进程 B 也是一样。那大家知道，这样一个同向增长，我们有两种增长，一种就是这个 栈的增长，一种是堆的增长，那么这样同向增长呢，你的空间留多大，是一个问题 那么第二种方案呢，大家可以看是这种方案。那么这种方案实际上我们可以看一下 那么，进程 A，啊占据了这么一块儿空间 那么然后呢，我们可以看到，数据呢是向上增长，而栈呢，是向下增长 向下增长。那这样的话呢，它是同向增长，所以我们留了一块儿空间呢，让它 同向增长。那这也是一种解决进程空间增长的这个这

个问题的方案，所以我们考虑到这个交换技术呢，主要是为了解决一个进程在运行过程中，啊，系统的内存不够了。那么这样的话，我们要通过交换把内存空间的相关的，这个这个内容呢，放到磁盘上。然后呢，再把磁盘上内容再调进来，占据刚才的空间。这样的话让这个进程很多进程都能运行起来。交换技术主要解决这个问题 本讲的重点呢其实有四个，一个呢是地址重定位这个概念，地址重定位，地址转换 地址变换，地址映射，地址翻译 第二个呢，就是要要求大家掌握基本的内存管理方案 一共六种。我们重点呢是页式管理方案，和可变分区 那么物理内存管理呢，有不同的数据结构，有不同的算法 最后呢，我们介绍了交换技术 主要是为了解决内存空间不足的这个问题。这是本周的教材的阅读要求，以及重点的一些概念 那么非常多，其实主要是名词、术语太多，同一个术语有多种说法，我们都列在这，希望大家呢能够熟悉这些术语 好，今天的内容呢 就介绍到这里，谢谢大家