

单一连续区、固定分区、可变分区

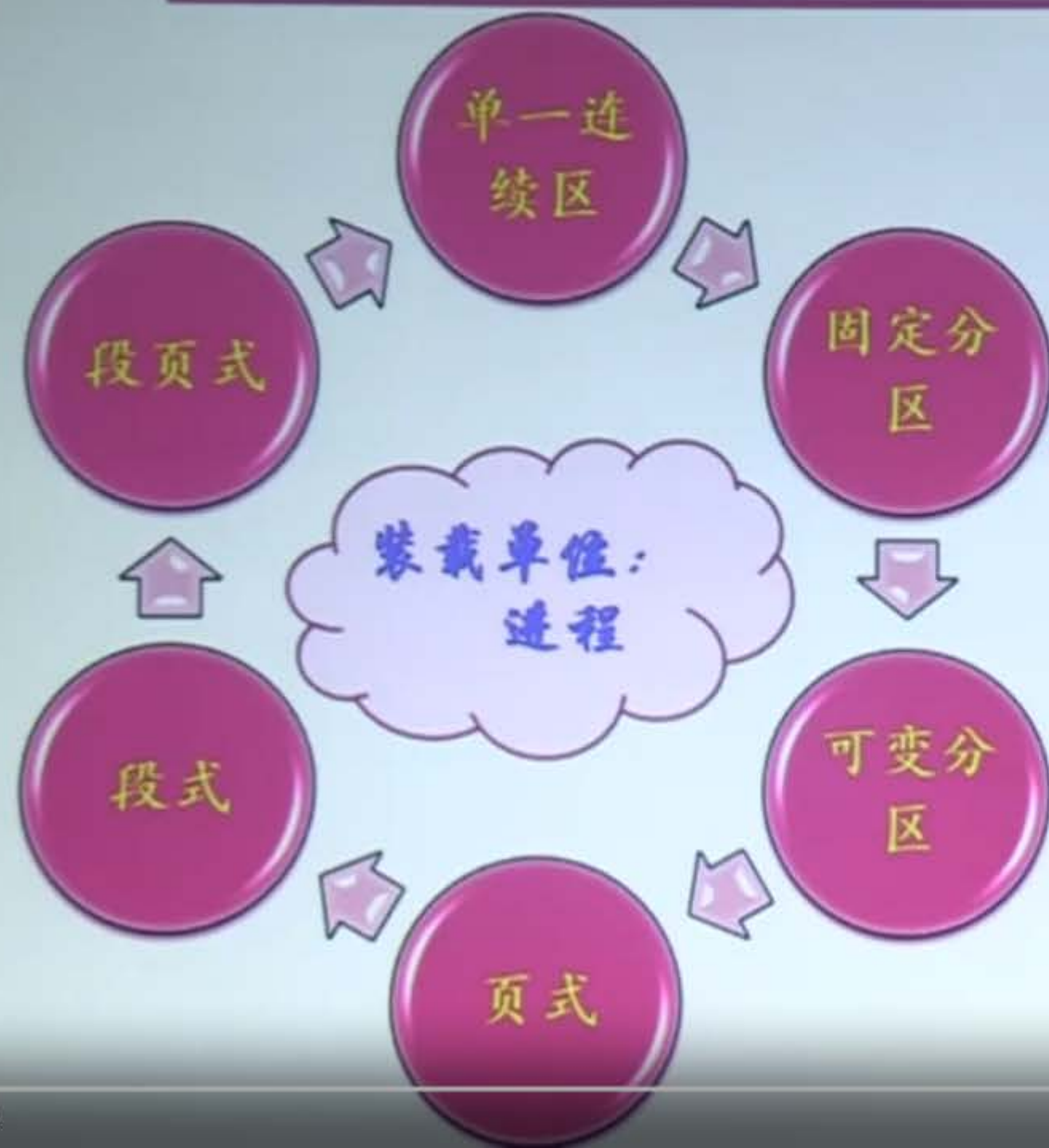
# 基本内存管理方案1

整个进程进入内存中一片

连续区域



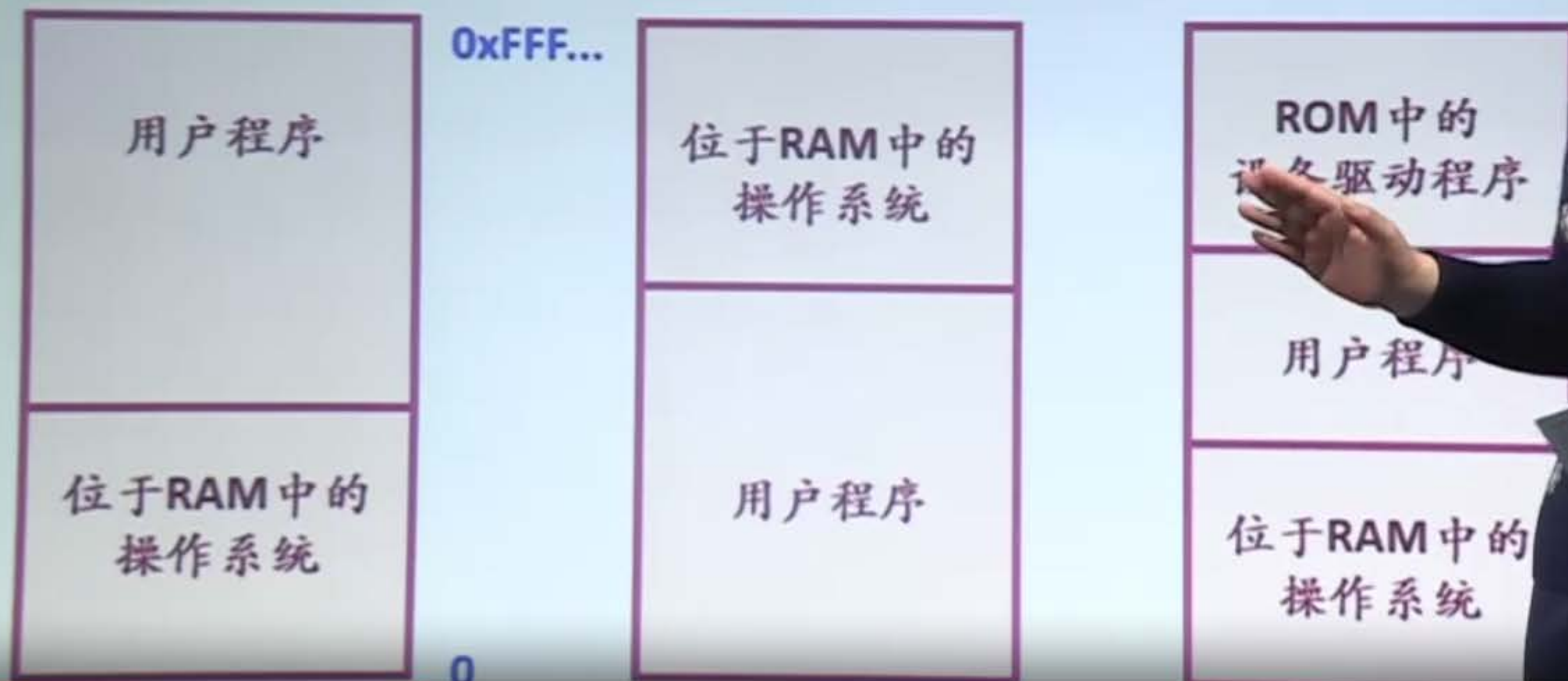
# 基本内存管理方案





# 单一连续区

特点：一段时间内只有一个进程在内存  
简单，内存利用率低



# 固定分区

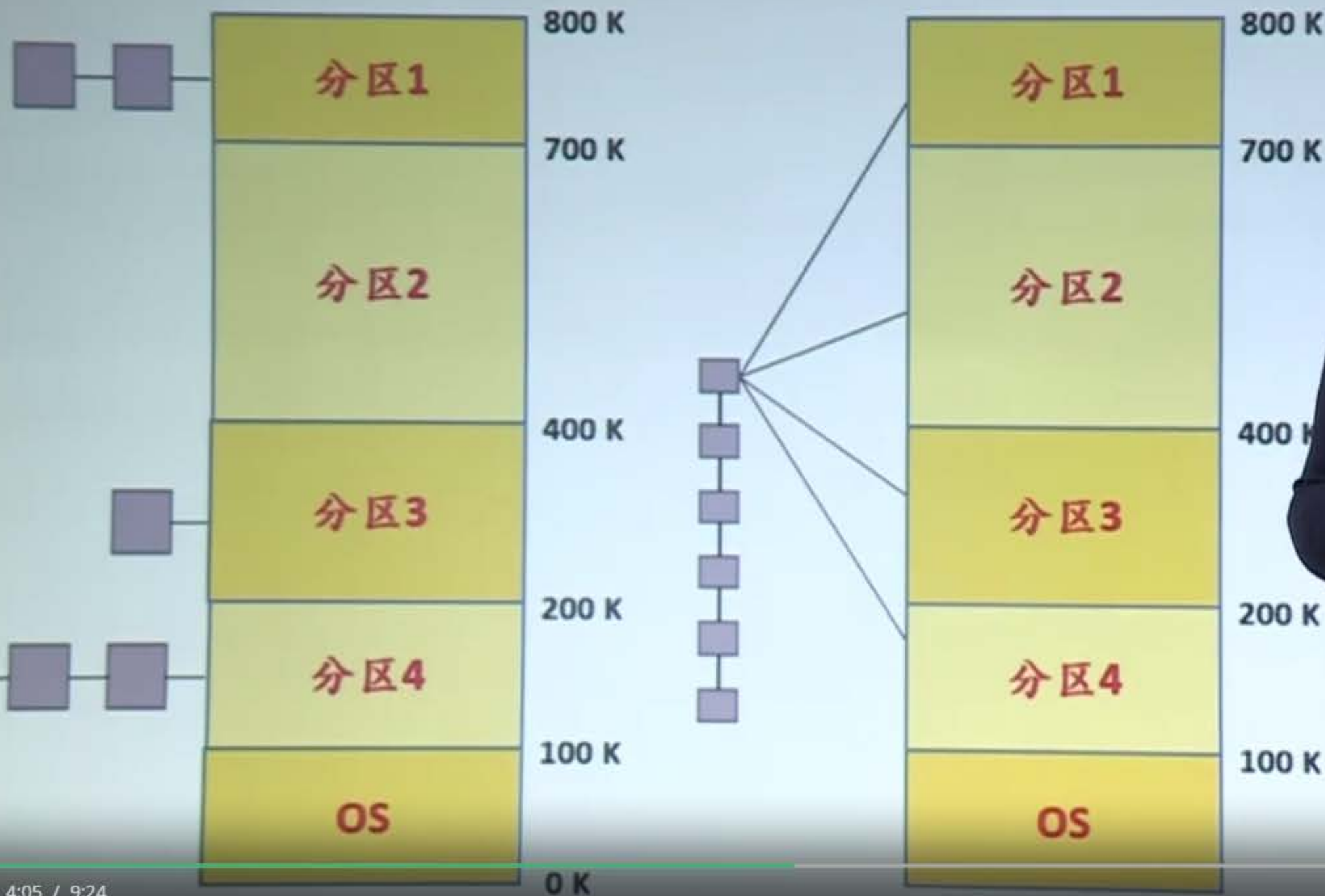
- 把内存空间分割成若干区域，称为分区
- 每个分区的大小可以相同也可以不同
- 分区大小固定不变
- 每个分区装一个且只能装一个进程



每个分区呢，装一个进程，也只能装一个进程，这就是固定分区的基本思想



# 固定分区示例



# 可变分区



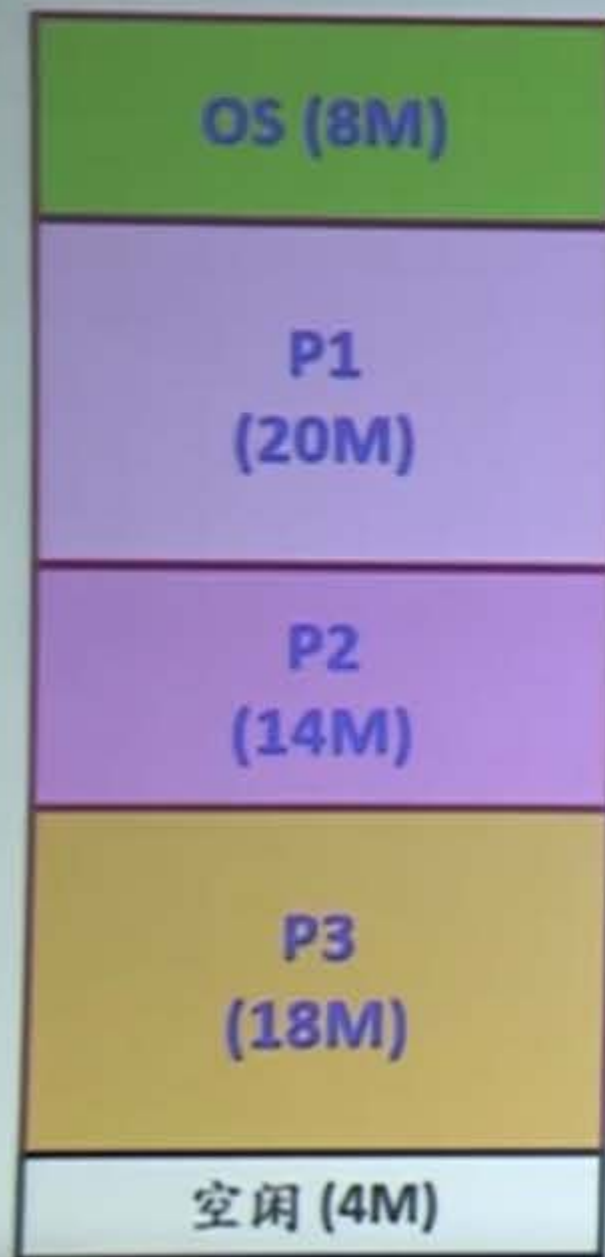
- 根据进程的需要，把内存空闲空间分割出一个分区，分配给该进程
- 剩余部分成为新的空闲区

我们看一个例子 假设进程p1进入内存p2进入内存





# 可变分区



- 根据进程的需要，把内存空闲空间分割出一个分区，分配给该进程
- 剩余部分成为新的空闲区



# 可变分区



- 根据进程的需要，把内存空闲空间分割出一个分区，分配给该进程
- 剩余部分成为新的空闲区

洞



那么就肯定会导致一个结果，内存的利用率就会下降



# 碎片问题解决

- ◎ 碎片 → 很小的、不易利用的空闲区  
→ 导致内存利用率下降
- ◎ 解决方案 → 紧缩技术 (**memory compaction**)  
在内存移动程序，将所有小的空闲区合并为较大的空闲区  
又称：压缩技术，紧致技术，搬家技术
- ◎ 紧缩时要考虑的问题  
系统开销？ 移动时机？

至少在某些时刻，某些进程不能够搬家 这就是碎片问题的解决。



下面我们来介绍基本的内存管理方案，我们先介绍三个，单一连续区、固定分区和可变分区，它的特点是整个进程进入内存一片连续的区域 进程作为一个整体进入一片连续的区域 那么首先我们来看，不同的内存管理方案 到目前为止，那么加载的单位是进程，就整个进程要进入内存 因此呢，我们有几种不同的方案 单一连续区，固定分区，可变分区 这是我们这一部分要介绍的，然后我们还有 页式存储管理方案，段式存储管理方案，还有段页式存储管理方案 我们先介绍这三种 这三种的特点就是整个的进程 作为一个整体，要占据内存一片连续的空间 所以第一种方案呢，我们成为单一连续区，它的基本思想是 一段时间内只有一个进程进入内存，在内存里头 当然他这个好处就很简单 一次只有一个进程在内存，但是他的内存的利用率就比较低 当然这个早期，有这样的简单的方案 那么有三种不同的布局 一种就是说，用户程序，总是从 在占据高地址的部分，那么操作系统呢，总是占据低地址的部分 还有一种能就是说，操作系统占据高地址部分 而用户程序总是从零开始进行装入内存 也有一种呢，早期在DOS里用的方法就是 有一些设备驱动程序，放在这个高地址的部分，然后呢操作系统占据低地址部分 那么用户程序呢，可能在中间，那当然这是一个比较简单的单一连续区的方案 现在很少能见到，但只是一种介绍 那么早期还有一种方案，叫固定分区，所谓固定分区呢，就是把整个内存空间 先分割成若干区域，那么每个区域我们称之为分区 每个分区的大小呢，可以相同，也可以不同 但是呢，一旦确定了，那么分区就不能再改变了 每个分区呢，装一个进程，也只能装一个进程，这就是固定分区的基本思想 那我们看看示例，假设我们把系统分成了这么四个分区 每个分区大小是不等的，那么来了一个进程，他可以根据自己对 内存的需求，排不同的队，比如说这两个进程排在分区一 如果分区一有其他进程占用，他们就先等着 这个进程可能在排队，在分区三这儿排队 那么还有几个进程在分区四排队 那么他的缺点呢，大家可以看到 比如说分区四，老被别的进程占用，那么这三个排队的就 得不到啊这个进入内存的机会，但其他的可能分区还是空闲的 因此呢，还有一种排队的方案 就是这样一种排队，像银行一样，我们排一个队 然后哪个分区空闲了，又能满足我的要求 我就进入哪个分区，这是比较 效率高一点的方案，当然 固定分区的方法呢，肯定现在也很少能见到，在系统里头 早起呢，我们很早的手机里头，最早的手机里头 内存空间有限，所以就是采用这样一个方法来管理空间的 下面我们介绍一下可变分区的存储管理方案 他的基本思想是，根据进程的需要 把内存的空闲分区呢分割出一个部分，然后分配给这个进程 这个进程需要多大，就分割它多大



空间 那么剩余的部分呢，就成为了一个新的空闲分区 我们看一个例子 假设进程p1进入内存p2进入内存p3进入内存，那么它要多大空间，就分配给它多大空间，最后 格局呢，就是有一个6兆的空闲分区留下了 预计一段时间之后，P2结束了，又进来一个新的进程，P4 那么它，P4只需要8兆空间，所以进程P2留下的这块儿分区呢 分给了8兆给进程P4，还剩下一个6兆的这么一个空闲区 又运行了一段时间，进程P1结束了，然后呢进程P5进来 那么它呢，用不了P1所占的所有内存空间，就把它 分割出一部分来之后，又剩下一个6兆的这个空闲区 那我们看到，经过了一段时间之后，内存的布局大致上是这样 一部分占用了，被进程占用，一部分呢就是空闲 那么这些空闲呢，我们来看，有这么几个空的空闲区，这几个空的空闲区呢 有的时候我们把它称之为一个洞，就好像在这个布局当中这几个地方是空的，就像空洞一样 这个是一个叫法，那当然这里头呢还有一个问题 什么问题呢，就是这些空闲区，这些洞比较小，那么这个小呢就是 成为了一个不好再分配的这么一个情况，出现了这么一种情况 太小了，每个进程都不能满足，分配不出去了 那么这个呢，我们把它称之为碎片 那么在这种情况下，这个碎片称为外碎片，也就是说 进程和进程之间，占据了一部分进程，在这个进程之间有一些 空闲区，这些碎片，称之为外碎片 如果外碎片很多的话，有很多很多小的空闲分区 那么就肯定会导致一个结果，内存的利用率就会下降 所以我们要解决碎片问题，下面我们来看一看碎片问题怎么解决 所谓碎片呢，是指的是在内存当中，有很多很小的 不容易利用的空闲区，碎片如果太多 那么进程要进入内存的需求呢，不能满足 但是系统中呢，又有很多的小的空闲区，这样的话，内存的利用率就很低 解决的方案呢，非常简单 我们称之为紧缩技术，所谓紧缩技术呢就是指的 在内存中把这些进程进行移动 把所有的空闲的小的分区呢，把它移动到内存的某一头 把它形成一个，合并成一个更大的空闲区，那这样的话呢 碎片问题，就把它合并起来了，其实也有不同的名词，比如说，压缩技术 然后紧致技术 搬家技术，其实都是指的内存紧缩，当然也可以说是一个内存的碎片的整理 那么在 做这项工作的时候，我们需要考虑几个问题 不是什么时候，不是所有的 进程都可以让你随便的搬家移动的 因此呢，我们要考虑两个问题，第一个问题呢，是开销 因为如果要把一个进程搬来搬去，在内存里头 搬一个进程还好一点，如果很多的进程都需要搬家 大家会想一下，会带来很多的时间，空间上的开销 当然另外一个问题，更重要，就是移动的时机 刚才我们已经说了，并不是所

有的在内存中的进程，让你随便移动的 比如说，一个进程，正在做IO，磁盘的IO操作 那这个时候这个进程，就不能够把它移到别处去，因为他会影响IO的结果 因此，在执行紧缩的时候 那你要考虑，第一开销是不是能够承受 第二呢，就是不是所有的进程都能够搬家的 至少在某些时刻，某些进程不能够搬家 这就是碎片问题的解决。