







固定分区分配

操作系统需要建立一个数据结构——分区说明表,来实现各个分区的分配与回收。每个表项对应一个分区,通常按分区大小排列。每个表项包括对应分区的大小、起始地址、状态(是否已分配)。

分区号	大小(MB)	起始地址(M)	状态
1	2	8	未分配
2	2	10	未分配
3	4	12	己分配

用数据结构 的数组(或 链表)即可 表示这个表

当某用户程序要装入内存时,由操作系统内核程序根据用户程序大小检索该表,从中找到一个能满足大小的、未分配的分区,将之分配给该程序,然后修改状态为"已分配"。

优点: 实现简单, 无外部碎片。

缺点: a. 当用户程序太大时,可能所有的分区都不能满足需求,此时不得不采用覆盖技术来解决,但这又会降低性能; b. 会产生内部碎片,内存利用率低。

系统区 (8MB)

分区1 (2MB)

/\ \(\sigma \, \tag{\cdots} \)

分区5 (8MB)

分区6 (12MB)

内存(分区大 小不等)

王道考研/CSKAOYAN.COM

动态分区分配

动态分区分配又称为可变分区分配。这种分配方式不会预先划分内存分区,而是在进程装入内存时,根据进程的大小动态地建立分区,并使分区的大小正好适合进程的需要。因此系统分区的大小和数目是可变的。(eg:假设某计算机内存大小为64MB,系统区8MB,用户区共56MB...)



- 1. 系统要用什么样的数据结构记录内存的使用情况?
- 2. 当很多个空闲分区都能满足需求时, 应该选择哪个分区进行分配?
- 3. 如何进行分区的分配与回收操作?

王道考研/CSKAOYAN.COM



















