

静态链表和动态链表区别详解 (无师自通)

前面学习了[链表](#)，其存储结构如[图 1](#) 所示：

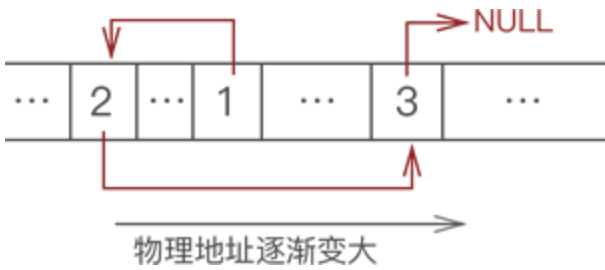


图 1 链表存储结构

类似图 1 这样的链表，它更喜欢人们称它为 "动态链表"。

随后又接触了[静态链表](#)。同样是存储图 1 中的数据 {1,2,3}，使用静态链表存储数据的状态如图 2 所示：

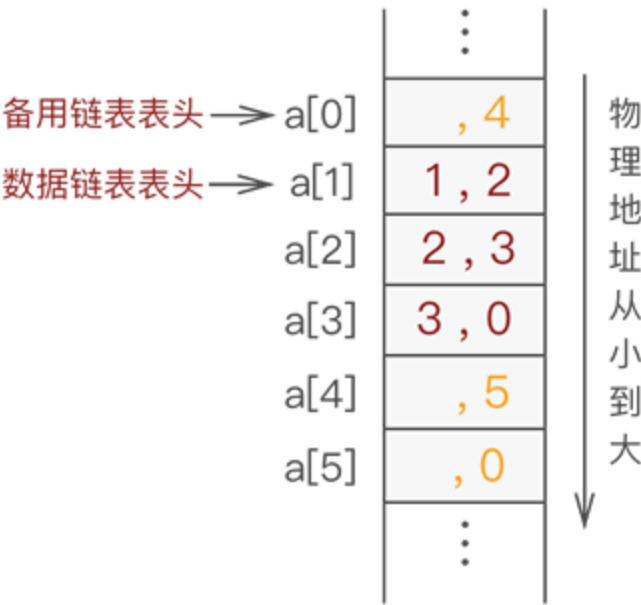


图 2 静态链表存储数据示意图

为了让读者分清动态链表和静态链表，本节来分析一下它们之间的区别和联系。

静态链表和动态链表的共同点是，数据之间"一对一"的逻辑关系都是依靠指针（静态链表中称"游标"）来维持，仅此而已。

静态链表

使用静态链表存储数据，需要预先申请足够大的一整块内存空间，也就是说，静态链表存储数据元素的个数从其创建的那一刻就已经确定，后期无法更改。

比如，如果创建静态链表时只申请存储 10 个数据元素的空间，那么在使用静态链表时，数据的存储个数就不能超过 10 个，否则程序就会发生错误。

不仅如此，静态链表是在固定大小的存储空间内随机存储各个数据元素，这就造成了静态链表中需要使用另一条链表（通常称为"备用链表"）来记录空间存储空间的位置，以便后期分配给新添加元素使用，如图 2 所示。

这意味着，如果你选择使用静态链表存储数据，你需要通过操控两条链表，一条是存储数据，另一条是记录空闲空间的位置。

动态链表

使用动态链表存储数据，不需要预先申请内存空间，而是在需要的时候才向内存申请。也就是说，动态链表存储数据元素的个数是无限的，想存多少就存多少。

同时，使用动态链表的整个过程，你也只需操控一条存储数据的链表。当表中添加或删除数据元素时，你只需要通过 malloc 或 free 函数来申请或释放空间即可，实现起来比较简单。

联系方式 购买教程（带答疑）