

牛客大学高薪加成系列课

链表



1. 链表 (linkedlist) 是一种有序的数据结构，且增删效率较高；
2. C语言没有内置这种数据结构，所以Redis构建了自己的链表实现。

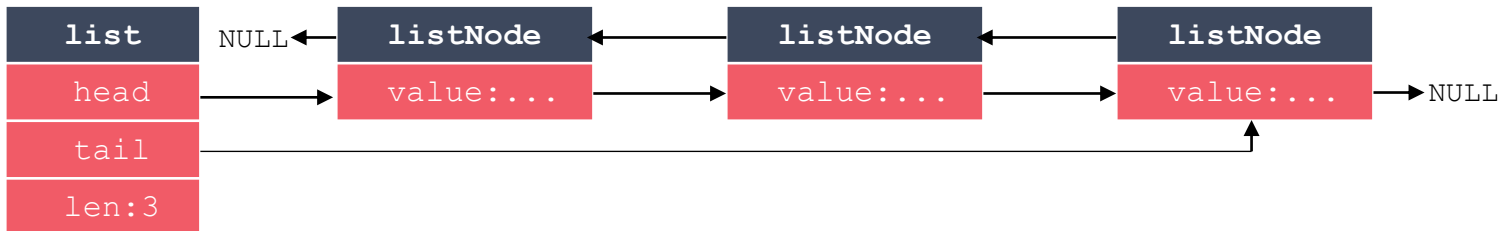
链表在Redis中应用十分广泛：

1. 链表是列表的底层实现方案之一；
2. 发布与订阅、慢查询、监视器等功能也用到了链表；
3. Redis服务器采用链表保存多个客户端的状态信息；
4. Redis客户端输出缓冲区是在链表的基础上实现的。

链表的实现主要涉及2个结构体：list、listNode，他们均位于adlist.h内。

```
typedef struct listNode {  
    // 前置节点  
    struct listNode *prev;  
    // 后置节点  
    struct listNode *next;  
    // 节点的值  
    void *value;  
} listNode;
```

```
typedef struct list {  
    listNode *head;    // 表头节点  
    listNode *tail;    // 表尾节点  
    void *(*dup)(void *ptr);    // 复制节点  
    void (*free)(void *ptr);    // 释放节点  
    int (*match)(void *ptr, void *key);    // 比较节点的值  
    unsigned long len;    // 节点数量  
} list;
```



■ 双端

链表节点带有prev指针和next指针，获取某个节点的前置和后置节点的复杂度为 $O(1)$ ；

■ 无环

表头节点的prev指针和表尾节点的next指针都指向NULL，对链表的访问以NULL为终点；

■ 多态

链表节点采用void*指针来保存节点的值，所以链表可以用于存储各种不同类型的值；

■ 有表头及表尾指针

通过list结构的head指针和tail指针，程序获取链表的表头和表尾节点的复杂度为 $O(1)$ ；

■ 有链表长度计数器

程序使用list结构的len属性对链表节点计数，所以获取链表中节点数量的复杂度为 $O(1)$ 。



牛客大学

- 专业求职辅导 -

THANKS



关注【牛客大学】公众号
回复“牛客大学”获取更多求职资料