

一、底层数据结构

▼ 简单动态字符串

▼ 由一个sdshdr结构表示，该结构中共有三个属性：

- len。记录数组中已使用的字节量，不包括结尾的空字符"\"。
- free。记录未使用的字节量。
- buf。字节数组，用来保存字符串。

▼ 特点：

- 常数复杂度获取字符串长度；
- 二进制安全；
- 杜绝缓冲区溢出；
- 兼容部分C字符串函数；
- 通过空间预分配和惰性空间释放减少修改字符串时带来的内存重分配次数；

▼ 链表

▼ 链表中的每一个节点由一个listNode结构表示，该结构共有三个属性：

- value。记录链表节点值；
- prev。指向上一个链表节点的指针；
- next。指向下一个链表节点的指针；

▼ 链表由一个list结构表示，该结构共有六个属性：

- head。指向链表表头节点的指针；
- tail。指向链表表尾节点的指正；
- len。记录链表节点数量；
- dup。节点值复制函数；
- free。节点值释放函数；
- mathch。节点值对比函数；

- 链表表头节点的前置指针和表尾结点后置指针均为NULL，即链表是一个无环双向链表。

▼ 字典

▼ 字典中的哈希表节点由一个dictEntry结构表示，该结构共有三个属性：

- key。哈希表节点key值。
- v。哈希表节点value值，可以是64bit整数，64bit的unsigned整数，指针。
- next指针。指向下一个哈希值相同的哈希表节点（字典使用链地址法解决哈希冲突）

▼ 字典中的哈希表由一个dictht结构表示，该结构共有四个属性：

- table。底层数组，存放哈希表节点；
- size。记录table数组大小；
- sizeMark。掩码，用于计算索引值，sizeMark恒等于size - 1；
- used。哈希表中节点数量；

▼ 字典由一个dict结构表示，该结构共有四个属性：

- dictht[2]。数组大小为2的哈希表；
- trehashidx。rehash索引；
- type。类型特定函数，是一个指向dictType结构的指针。dictType结构中包含了：计算哈希值的函数、销毁键的函数、销毁值的函数、复制键的函数、复制值的函数、对比键的函数；
- privdata。私有数据，保存了需要传递给类型特定函数的可选参数；

▼ 特点：

- 字典的rehash过程并不是一次性、集中式的完成，而是分多次、渐进式的完成。
- 在rehash过程中，新的键值对总是被添加到hashtable[1]中，hashtable[0]不在进行节点添加操作。
- rehash结束后，将hashtable[1]设置为hashtable[0]，然后为hashtable[1]重新分配一张空白哈希表。

▼ 哈希表扩展与收缩：

- 加载因子loadFactor = hashtable[0].used / hashtable[0].size
- loadFactor小于0.1会触发哈希表的收缩操作；
- 当redis服务器没有在执行BGSAVE命令或BGREWRITEAOF命令时，loadFactor大于等于1会触发哈希表的扩展操作；
- 当redis服务器正在执行BGSAVE命令或BGREWRITEAOF命令时，loadFactor大于等于5才会触发哈希表的扩展操作；
- redis服务器执行BGSAVE命令或BGREWRITEAOF命令时会创建当前服务器进程的子进程，而目前大多数操作系统基于写时复制技术优化子进程的执行效率，所以在子进程执行期间扩展哈希表会导致不必要的内存写入动作，因此应当尽可能的避免在子进程执行期间扩展哈希表。

▼ 跳跃表

▼ 跳跃表节点由一个zskiplistnode结构表示，该结构共有四个属性：

- obj。成员对象；
- score。成员分数；
- backward。后退指针；
- level数组。数组中每个元素包含两个属性：跨度span、前进指针forward；

▼ 跳跃表由一个zskiplist结构表示，该结构共有四个属性：

- head。指向跳跃表表头节点的指针；
- tail。指向跳跃表表尾结点的指针；
- length。跳跃表节点数量；
- level。跳跃表中层数最大的节点的层数；

▼ 特点：

- 跳跃表中所有节点的成员对象是唯一的，所有节点按照成员分数从小到大排序，成员分数相同的节点按照成员对象的字典序进行排序。
- 向跳跃表中添加节点时，redis服务器会按照幂等定律随机生成一个1~32之间的随机数作为新节点的层数，也就是该节点的level数组的大小。

▼ 整数集合

▼ 整数集合由一个intset结构表示，该结构共有三个属性：

- encoding。编码方式，可以保存int16，int32，int64的整数；
- length。整数集合的元素数量；
- contents。底层数组，保存整数集合中的所有元素；

▼ 特点：

- 整数集合中的所有元素均为整数，且是唯一的，按照从小到大的顺序排序；
- 向整数集合中添加一个超出encoding编码范围支持的数据时，会触发整数集合的升级操作。整数集合支持升级操作，但是不支持降级操作。

▼ 压缩列表

▼ 压缩列表是由一系列特殊编码的连续内存块组成的特殊数据结构。完整的压缩列表由五部分组成：

- zlbytes。记录压缩列表占用的字节总数；
- zltail。记录压缩列表表尾节点的偏移量；
- zllen。记录压缩列表中的节点数量；
- entryX。记录压缩列表中的各个节点信息；
- zlend。压缩列表的末端。

▼ 压缩列表的每个每个节点由三部分组成：

- previous\_entry。上一个节点占用的字节总数。结合zltail属性，压缩列表组成了一条从后向前的单向链表；
- encoding。保存了该节点的数据类型及数据长度；
- content。该节点的具体值；