## 二、redis对象

- **redis服务器是一个键值对数据库，但是并没有直接使用基本数据结构实现键值对数据库，而是创建了一个对象系统，系统中包含字符串对象、列表对象、哈希对象、集合对象、有序集合对象。**

### 对象对应redisObject结构

- encoding属性表明对象的编码方式；
- ptr是一个指向底层实现数据结构的指针；

### 对象系统

#### 字符串对象

##### 1.int编码方式

- 如果字符串对象保存的是一个整数值，并且可以用long类型表示，将使用int编码方式；

##### 2.raw编码方式

- 如果字符串对象保存的是一个字符串值，并且长度大于等于40字节，使用raw编码方式；

##### 3.emb_str编码方式

- 如果字符串对象保存的是一个字符串值，并且长度小于40字节，使用emb_str编码方式；

##### raw与emb_str差异

- raw字符串对象调用两次内存分配函数申请空间依次存放redisObject结构和sdshdr结构； 相应调用两次内存释放函数释放所占用的内存空间；
- emb_str字符串对象只需要调用一次内存分配函数申请一片连续的内存空间，依次存放redisObject结构和sdshdr结构，相应也只需要调用一次内存释放函数；
- emb_str字符串对象的redisObject结构和sdshdr结构在内存上是相邻的，因此能够更好地利用缓存带来的优势；

#### 列表对象

##### 1.ziplist编码方式

- 使用压缩列表作为底层实现，压缩列表中每个节点保存了列表对象中某个元素的值。

##### 2.linkedlist编码方式

- 使用双向链表作为底层实现，链表中的每个节点都是一个字符串对象，字符串对象中保存了列表对象某个元素的值；

##### 在满足以下两个条件时使用ziplist编码方式，否则升级为linkedlist编码方式：

- 列表对象中每个元素的长度均小于64字节；
- 列表对象中的元素数量小于512个；

#### 哈希对象

##### 1.ziplist编码方式

- 使用压缩列表作为底层实现，列表中的所有节点成对出现。每对结点中的第一个节点表示哈希对象某个键值对的key值，第二个节点表示某个键值对的value值；
- 新的键值对总是被添加到压缩列表的表尾；

##### 2.hashtable编码方式

- 使用字典作为底层实现，字典中的每一个键都是一个字符串对象，用来保存哈希对象中某个键值对的key值，每一个值也是一个字符串独享，用来保存某个键值对的value值；

##### 在满足以下两个条件时使用ziplist编码方式，否则升级为hashtable编码方式：

- 哈希对象中每个键值对的key值、value值长度均小于64字节；
- 哈希对象中的键值对数量小于512个；

#### 集合对象

##### 1.intset编码方式

- 使用整数集合作为底层实现；

##### 2.hashtable编码方式

- 使用字典作为底层实现。字典中的每一个键都是一个字符串对象，用来保存集合对象中某个元素的值，每一个值都被设置为NULL；

##### 在满足以下两个条件时使用intset编码方式，否则升级为hashtable编码方式：

- 集合对象中的所有元素均为整数值，且不重复；
- 集合对象中的元素数量小于512个；

#### 有序集合对象

##### 1.ziplist编码方式

- 使用压缩列表作为底层实现，列表中所有节点成对出现，每对结点中第一个节点表示有序集合对象中某个元素的成员对象，第二个节点表示某个元素的成员分数；

##### 2.skiplist编码方式

- 使用zset结构作为底层实现，该结构中同时包含了字典dict结构和zskiplist结构。
- 通过dict结构构建了有序集合对象中所有元素从成员对象到成员分数的映射；
- 通过zskiplist结构将有序集合对象中所有元素保存起来；
- 虽然同时使用了dict结构和zskiplist结构，但通过指针共享相同元素的成员对象和成员分数，没有浪费额外的内存空间；

##### 在满足以下两个条件时使用ziplist编码方式，否则升级为skiplist编码方式：

- 有序集合对象中所有元素的成员对象均小于64字节；
- 有序集合对象中的元素数量小于128个；

### 对象回收

- redis基于C语言实现，而C语言并没有提供内存回收机制。因此redis基于对象引用计数算法实现了内存回收，通过统计对象的引用计数信息，回收对象并释放他们所占用的存储空间；

### 对象共享

- redis基于对象的引用计数信息实现对象共享功能。对象对应redisObject结构中的refCount属性用于记录该对象被引用的次数；
- redis服务器在启动时会初始化10000个字符串对象，包含了0~9999的所有值，之后使用其中的数据时直接进行共享即可，而不需要额外创建对象；

#### redis服务器共享对象：

- 让一个数据库键的指针指向某个已有的值对象；
- 将值对象的引用计数信息+1；

### 对象空转时长lru

- 对象对应redisObject结构中的lru属性用于记录对象最后一次被访问的时间；
- 用redis服务器状态redisServer结构中的lruLock属性减去某个对象lru，即可得出该对象的空转时长；
- 如果redis服务器开启了maxMemory选项，并且使用volatile-lru或allkeys-lru算法，当redis服务器内存占用超出maxMemory后，会优先将空转时长较长的对象进行回收，并释放他们所占用的内存空间；