# 一二 数据操作

## 0 开始使用

启动服务器：redis-server 按ctrl+c停止

启动客户端：redis-cli 在新终端中运行如下代码

ping

set 'a' '123'

当添加键值后，发现在当前运行的目录下，创建了一个文件：dump.rdb，这个文件用于将数据持久化存储

## 1 数据类型

redis是key-value的数据，所以每个数据都是一个键值对，键的类型是字符串，值的类型有五种：字符串string、哈希hash、列表list、无序集合set、有序集合zset。

## 2 string与常见命令

string是redis最基本的类型，最大能存储512MB数据，string类型是二进制安全的，即可以为任何数据，比如数字、图片、序列化对象等。

设置：

set key value 设置键值

SETEX key seconds value 设置键值及过期时间，以秒为单位（如存储session）

MSET key value [key value ...] 设置多个键值

获取：

GET key 根据键获取值，如果不存在此键则返回nil

MGET key1 key2 ... 根据多个键获取多个值

运算：值是数字 虽然用的是tring类型，但是如果是数字可以进行运算

INCR key 将key对应的value加1

INCRBY key increment 将key对应的value加整数

DECR key 将key对应的value减1

DECRBY key decrement 将key对应的value减整数

其他：

APPEND key value 追加值

STRLEN key 获取值长度

## 3 键的命令

KEYS pattern 查找键，参数支持正则

EXISTS key key ... 判断键是否存在，如果存在返回1，不存在返回0

TYPE key 查看键对应的value的类型

DEL key key ... 删除键及对应的值

EXPIRE key seconds 设置过期时间（秒），没有则一直在，直到DEL移除

TTL key 查看有效时间

## 4 hash 存储对象属性的键值对

设置：

HSET key field value 设置单个属性：hset person name 'lisi'

HMSET key field value field value ... 设置多个属性

获取：

HGET key field 获取一个属性的值

HMGET key field field ... 获取多个属性的值

HGETALL key 获取所有属性和值

HKEYS key 获取所有的属性

HLEN key 返回包含属性的个数

HVALS key 获取所有值

其它:

HEXISTS key field 判断属性是否存在

HDEL key field field ... 删除属性及值

HSTRLEN key field 返回值的字符串长度

## 5 list存储数组

列表的元素类型为string，按照插入顺序排序，在列表的头部或者尾部添加元素

设置：

LPUSH key value value ... 在头部插入数据

RPUSH key value value ... 在尾部插入数据

LINSERT key BEFORE|AFTER pivot value在一个元素的前|后插入新元素

LSET key index value 设置指定索引的元素值，索引是基于0的下标，可以是 负数，表示偏移量从list尾部开始计数，如-1表示列表 的最后一个元素

获取：

LPOP key 移除并且返回 key 对应的 list 的第一个元素

RPOP key 移除并返回存于 key 的 list 的最后一个元素

LRANGE key start stop 返回存储在 key 的列表里指定范围内的元素

start 和 end 偏移量都是基于0的下标，偏移量也可以 是负数，表示偏移 量是从list 尾部开始计 数， 如-1表示列表的最后一个元素

其它：

LTRIM key start stop 裁剪列表，改为原集合的一个子集

start 和 end 偏移量都是基于0的下标

偏移量也可以是负数，表示偏移量是从list尾部开始计 数，如-1表示列表的最后一个元素

LLEN key 返回存储在 key 里的list的长度

LINDEX key index 返回列表里索引对应的元素

## 6 set存储无序集合

元素为string类型，元素具有唯一性，不重复

设置：

SADD key member member ... 添加元素

获取：

SMEMBERS key 返回key集合所有的元素

SCARD key 返回集合元素个数

其它：

SINTER key key ... 求多个集合的交集

SDIFF key key ... 求某集合与其它集合的差集

SUNION key key ... 求多个集合的合集

SISMEMBER key member 判断元素是否在集合中

## 7 zset存储有序集合

元素为string类型，元素具有唯一性，不重复。

每个元素都会关联一个double类型的score，表示权重，通过权重将元素从小到大排序，

元素的score可以相同。

ZADD key score member score member ... 添加

ZRANGE key start stop 返回指定范围内的元素

ZCARD key 返回元素个数

ZCOUNT key min max 返回有序集key中，score在min和max之间的成员

ZSCORE key member 返回有序集key中，成员member的score值

## 8 list set zset区别

list：有顺序的集合，元素不能重复，存储的集合按照我们自己添加的顺序排序；

set：没顺序的集合，元素可以重复，不依赖于添加、大小顺序去，且值可以重复；

zset：有顺序的集合，元素不能重复，根据权重大小从小到大排列；

# 二 Redis深入

## 1 发布订阅

发布订阅其实是一种设计模式（如单例模式）：

发布者不是计划发送消息给特定的接收者（订阅者），而是发布的消息分到不同的频道，不需要知道什么样的订阅者订阅；

订阅者对一个或多个频道感兴趣，只需接收感兴趣的消息，不需要知道什么样的发布者发布的；

发布者和订阅者的解耦合可以带来更大的扩展性和更加动态的网络拓扑；

客户端发到频道的消息，将会被推送到所有订阅此频道的客户端；

客户端不需要主动去获取消息，只需要订阅频道，这个频道的内容就会被推送过来

消息的格式。

推送消息的格式包含三部分：

part1:消息类型，包含三种类型

subscribe，表示订阅成功

unsubscribe，表示取消订阅成功

message，表示其它终端发布消息

如果第一部分值为subscribe，则第二部分是频道，第三部分是现在订阅的频道的数量；

如果第一部分值为unsubscribe，则第二部分是频道，第三部分是现在订阅的频道的数量，如果为0则表示当前没有订阅任何频道，当在Pub/Sub以外状态，客户端可以发出任何redis命令；

如果第一部分的值为message，则第二部分是来源频道的名称，第三部分是消息的内容；

SUBSCRIBE 频道名称 频道名称 ... 订阅

UNSUBSCRIBE 频道名称 频道名称 ... 取消订阅，如果不写参数，表示取消所有订阅

PUBLISH 频道 消息 发布

## 2 主从配置

一个master可以拥有多个slave，一个slave又可以拥有多个slave，如此下去，形成了强大的多级服务器集群架构，比如，将ip为192.168.1.10的机器作为主服务器，将ip为192.168.1.11的机器作为从服务器

bind 192.168.1.10 设置主服务器的配置

bind 192.168.1.11

slaveof 192.168.1.10 6379 设置从服务器的配置，在slaveof后面写主机ip， 再写端口，而且端口必须写

在master和slave分别执行info命令，查看输出信息

set hello world 在master上写数据

get hello 在slave上读数据