Redis提供了丰富的命令（command）对数据库和各种数据类型进行操作，这些command可以在Linux终端使用。在编程时，比如使用Redis 的Java语言包，这些命令都有对应的方法。下面将Redis提供的命令做一总结。

官网命令列表：http://redis.io/commands

**启动 Redis 服务**src/redis-server或者src/redis-server redis.conf

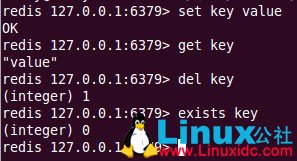
src/redis-server redis.conf 1>log.log 2>errlog.log

1为标准输出，2为错误输出

**将 Redis 作为 Linux 服务随机启动**  
vi /etc/rc.local, 加入代码：

/root/install/redis-2.4.9/src/redis-server

**客户端连接**  
src/redis-cli  
  
**停止redis服务：**  
src/redis-cli shutdown

**增删改查：**  
  


**keys \***  
取出当前匹配的所有key

**> exists larry**  
(integer) 0

当前的key是否存在

**del lv**  
删除当前key

**expire**  
设置过期时间

> expire larry 10  
(integer) 1

**> move larry ad4**  
(integer) 1

移动larry键值对到ad4数据库

**> persist lv**  
(integer) 1  
移除当前key的过期时间

**randomkey**

随机返回一个key

**rename**  
重命名key

**type**  
返回值的数据类型

 type testlist  
list

> **ping**  
PONG  
  
测试连接是否还在

**>echo name**

"larry"

打印

> **select ad4databank**  
OK  
数据库切换

**> quit**  
退出连接

**> dbsize**  
(integer) 12

当前数据库中key的数量

**> info**  
服务器基本信息

**monitor**

实时转储收到的请求

**config get**  
获取服务器的参数配置

**flushdb**  
清空当前数据库

**flushall**  
清除所有数据库

1、连接操作相关的命令

quit：关闭连接（connection）

auth：简单密码认证

2、对value操作的命令

exists(key)：确认一个key是否存在

del(key)：删除一个key

type(key)：返回值的类型

keys(pattern)：返回满足给定pattern的所有key

randomkey：随机返回key空间的一个key

rename(oldname, newname)：将key由oldname重命名为newname，若newname存在则删除newname表示的key

dbsize：返回当前数据库中key的数目

expire：设定一个key的活动时间（s）

ttl：获得一个key的活动时间

select(index)：按索引查询

move(key, dbindex)：将当前数据库中的key转移到有dbindex索引的数据库

flushdb：删除当前选择数据库中的所有key

flushall：删除所有数据库中的所有key

3、对String操作的命令

set(key, value)：给数据库中名称为key的string赋予值value

get(key)：返回数据库中名称为key的string的value

getset(key, value)：给名称为key的string赋予上一次的value

mget(key1, key2,…, key N)：返回库中多个string（它们的名称为key1，key2…）的value

setnx(key, value)：如果不存在名称为key的string，则向库中添加string，名称为key，值为value

setex(key, time, value)：向库中添加string（名称为key，值为value）同时，设定过期时间time

mset(key1, value1, key2, value2,…key N, value N)：同时给多个string赋值，名称为key i的string赋值value i

msetnx(key1, value1, key2, value2,…key N, value N)：如果所有名称为key i的string都不存在，则向库中添加string，名称key i赋值为value i

incr(key)：名称为key的string增1操作

incrby(key, integer)：名称为key的string增加integer

decr(key)：名称为key的string减1操作

decrby(key, integer)：名称为key的string减少integer

append(key, value)：名称为key的string的值附加value

substr(key, start, end)：返回名称为key的string的value的子串

4、对List操作的命令

rpush(key, value)：在名称为key的list尾添加一个值为value的元素

lpush(key, value)：在名称为key的list头添加一个值为value的 元素

llen(key)：返回名称为key的list的长度

lrange(key, start, end)：返回名称为key的list中start至end之间的元素（下标从0开始，下同）

ltrim(key, start, end)：截取名称为key的list，保留start至end之间的元素

lindex(key, index)：返回名称为key的list中index位置的元素

lset(key, index, value)：给名称为key的list中index位置的元素赋值为value

lrem(key, count, value)：删除count个名称为key的list中值为value的元素。count为0，删除所有值为value的元素，count>0从 头至尾删除count个值为value的元素，count<0从尾到头删除|count|个值为value的元素。 lpop(key)：返回并删除名称为key的list中的首元素 rpop(key)：返回并删除名称为key的list中的尾元素 blpop(key1, key2,… key N, timeout)：lpop命令的block版本。即当timeout为0时，若遇到名称为key i的list不存在或该list为空，则命令结束。如果timeout>0，则遇到上述情况时，等待timeout秒，如果问题没有解决，则对 keyi+1开始的list执行pop操作。

brpop(key1, key2,… key N, timeout)：rpop的block版本。参考上一命令。

rpoplpush(srckey, dstkey)：返回并删除名称为srckey的list的尾元素，并将该元素添加到名称为dstkey的list的头部

5、对Set操作的命令

sadd(key, member)：向名称为key的set中添加元素member

srem(key, member) ：删除名称为key的set中的元素member

spop(key) ：随机返回并删除名称为key的set中一个元素

smove(srckey, dstkey, member) ：将member元素从名称为srckey的集合移到名称为dstkey的集合

scard(key) ：返回名称为key的set的基数

sismember(key, member) ：测试member是否是名称为key的set的元素

sinter(key1, key2,…key N) ：求交集

sinterstore(dstkey, key1, key2,…key N) ：求交集并将交集保存到dstkey的集合

sunion(key1, key2,…key N) ：求并集

sunionstore(dstkey, key1, key2,…key N) ：求并集并将并集保存到dstkey的集合

sdiff(key1, key2,…key N) ：求差集

sdiffstore(dstkey, key1, key2,…key N) ：求差集并将差集保存到dstkey的集合

smembers(key) ：返回名称为key的set的所有元素

srandmember(key) ：随机返回名称为key的set的一个元素

6、对zset（sorted set）操作的命令

zadd(key, score, member)：向名称为key的zset中添加元素member，score用于排序。如果该元素已经存在，则根据score更新该元素的顺序。

zrem(key, member) ：删除名称为key的zset中的元素member

zincrby(key, increment, member) ：如果在名称为key的zset中已经存在元素member，则该元素的score增加increment；否则向集合中添加该元素，其score的值为increment

zrank(key, member) ：返回名称为key的zset（元素已按score从小到大排序）中member元素的rank（即index，从0开始），若没有member元素，返回“nil”

zrevrank(key, member) ：返回名称为key的zset（元素已按score从大到小排序）中member元素的rank（即index，从0开始），若没有member元素，返回“nil”

zrange(key, start, end)：返回名称为key的zset（元素已按score从小到大排序）中的index从start到end的所有元素

zrevrange(key, start, end)：返回名称为key的zset（元素已按score从大到小排序）中的index从start到end的所有元素

zrangebyscore(key, min, max)：返回名称为key的zset中score >= min且score <= max的所有元素 zcard(key)：返回名称为key的zset的基数 zscore(key, element)：返回名称为key的zset中元素element的score zremrangebyrank(key, min, max)：删除名称为key的zset中rank >= min且rank <= max的所有元素 zremrangebyscore(key, min, max) ：删除名称为key的zset中score >= min且score <= max的所有元素

zunionstore / zinterstore(dstkeyN, key1,…,keyN, WEIGHTS w1,…wN, AGGREGATE SUM|MIN|MAX)：对N个zset求并集和交集，并将最后的集合保存在dstkeyN中。对于集合中每一个元素的score，在进行 AGGREGATE运算前，都要乘以对于的WEIGHT参数。如果没有提供WEIGHT，默认为1。默认的AGGREGATE是SUM，即结果集合中元素 的score是所有集合对应元素进行SUM运算的值，而MIN和MAX是指，结果集合中元素的score是所有集合对应元素中最小值和最大值。

7、对Hash操作的命令

hset(key, field, value)：向名称为key的hash中添加元素field<—>value

hget(key, field)：返回名称为key的hash中field对应的value

hmget(key, field1, …,field N)：返回名称为key的hash中field i对应的value

hmset(key, field1, value1,…,field N, value N)：向名称为key的hash中添加元素field i<—>value i

hincrby(key, field, integer)：将名称为key的hash中field的value增加integer

hexists(key, field)：名称为key的hash中是否存在键为field的域

hdel(key, field)：删除名称为key的hash中键为field的域

hlen(key)：返回名称为key的hash中元素个数

hkeys(key)：返回名称为key的hash中所有键

hvals(key)：返回名称为key的hash中所有键对应的value

hgetall(key)：返回名称为key的hash中所有的键（field）及其对应的value

8、持久化

save：将数据同步保存到磁盘

bgsave：将数据异步保存到磁盘

lastsave：返回上次成功将数据保存到磁盘的Unix时戳

shundown：将数据同步保存到磁盘，然后关闭服务

9、远程服务控制

info：提供服务器的信息和统计

monitor：实时转储收到的请求

slaveof：改变复制策略设置

config：在运行时配置Redis服务器