一、Redis控制台相关操作

1.安装完后进入redis

（1）启动Redis服务：**redis -server /myredis/redis.conf**

（2）查看Redis服务有没有启动：**ps -ef|grep redis**

（3）进入redis客户端：**redis-cli -p 6379**

2.Redis的库

从配置文件中可以看出，默认有16个库，分别为0-15

用命令 ： **select 7** 来选择八号库

3.有关于key的命令

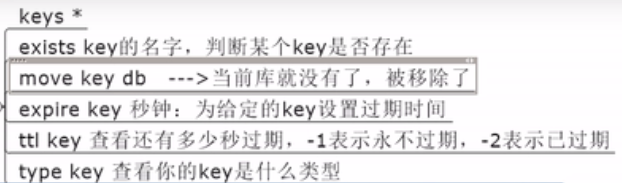
（1）查看当前数据库有多少个key： **dbsize**

（2）查看当前数据库所有key都是什么：**key s \*** # \*代表所有，?代表占一个

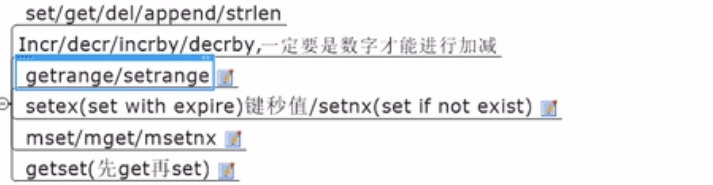
（3）删除当前库的所有key：**flushdb** 删除所有库的key：**flushall**

（4）移动当前库的key到其他库： **move key1 3** #将key1移动到3号库

（5）其他



4.有关string类型的命令



append：向字符串末尾添加新字符串

strlen：查询当前字符串长度

incrby：例如：incrby key1 2 #key1这个字符串的值依次增加2

getrange：例如：getrange key1 0 3 #获取key1字符串0-3位的自符

例如：getrange key1 0 -1 #获取key1字符串所有自符

setrange：例如：setrange key1 0 \*\*\* #原key1的值为“12345”，现为“\*\*\*45”**会覆盖**

setex：设置key时，定义其生效时间

setnx：设置key时，如果存在就不覆盖；如果不存在才会set进数据

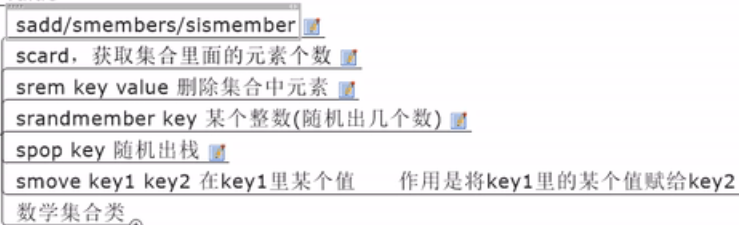
mset：批量set，例如：set key1 v1 key2 v2 key3 v3

5.有关list的命令



lrem：例如 lrem list1 2 3 则原list1为“1,1,2,2,3,3,4,4”，现为“1,1,2,2,4,4”（从左边开始删除两个3）

6.有关set的命令





smembers set1 查看set1中所有数据

sismember set1 X 查看X是否在set1中

sdiff：例如 sdiff set1 set2 set1中数据为"1,2,3,4,5"，set2中数据为"1,2,3,a,b" 则输出"4","5"，即输出set2中没有set1的数据

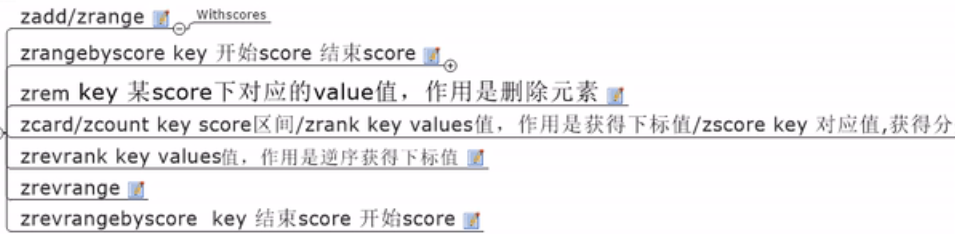
7.有关hash的命令



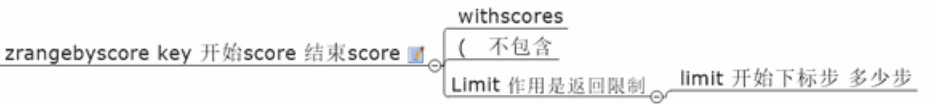
hkeys：获取所有的key

hvals：获取所有的value

8.有关zset的命令



zrange zset 0 -1 withscores 带上分数一起输出



zrangebyscore zset1 (60 (90 limit 2 2 查询分数在60-90的数据，从2这个位置开始截取2个

二、Redis的配置文件（redis.conf）

1.includes 可以包含其他配置文件

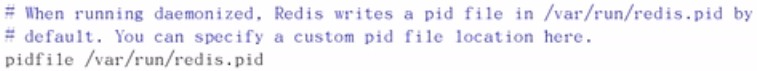


2.general

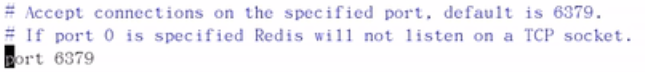
（1）守护进程



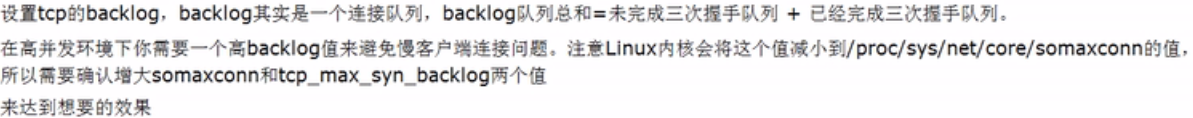
（2）配置pid文件路径



（3）配置端口号



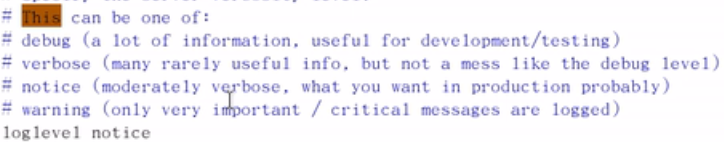
（4）tcp-backlog 511



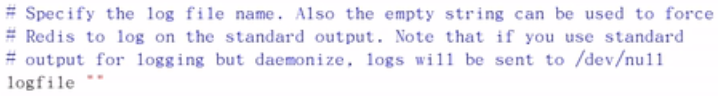
（5）bind

（6）tcp-keepalive 60

（7）日志级别



（8）日志名称



（9）syslog-enabled no 是否在系统日志中生成

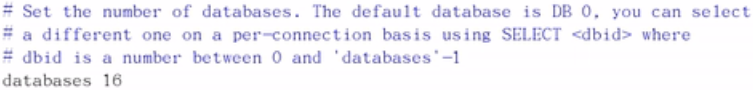


（10）syslog-ident redis 在系统日志中用“redis”标识日志身份

（11）syslog-facility local0 在系统日志中标识redis服务名称



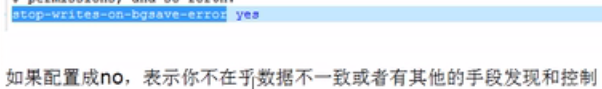
（12）设置redis服务中有多少个数据库



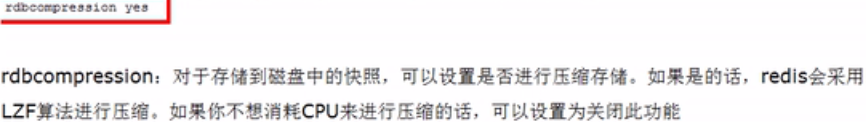
3.snapshotting快照

（1）save <second> <count> #在second时间内至少变更count次，就执行save

（2）stop-writes-on-bgsave-error



（3）rdbcompression



（4）rdbchecksum

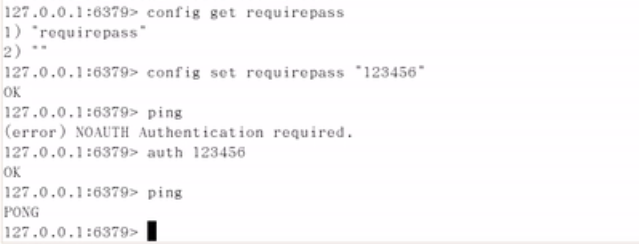


（5）dbfilename

（6）dir

4.replication复制

5.security 安全



获取redis的密码： config get requirepass

设置redis的密码： config set requirepass "123456"

在之后打出任何命令前，必须加上auth 123456。例如：auth 123456 ping

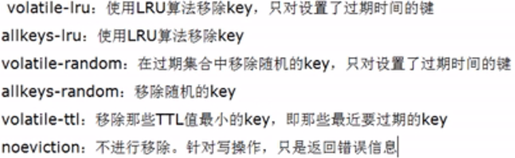
6.limits限制

（1）maxclients

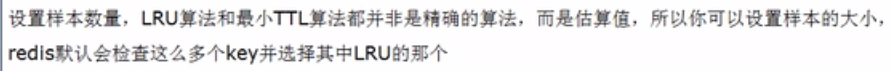
（2）maxmemory

（3）maxmemory-policy

lru：least recently used ttl：time to live



（4）maxmemory-samples



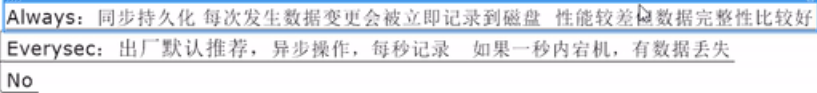
8.append only mode 新追加的AOF模块

（1）appendonly yes为启用

（2）appendfilename

（3）appendfsync

三种持久化方式



（4）no-appendfsync-on-rewrite 重写时是否异步，设为no可以保证数据安全性

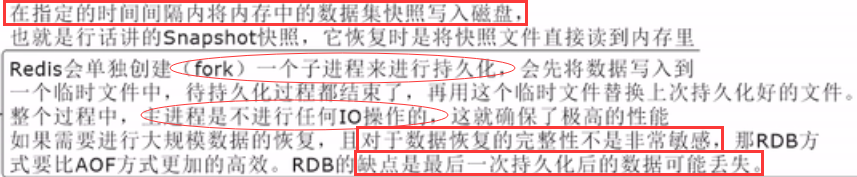
（5）auto-aof-rewrite-min-size 64mb aof文件的大小超过64mb会rewrite

（6）auto-aof-rewrite-percentage 100 #这里表示100%，即是上一次rewrite的一倍

三、Redis持久化

（1）RDB redis database

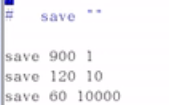
1）是什么



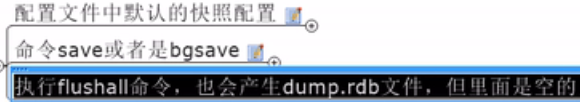
2）rdb的配置与触发

配置文件中配置：save <second> <count> #在second时间内至少变更count次，就执行save

不采用任何策略 就用save ""



save 全阻塞，执行时只管存 bgsave异步保存



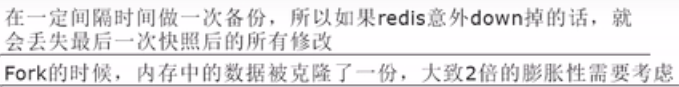
3）数据恢复



4）优势

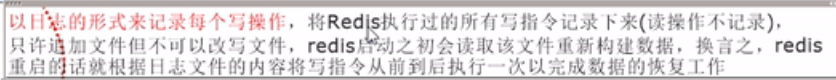


5）劣势



6）动态停止rdb持久化（使用命令停止）redis-cli config set save ""

（2）AOF （append of file）



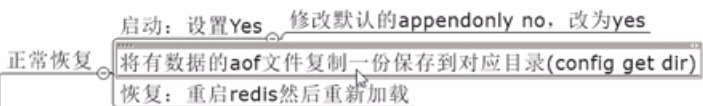
aof可以和rdb共存，先读取的是aof，aof文件修复命令 redis-check-aof --fix appendonly.aof

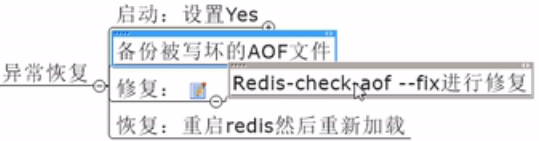
1）配置



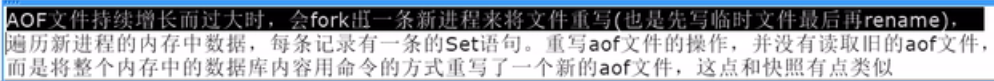


2）启动/修复/恢复





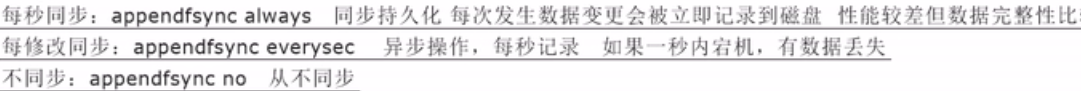
3）rewrite



触发机制：



4）优势



5）劣势



四、Redis事务

1.是什么



2.命令

（1）multi 启动事务

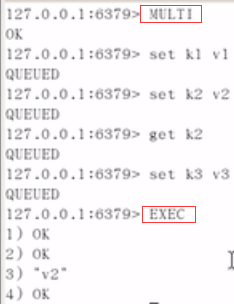
（2）exec 提交事务

（3）discard 放弃事务

（4）upwatch 放弃对所有key的监控

（5）watch key[key1,key2] 对key进行监控

例子：



3.redis对事务部分支持

一串命令中有 inner key1时，如果key1的值不是整数，则该条命令不会执行，其他正常执行。所以redis对事务部分支持

4.watch监控

（1）悲观锁、乐观锁、CAS（check and set）

悲观锁，在执行操作前先上锁

乐观锁，在执行时不上锁，但是提交前会检查这条数据是否正常，利用版本号机制（提交的 版本必须大于当前的版本才能执行更新），watch命令就类似与乐观锁

（2）加上watch后，如果其他进行修改了数据，则提交数据就会报错（执行的事务都会放弃，同时返回nullmulti-bulk应答以通知调用者事务执行失败）

（3）unwatch取消所监控

（4）在 exec后，所有watch都会清空

五、Redis发布订阅机制

1.命令



六、Redis的主从复制

1.主库负责写，从库负责读

2.从库的配置 slaveof 127.0.0.0 5151

（1）查看redis的信息 info replication

（2）使从库复制主库：slaveof 主机ip 主机端口号

slave与master端口后，需要重新使用slaveof命令连接，除非配置进redis.conf文件

（3）当mater死后，其他slave等待mater重新连接；其中某个slave也可以变为master（使用命令：slaveof no one）

3.修改配置文件

（1）拷贝多个redis.conf文件

（2）开启daemonize yes



（3）pid文件名字



（4）指定端口



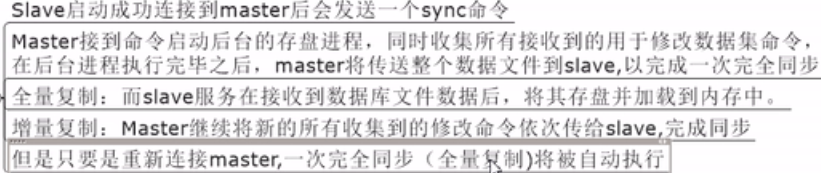
（5）log文件名字



（6）dump.rdb名字



4.复制的原理



5.哨兵模式

**哨兵模式配置：**

（1）新建sentinel.conf文件

（2）在这个配置文件中配置：



host6379：被监控数据库的名字（自己起的名字）

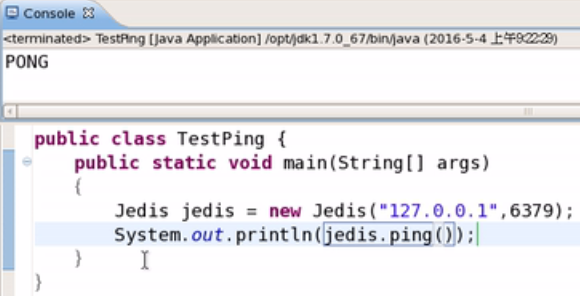
1：投票时超过1票才能被选为master

（3）启动哨兵 redis-sentinel /配置文件的路径/sentinel.conf

缺点：选举的有点慢

七、Jedis

1.测试redis是否连通



2.事务

Jedis jedis = new Jedis("127.0.0.1",6379);

jedis.watch();

Transaction tx = jedis.multi();

tx..set("key1","v1");

tx..set("key1","v1");

ts.exec();

3.主从复制

Jedis jedis\_M = new Jedis("127.0.0.1",6379);

Jedis jedis\_S = new Jedis("127.0.0.1",6380);

jedis\_S.slaveof("127.0.0.1",6379)

jedis\_M.set("key1","v1");

Thread.sleep(1000); //给从库同步数据争取点时间

String result = jedis\_S.get("key1");

System.out.println(result);

4JedisPool

public class JedisPoolUtil{

private static volatile JedisPool jedisPool = null;

private JedisPoolUtil(){}

public static JedisPool getJedisPoolInstance(){

if(JedisPool == null){

synchronized(jedisPool.class){

if(JedisPool == null){

JedisPoolConfig jedisPoolConfig = new JedisPoolConfig();

jedisPoolConfig.setMaxActive(1000);

jedisPoolConfig.setMaxIdle (32);//最大空闲数

jedisPoolConfig.setMaxWait(100\*1000);

jedisPoolConfig.setTestOnBorrow(true);//获得一个jedis实例时是否检查连接可用性(ping())

jedisPool = new JedisPool(jedisPoolConfig,"127.0.0.1",6379);

}

}

}

}

public static void release(JedisPool jedisPool, Jedis jedis){

if(jedis != null){

jedisPool.returnResourceObject(jedis);

}

}

}

**test:**

