

nosql

<https://baike.baidu.com/item/NoSQL/8828247?fr=aladdin>

<https://blog.csdn.net/wyz0516071128/article/details/80877984>

1, 准备工作

先要安装gcc-c++

```
yum install -y gcc-c++
```

2, 下载redis

中文官网: <http://www.redis.cn/>

linux下载地址: wget [http://download.redis.io/releases/redis-](http://download.redis.io/releases/redis-4.0.1.tar.gz)

[4.0.1.tar.gz](http://download.redis.io/releases/redis-4.0.1.tar.gz)

-c 断点续传 -b后台下载

```
tar -zxvf redis-4.0.1.tar.gz -C 安装位置
```

3, 安装redis

编译: make

安装: make install PREFIX=指定安装目录

通常安装在usr下 :

```
make install PREFIX=/usr/redis
```

4, 启动

方法1: 进入bin目录 ./redis-server (进入安装目录 下)

方法2: 复制安装包中 redis.conf 到安装目录bin下 修改里面守护进程配置为

daemonize yes

(方法2启动方式可以redis启动, 是后台运行, 无需新开窗口就可以测

试)

服务器端启动: ./redis-server redis.conf

查看是否启动: ps aux|grep redis

客户端连接, 仍然是bin下的执行: ./redis-cli 回车

使用ping 命令, 如果得到 pong说明可以用

在客户端使用该命令退出: shutdown NOSAVE

exit

5, 数据类型

查看密码: config get requirepass

设置密码: `config set requirepass "admin"`

如果设置了密码, 登录时需要验证: `auth admin`

测试服务是否能连通: `ping` 如果正常返回pong

<http://www.redis.net.cn/order/>

<http://blog.csdn.net/hechurui/article/details/49508735>

`flushall` 清空所有数据

字符串 `string`(字符串是Redis中最基本的数据类型, 它能够存储任何类型的字符串)

(`set` `get` `del` `mset` `mget` `incr` `decr` `append`)

```
[root@serverone bin]# ./redis-server redis.conf
2658:C 08 Apr 11:44:08.952 # oO0OoO0OoO0Oo Redis is starting oO0OoO0OoO0Oo
2658:C 08 Apr 11:44:08.952 # Redis version=4.0.1, bits=64, commit=00000000, modified=0, pid=2658, just started
2658:C 08 Apr 11:44:08.952 # Configuration loaded
[root@serverone bin]# ./redis-cli
127.0.0.1:6379> set aaa 111
OK
127.0.0.1:6379> get aaa
"111"
127.0.0.1:6379> incr aaa
(integer) 112
127.0.0.1:6379> decr aaa
(integer) 111
127.0.0.1:6379> mset bbb 222 ccc 333 ddd 444
OK
127.0.0.1:6379> mget aaa bbb ccc ddd
1) "111"
2) "222"
3) "333"
4) "444"
127.0.0.1:6379> del ddd
(integer) 1
127.0.0.1:6379> mget aaa bbb ccc ddd
1) "111"
2) "222"
3) "333"
4) (nil)
127.0.0.1:6379> append aaa aaa
(integer) 6
127.0.0.1:6379> get aaa
"111aaa"
127.0.0.1:6379>
```

散列类型 `hashmap` (散列类型相当于Java中的HashMap, 他的值是一个字典, 保存很多key, value对)

(`hset` `hget` `hmset` `hmget` `hgetall` `hdel` `hkeys` `hvals` `hlen` `del`删除key)

```
127.0.0.1:6379> hget mapa a
"1"
127.0.0.1:6379> hmset mapb b 2 c 3 d 4
OK
127.0.0.1:6379> hmget mapb b c d
1) "2"
2) "3"
3) "4"
127.0.0.1:6379> hgetall mapb
1) "b"
2) "2"
3) "c"
4) "3"
5) "d"
6) "4"
127.0.0.1:6379> hdel mapb d
(integer) 1
127.0.0.1:6379> hmget mapb b c d
1) "2"
2) "3"
3) (nil)
127.0.0.1:6379> hkeys mapb
1) "b"
2) "c"
127.0.0.1:6379> hvals mapb
1) "2"
2) "3"
127.0.0.1:6379> hlen mapb
(integer) 2
```

列表类型 `list`(列表类型(`list`)用于存储一个有序的字符串列表，常用的操作是向队列两端添加元素或者获得列表的某一片段)

(`lpush` `rpush` `lpop` `rpop` `llen` `lindex`)

```
(integer) 2
127.0.0.1:6379> lpush a 1 2 3
(integer) 3
127.0.0.1:6379> lpop a
"3"
127.0.0.1:6379> lpop a
"2"
127.0.0.1:6379> lpop a
"1"
127.0.0.1:6379> llen a
(integer) 0
127.0.0.1:6379> lpush a 1 2 3
(integer) 3
127.0.0.1:6379> rpop a
"1"
127.0.0.1:6379> rpop a
"2"
127.0.0.1:6379> rpop a
"3"
127.0.0.1:6379> llen a
(integer) 0
127.0.0.1:6379> rpush a 1 2 3
(integer) 3
127.0.0.1:6379> rpop a
"3"
127.0.0.1:6379> rpop a
"2"
127.0.0.1:6379> rpop a
"1"
127.0.0.1:6379> rpop a
(nil)
127.0.0.1:6379> llen a
(integer) 0
127.0.0.1:6379> rpush a 1 2 3
(integer) 3
127.0.0.1:6379> lindex a 2
"3"
127.0.0.1:6379> lindex a 0
"1"
```

集合类型 set(集合中每个元素都是不同的)

(sadd srem smembers sismember scard spop srandmember)

```

[root@serverone bin]# ./redis-cli
127.0.0.1:6379> sadd seta 1 2 3
(integer) 3
127.0.0.1:6379> srem seta 2
(integer) 1
127.0.0.1:6379> smembers seta
1) "1"
2) "3"
127.0.0.1:6379> sismember seta 3
(integer) 1
127.0.0.1:6379> scard seta
(integer) 2
127.0.0.1:6379> spop seta 1
1) "3"
127.0.0.1:6379> scard seta
(integer) 1
127.0.0.1:6379> srandmember seta
"1"
127.0.0.1:6379> srandmember seta 2
1) "1"
127.0.0.1:6379> sadd seta 1 2 3 4 5 6
(integer) 5
127.0.0.1:6379> srandmember seta 3
1) "2"
2) "6"
3) "4"
127.0.0.1:6379> srandmember seta 3
1) "3"
2) "5"
3) "6"

```

有序集合类型 sortedset(有序集合类型与集合类型的区别就是他是有序的)

(zadd zrem zcard zcount)

```

(integer) 1
127.0.0.1:6379> zadd zseta 1 zhangsan 2 lisi 3 wangwu 4 maliu 5 maliu
(integer) 4
127.0.0.1:6379> zcard zseta
(integer) 4
127.0.0.1:6379> zrem zseta 3
(integer) 0
127.0.0.1:6379> zrem zseta wangwu
(integer) 1
127.0.0.1:6379> zcard zseta
(integer) 3
127.0.0.1:6379> zcount zseta 1 2
(integer) 2
127.0.0.1:6379> zcount zseta 1 5
(integer) 3

```

6, 关闭redis

shutdown save/nosave (在连接窗口使用shutdown 使用保存或者不保存数据)

quit (退出后, redis服务会关闭, 再次连接需要启动服务再次调用./redis-server redis.conf)