

大数据库系统

3.2 Redis的安装、五大数据类型概述

3.2 Redis的安装、五大数据类型概述

◆ 主要内容

3.2.1 Redis的安装

3.2.2 Redis的启动与退出

3.2.3 五大数据类型概述

3.2.1 Redis的安装

◆ Redis安装版本

➤ Redis有Linux版和Windows版

- 由于企业里面做Redis开发，多数是Linux版的运用和安装，很少使用Windows版

Redis is written in **ANSI C** and works in most POSIX systems like Linux, *BSD, OS X without external dependencies. Linux and OS X are the two operating systems where Redis is developed and more tested, and we **recommend using Linux for deploying**. Redis may work in Solaris-derived systems like SmartOS, but the support is *best effort*. There is no official support for Windows builds, but Microsoft develops and maintains a [Win-64 port of Redis](#).

官方也推荐使用linux版本

3.2.1 Redis的安装

◆ Redis下载

1、下载安装包或者使用群内分享的安装包

官网下载: <https://redis.io> 下载最新的压缩包

redis-5.0.5.tar.gz

将下载后的redis-5.0.5.tar.gz文件，保存在“/home/hadoop/下载/”目录下

- 如果在windows系统上下载了，可以使用FileZilla等FTP文件传输软件将压缩包传输到虚拟机的linux系统下

[GET STARTED](#)[DOCS](#)[COMMANDS](#)[RESOURCES](#)[COMMUNITY](#)[SUPPORT](#)[Try Redis Cloud](#)[Download](#)

Redis

The open-source, **in-memory data store** used by millions of developers as a **cache**, **vector database**, **document database**, **streaming engine**, and **message broker**.

[Get Started](#)[Read the docs](#)

```
redis> PING
"PONG"
redis> HSET user:1 name antirez vocation artist
(integer) 2
redis> SET e 2.71
"OK"
redis> INCRBYFLOAT e 0.43
"3.14"
redis> RENAME e pi
"OK"
redis>
```

3.2.1 Redis的安装

2、打开一个终端，执行以下命令将Redis解压至/usr/local/目录下并重命名：

```
$cd ~
```

```
$sudo tar -zxvf ./下载/redis-5.0.5.tar.gz -C /usr/local
```

```
$cd /usr/local
```

```
$sudo mv ./redis-5.0.5.tar.gz ./redis
```

tar命令：打包压缩和解压缩

压缩：tar -czvf

sudo命令：使用超级用户权限执行命令，要输入密码

3.2.1 Redis的安装

解压后文件如下图（可能会更多，版本不同）：

```
00-RELEASENOTES  COPYING  Makefile  redis.conf  src
BUGS             deps      MANIFESTO  runtest     tests
CONTRIBUTING    INSTALL  README     sentinel.conf  utils
```

看到里面的makefile文件，很明显接下来使用make命令

3、redis目录的权限赋予hadoop用户（如果管理员用户不是hadoop这里需要修改成对应用户），命令如下：

```
$sudo chown -R hadoop:hadoop ./redis
```

- Chown命令的作用为更改文件和文件夹拥有者：-R表示处理指定目录以及其子目录下的所有文件

3.2.1 Redis的安装

4、接下来进入/usr/local/redis目录（刚刚已经将目录改成了redis），输入以下命令编译Redis：

```
$sudo make
```

同样，执行命令后可能会要求用户输入密码

```
cc migrate.o
cc endianconv.o
cc slowlog.o
cc scripting.o
cc bio.o
cc rio.o
cc rand.o
cc memtest.o
cc crc64.o
cc bitops.o
cc sentinel.o
LINK redis-server
INSTALL redis-sentinel
cc redis-cli.o
LINK redis-cli
cc redis-benchmark.o
LINK redis-benchmark
cc redis-check-dump.o
LINK redis-check-dump
cc redis-check-aof.o
LINK redis-check-aof

Hint: To run 'make test' is a good idea ;)
make[1]: Leaving directory '/usr/local/src/redis-2.6.16/src'
```

这里可以看到，在make最后会提示说，运行make test进行测试，我们可以不用做make test，直接进行下一步

3.2.1 Redis的安装

如果要做make test进行测试的话，则在命令行输入命令：

```
$make test
```

此时可能会出现如下报错信息：

```
cd src && make test
make[1]: Entering directory `/usr/local/src/redis-2.6.16/src'
You need tcl 8.5 or newer in order to run the Redis test
make[1]: *** [test] Error 1
make[1]: Leaving directory `/usr/local/src/redis-2.6.16/src'
make: *** [test] Error 2
```

错误信息提示：要求tcl 8.5或者更高版本才能运行

如果使用的是Ubuntu系统输入如下命令（需要联网）：

```
$ sudo apt install tcl
```

如果是Redhat、Centos则输入（需要联网）

```
$yum install tcl
```

安装完成后再次make test即可

3.2.1 Redis的安装

◆ Make过程中可能碰到的错误

1) 时间错误

- 原因：源码是官方configure过的，官方configure时生成的文件有时间戳信息，Make只能发生在configure之后
- 解决： 修改虚拟机linux系统的时间，`date -s 'yyyy-mm-dd hh:mm:ss'` 重写时间

3.2.1 Redis的安装

◆ Make过程中可能碰到的错误

2) 出现如下错误

```
make[3]: gcc: 命令未找到
make[3]: *** [net.o] 错误 127
make[3]: Leaving directory `/opt/redis-3.0.4/deps/hiredis'
make[2]: *** [hiredis] 错误 2
make[2]: Leaving directory `/opt/redis-3.0.4/deps'
make[1]: [persist-settings] 错误 2 (忽略)
    CC adlist.o
/bin/sh: cc: command not found
make[1]: *** [adlist.o] 错误 127
make[1]: Leaving directory `/opt/redis-3.0.4/src'
make: *** [all] 错误 2
```

- 原因：没有安装gcc，因为redis是C编写的，因此linux下需要gcc编译器
- 解决方法：安装gcc，联网，输入如下命令

```
sudo apt install gcc
```

3.2.1 Redis的安装

◆ Make过程中可能碰到的错误

3) 出现如下错误信息¹:

```
cd src && make all
make[1]: Entering directory `/opt/redis-3.0.4/src'
    CC adlist.o
在包含自 adlist.c: 34 的文件中:
zmalloc.h:50:31: 错误: jemalloc/jemalloc.h: 没有那个文件或目录
zmalloc.h:55:2: 错误: #error "Newer version of jemalloc required"
make[1]: *** [adlist.o] 错误 1
make[1]: Leaving directory `/opt/redis-3.0.4/src'
make: *** [all] 错误 2
```

- 原因：一般出现在第一次make失败后，再make时出现，文件残留问题
- 解决方法：运行“make distclean”之后再make

3.2.1 Redis的安装

5、输入如下命令安装redis:

```
$make install
```

如果想自己制定安装路径的话，可以在make命令之后添加PREFIX参数指定安装路径

例如：\$ make PREFIX=/usr/local/redis install

注意： PREFIX要大写

```
[root@localhost redis]# make PREFIX=/usr/local/redis install
cd src && make install
make[1]: Entering directory `/usr/local/src/redis-2.6.16/src'
Hint: To run 'make test' is a good idea ;)
mkdir -p /usr/local/redis/bin
INSTALL install
INSTALL install
INSTALL install
INSTALL install
INSTALL install
make[1]: Leaving directory `/usr/local/src/redis-2.6.16/src'
```

等待一段时间安装完成

3.2.1 Redis的安装

6、建议备份redis.conf配置文件

- 在redis服务器启动时，需要指定服务器载入一份配置文件²（conf文件）
 - 为了安全起见，将官方原版的配置文件复制一份进行备份，方便修改
- 将redis目录下的redis.conf复制一份到/usr/local/redis/src/目录下

```
cp redis.conf /usr/local/redis/src/
```

今后修改服务器配置文件均对/usr/local/redis/下的redis.conf进行修改，官方原版的作为备份文件存放在/usr/local/redis/src目录下

3.2.2 Redis的启动与退出

◆ Redis组件

安装完成后，进入redis目录的src下可以看到如下几个文件

```
hadoop@fisher-VirtualBox:/usr/local/redis/src$ redis-  
redis-benchmark  redis-check-rdb  redis-sentinel  
redis-check-aof   redis-cli       redis-server
```

redis-benchmark 性能测试工具

redis-cli 客户端

redis-server 服务端

redis-check-aof 日志文件检测工具(比如断电造成日志损坏,可以检测并修复)

redis-check-dump 快照文件检测工具

Redis-sentinel 哨兵模式，运维相关

3.2.2 Redis的启动与退出

◆ 启动redis服务器 (redis-server)

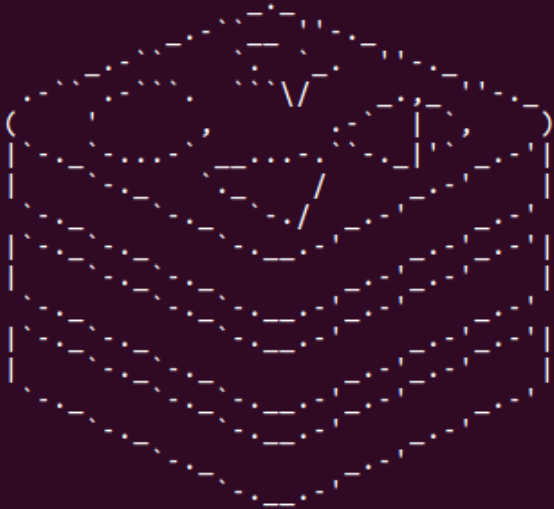
使用redis-server可以启动redis服务器：

1、进入/usr/local/redis目录下，输入如下语句：

```
./src/redis-server redis.conf
```

- redis服务器的配置信息存储在redis.conf中，用户可以自行修改
- redis-server后面跟的redis.conf参数表示的是开启服务器载入的是哪个配置文件


```
hadoop@fisher-VirtualBox: /usr/local/redis/src$ redis-server
3921:C 22 Jan 2021 19:55:48.120 # o000o000o000o Redis is starting o000o000o000o
3921:C 22 Jan 2021 19:55:48.120 # Redis version=5.0.5, bits=64, commit=00000000, modified=0, pid=3921,
just started
3921:C 22 Jan 2021 19:55:48.120 # Warning: no config file specified, using the default config. In orde
r to specify a config file use redis-server /path/to/redis.conf
3921:M 22 Jan 2021 19:55:48.121 * Increased maximum number of open files to 10032 (it was originally s
et to 1024).
```



Redis 5.0.5 (00000000/0) 64 bit

Running in standalone mode
Port: 6379
PID: 3921

<http://redis.io>

```
3921:M 22 Jan 2021 19:55:48.121 # WARNING: The TCP backlog setting of 511 cannot be enforced because /
proc/sys/net/core/somaxconn is set to the lower value of 128.
3921:M 22 Jan 2021 19:55:48.121 # Server initialized
3921:M 22 Jan 2021 19:55:48.121 # WARNING overcommit_memory is set to 0! Background save may fail unde
r low memory condition. To fix this issue add 'vm.overcommit_memory = 1' to /etc/sysctl.conf and then
reboot or run the command 'sysctl vm.overcommit_memory=1' for this to take effect.
3921:M 22 Jan 2021 19:55:48.121 # WARNING you have Transparent Huge Pages (THP) support enabled in you
r kernel. This will create latency and memory usage issues with Redis. To fix this issue run the comma
nd 'echo never > /sys/kernel/mm/transparent_hugepage/enabled' as root, and add it to your /etc/rc.loca
l in order to retain the setting after a reboot. Redis must be restarted after THP is disabled.
3921:M 22 Jan 2021 19:55:48.121 * DB loaded from disk: 0.000 seconds
3921:M 22 Jan 2021 19:55:48.121 * Ready to accept connections
```

启动redis界面如上，可以发现终端一直停留在这个界面上，我们无法在这个终端做任何事情，只能再开一个终端运行redis客户端连接到这个127.0.0.1:6379服务器上

3.2.2 Redis的启动与退出

◆ 使Redis服务器后台运行

Redis服务器占据前台执行不是很方便

可以修改redis.conf文件

1、使用vim打开redis/目录下的redis.conf文件命令如下：

```
vim /usr/local/redis/redis.conf
```

打开后可以看到文件内容，其中找到“daemonize”这个选项

```
##### GENERAL #####  
  
# By default Redis does not run as a daemon. Use 'yes' if you need it.  
# Note that Redis will write a pid file in /var/run/redis.pid when daemonized.  
daemonize no  
  
# When running daemonized, Redis writes a pid file in /var/run/redis.pid by  
# default. You can specify a custom pid file location here.  
pidfile /var/run/redis.pid
```

从描述可以看到，这个选项负责配置redis是否后台执行，默认为no，我们将其改成yes，然后按“:wq”回车，保存退出

3.2.2 Redis的启动与退出

2、再次开启redis客户端，载入刚刚修改的redis.conf文件，命令如下：

```
/usr/local/redis/src/redis-server /usr/local/redis/redis.conf
```

执行后信息显示如下：

```
hadoop@fisher-VirtualBox:~$ redis-server /usr/local/redis/redis.conf
3591:C 09 Mar 2021 10:04:58.031 # o000o000o000o Redis is starting o000o000o000o
3591:C 09 Mar 2021 10:04:58.032 # Redis version=5.0.5, bits=64, commit=00000000,
modified=0, pid=3591, just started
3591:C 09 Mar 2021 10:04:58.032 # Configuration loaded
hadoop@fisher-VirtualBox:~$
```

这时我们依然可以使用终端在命令行输入命令；

3、为了确定redis服务器确实在后台运行，使用ps命令查看进程，输入命令：

```
ps -ef | grep redis
```

```
hadoop@fisher-VirtualBox:/usr/local/redis/src$ ps -ef | grep redis
hadoop    4095   1700    0 20:34 ?        00:00:00 redis-server 127.0.0.1:6379
hadoop    4102   2498    0 20:36 pts/2    00:00:00 grep --color=auto redis
hadoop@fisher-VirtualBox:/usr/local/redis/src$
```

可以看到redis-server的进程正在运行

3.2.2 Redis的启动与退出

◆ 使用客户端连接redis服务器 (redis-cli)

- 在开启了redis服务器端后，可以使用redis-cli命令连接服务器
- redis-cli (Redis Command Line Interface) 是Redis自带的基于命令行的客户端，主要用于与服务器端进行交互，可以使用该客户端来操作Redis的各种命令
- 有3种形式的启动命令
 - 命令1: `./redis-cli` (不指定启动端口)
 - 命令2: `./redis-cli -p 6379` (指定启动端口)
 - 命令3: `./redis-cli -h 127.0.0.1 -p 6379` (指定IP和启动端口)

在服务器端的配置文件redis.conf中可以指定服务器的端口号，默认为6379

```
# Accept connections on the specified port, default is 6379 (IANA #815344).  
# If port 0 is specified Redis will not listen on a TCP socket.  
port 6379
```

1、启动redis服务器（6379）输入命令：

```
/usr/local/src/redis-server /usr/local/redis/redis.conf
```

2、使用redis-cli连接刚刚启动的服务器输入命令：

```
hadoop@fisher-VirtualBox:/usr/local/redis/src$ redis-cli -p 6379  
127.0.0.1:6379> █
```

可以看到提示符变成了127.0.0.1:6379，表示连接成功

3、在客户端输入指令info server 查询服务器信息：

可以看到服务器的版本号、系统、

进程id、端口号等等信息

4、在客户端输入ping，服务器端会回复pong

```
127.0.0.1:6379> ping  
PONG  
127.0.0.1:6379> █
```

```
hadoop@fisher-VirtualBox:~$ redis-cli  
127.0.0.1:6379> info server  
# Server  
redis_version:5.0.5  
redis_git_sha1:00000000  
redis_git_dirty:0  
redis_build_id:644db62e8f7bd816  
redis_mode:standalone  
os:Linux 4.4.0-203-generic x86_64  
arch_bits:64  
multiplexing_api:epoll  
atomicvar_api:atomic-builtin  
gcc_version:5.3.1  
process_id:3592  
run_id:2a0937cc6ee5cf19f2b91610cbc12b93b1e95a6b  
tcp_port:6379  
uptime_in_seconds:149  
uptime_in_days:0  
hz:10  
configured_hz:10  
lru_clock:4642911  
executable:/home/hadoop/redis-server  
config_file:/usr/local/redis/redis.conf  
127.0.0.1:6379>
```

3.2.2 Redis的启动与退出

◆ Redis Hello World!

- 1、启动服务器6379，并开启客户端，连接至6379
- 2、添加键值k1和k2³

```
127.0.0.1:6379> set k1 "hello world"
OK
127.0.0.1:6379> set k2 redis
OK
```

- 3、使用keys命令查看所有键

```
127.0.0.1:6379> keys *
1) "k2"
2) "k1"
127.0.0.1:6379> █
```

- 4、使用get命令获得k1和k2的内容

```
127.0.0.1:6379> get k1
"hello world"
127.0.0.1:6379> get k2
"redis"
127.0.0.1:6379> █
```


3.2.2 Redis的启动与退出

◆ 关闭服务器与客户端

- 1、关闭服务器，在客户端输入shutdown命令

```
127.0.0.1:6379> shutdown  
not connected> 
```

- 2、关闭当前客户端，在客户端输入命令exit

```
127.0.0.1:6379> shutdown  
not connected> exit  
hadoop@fisher-VirtualBox:/usr/local/redis/src$ 
```

即可直接返回linux命令行

- 3、使用ps查看redis进程

```
hadoop@fisher-VirtualBox:/usr/local/redis/src$ ps -ef |grep redis  
hadoop    4275  2498  0 21:23 pts/2    00:00:00 grep --color=auto redis  
hadoop@fisher-VirtualBox:/usr/local/redis/src$ 
```

Redis的服务器进程已经关闭了

3.2.2 Redis的启动与退出

◆ redis-benchmark工具

➤ redis-benchmark是一个压力测试工具，可以指定redis对某个服务器执行一定次数的读写操作（检测电脑读写性能）

1、启动6379服务器

2、进入redis/src/目录

3、执行命令：redis-benchmark -n 100000

-n参数表示输入多少次请求，不加-n，默认100000，今后学RDB，AOF用得到

```
===== SET =====
100000 requests completed in 0.91 seconds
50 parallel clients
3 bytes payload
keep alive: 1

98.42% <= 1 milliseconds
99.67% <= 2 milliseconds
99.80% <= 3 milliseconds
99.84% <= 5 milliseconds
99.90% <= 6 milliseconds
100.00% <= 8 milliseconds
100.00% <= 8 milliseconds
109769.48 requests per second
```

```
===== GET =====
100000 requests completed in 0.89 seconds
50 parallel clients
3 bytes payload
keep alive: 1

98.78% <= 1 milliseconds
99.43% <= 2 milliseconds
99.61% <= 3 milliseconds
99.78% <= 4 milliseconds
99.78% <= 5 milliseconds
99.79% <= 6 milliseconds
99.95% <= 8 milliseconds
100.00% <= 8 milliseconds
111856.82 requests per second
```


3.2.3 五大数据类型概述

◆ Redis五大基本数据类型

STRING（字符串，最基本类型）

HASH（哈希，键值对集合）

LIST（列表，即是堆栈也是队列）

SET（无序集合）

ZSET（sorted set：有序集合）

3.2.3 五大数据类型概述

◆ STRING（字符串，最基本类型）

```
redis> SET key "value"
```

```
OK
```

```
redis> GET key
```

```
"value"
```

字符串

- SET
- SETNX
- SETEX
- PSETEX
- GET
- GETSET
- STRLEN
- APPEND
- SETRANGE
- GETRANGE
- INCR
- INCRBY
- INCRBYFLOAT
- DECR
- DECRBY
- MSET
- MSETNX
- MGET

3.2.3 五大数据类型概述

◆ HASH（哈希，键值对集合）

```
redis> HSET website google www.google.cn
```

```
(integer) 1
```

```
redis> HGET website google
```

```
"www.google.cn"
```

key
↓
website

field	value
Google	www.google.com
Baidu	www.baidu.com
Taobao	www.taobao.com

哈希表

- HSET
- HSETNX
- HGET
- HEXISTS
- HDEL
- HLEN
- HSTRLEN
- HINCRBY
- HINCRBYFLOAT
- HMSET
- HMGET
- HKEYS
- HVALS
- HGETALL
- HSCAN

3.2.3 五大数据类型概述

◆ LIST (列表, 同时兼具堆栈与队列)

```
redis> LPUSH languages python
```

```
(integer) 1
```

```
redis> LPUSH languages java
```

```
(integer) 2
```

```
redis> LRANGE languages 0 -1
```

```
1) "java"
```

```
2) "python"
```

```
redis> LPUSH mylist a b c
```

```
(integer) 3
```

```
redis> LRANGE mylist 0 -1
```

```
1) "c"
```

LPUSH

```
2) "b"
```

```
3) "a"
```



列表

- LPUSH
- LPUSHX
- RPUSH
- RPUSHX
- LPOP
- RPOP
- RPOPLPUSH
- LREM
- LLEN
- LINDEX
- LINSERT
- LSET
- LRANGE
- LTRIM
- BLPOP
- BRPOP
- BRPOPLPUSH

3.2.3 五大数据类型概述

◆ SET（无序集合）

```
redis> SADD bbs "discuz.net"
```

```
(integer) 1
```

```
redis> SADD bbs "tianya.cn" "groups.google.com"
```

```
(integer) 2
```

```
redis> SMEMBERS bbs
```

```
1) "discuz.net"
```

```
2) "groups.google.com"
```

```
3) "tianya.cn"
```

集合

- SADD
- SISMEMBER
- SPOP
- SRANDMEMBER
- SREM
- SMOVE
- SCARD
- SMEMBERS
- SSCAN
- SINTER
- SINTERSTORE
- SUNION
- SUNIONSTORE
- SDIFF
- SDIFFSTORE

3.2.3 五大数据类型概述

◆ ZSET (sorted set: 有序集合)

```
redis> ZADD page_rank 10 google.com
```

```
(integer) 1
```

```
redis> ZADD page_rank 9 baidu.com 8 bing.com
```

```
(integer) 2
```

```
redis> ZRANGE page_rank 0 -1 WITHSCORES
```

```
1) "bing.com"
```

```
2) "8"
```

```
3) "baidu.com"
```

```
4) "9"
```

```
5) "google.com"
```

```
6) "10"
```

有序集合

- ZADD
- ZSCORE
- ZINCRBY
- ZCARD
- ZCOUNT
- ZRANGE
- ZREVRANGE
- ZRANGEBYSCORE
- ZREVRANGEBYSCORE
- ZRANK
- ZREVRANK
- ZREM
- ZREMRANGEBYRANK
- ZREMRANGEBYSCORE
- ZRANGEBYLEX
- ZLEXCOUNT
- ZREMRANGEBYLEX
- ZSCAN
- ZUNIONSTORE
- ZINTERSTORE

3.2.3 五大数据类型概述

◆ 其他redis命令

通用key的基本命令

运维的常用命令

事务命令

消息订阅命令

等等

中文的redis常用命令网站: www.redis.net.cn/order

Redis 集合(Set)

Redis Sunion 命令

Redis Scard 命令

Redis Srandmember 命令

Redis Smembers 命令

Redis Sinter 命令

Redis Srem 命令

Redis Smove 命令

Redis Sadd 命令

Redis Sismember 命令

Redis Sdiffstore 命令

Redis Sdiff 命令

Redis Sscan 命令

Redis Sinterstore 命令

Redis Sunionstore 命令

Redis Spop 命令

Redis Sadd 命令 - 向集合添加一个或多个成员

Redis 集合(Set)

Redis Sadd 命令将一个或多个成员元素加入到集合中，已经存在于集合的成员元素将被忽略。

假如集合 key 不存在，则创建一个只包含添加的元素作成员的集合。

当集合 key 不是集合类型时，返回一个错误。

注意：在Redis2.4版本以前，SADD 只接受单个成员值。

语法

redis Sadd 命令基本语法如下：

```
1. redis 127.0.0.1:6379> SADD KEY_NAME VALUE1..VALUEN
```

可用版本

>= 1.0.0

返回值

被添加到集合中的新元素的数量，不包括被忽略的元素。

实例

```
1. redis 127.0.0.1:6379> SADD myset "hello"
2. (integer) 1
3. redis 127.0.0.1:6379> SADD myset "foo"
4. (integer) 1
5. redis 127.0.0.1:6379> SADD myset "hello"
6. (integer) 0
7. redis 127.0.0.1:6379> SMEMBERS myset
8. 1) "hello"
9. 2) "foo"
```

Redis 集合(Set)

字符串

哈希表

列表

集合

有序集合

HyperLogLog

地理位置

位图

数据库

自动过期

事务

Lua 脚本

持久化

发布与订阅

复制

客户端与服务器

配置选项

调试

内部命令

功能文档

事务

- MULTI
- EXEC
- DISCARD
 - 返回值
 - 代码示例
- WATCH
- UNWATCH

持久化

- SAVE
- BGSAVE
- BGREWRITEAOF
- LASTSAVE

发布与订阅

- PUBLISH
- SUBSCRIBE
- PSUBSCRIBE
- UNSUBSCRIBE
- PUNSUBSCRIBE
- PUBSUB

自动过期

- EXPIRE
- EXPIREAT
- TTL
- PERSIST
- PEXPIRE
- PEXPIREAT
- PTTL

位图

- SETBIT
- GETBIT
- BITCOUNT
- BITPOS
- BITOP
- BITFIELD

数据库

EXISTS

TYPE

RENAME

RENAMENX

MOVE

DEL

RANDOMKEY

DBSIZE

KEYS

SCAN

SORT

FLUSHDB

FLUSHALL

SELECT

SWAPDB

总结

◆ 主要内容

3.2.1 Redis的安装

3.2.2 Redis的启动与退出

3.2.3 五大数据类型概述