



链滴

# Redis 位图 - 学习笔记

作者: [Calon](#)

原文链接: <https://ld246.com/article/1540813724709>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



我们学习Redis时，都知道Redis支持以下几种数据类型：string（字符串），hash（哈希），list（表），set（集合）及zset(sorted set: 有序集合)。现在介绍另一种比较少见的类型：位图。其实位并不是一种特殊的数据结构，它的内容其实就是普通的字符串，也就是 byte 数组。我们可以使用普通的 get/set 直接获取和设置整个位图的内容，也可以使用位图操作命令 getbit/setbit 来处理。在开过程中，位图有什么作用呢？大家想一下，比如需要保存用户一年的签到记录，签了是 1，没签是 0 要记录 365 天。如果使用数据库表来存储，每个用户每年就会产生365条记录，当用户上亿的时候，要的存储空间是惊人的。

## 基本使用

setbit/getbit命令：

```
127.0.0.1:6379> setbit s 1 1
```

```
(integer) 0
```

```
127.0.0.1:6379> setbit s 2 1
```

```
(integer) 0
```

```
127.0.0.1:6379> setbit s 4 1
```

```
(integer) 0
```

```
127.0.0.1:6379> setbit s 9 1
```

```
(integer) 0
```

```
127.0.0.1:6379> setbit s 10 1
```

```
(integer) 0
```

```
127.0.0.1:6379> setbit s 13 1
```

```
(integer) 0
```

```
127.0.0.1:6379> setbit s 15 1
```

```
(integer) 0
```

```
127.0.0.1:6379> get s
```

```
"he"
```

```
127.0.0.1:6379> getbit s 1  
(integer) 1
```

从上面的命令大家可以看到以s为key进行操作的，用get命令可以取出“he”的字符串，说明上面的setbit操作刚好凑成he的字符串，如果使用getbit key offset命令取指定偏移位置的值，则可以按照设置取出来。

## 统计和查找

除了上面的命令当然不能满足开发的要求，Redis 还提供了位图统计指令 bitcount 和位图查找指令 bitpos，bitcount 用来统计指定位置范围内 1 的个数，bitpos 用来查找指定范围内出现的第一个 0 或

。

比如我们可以通过 bitcount 统计用户一共签到了多少天，通过 bitpos 指令查找用户从哪一天开始第几次签到。如果指定了范围参数[start, end]，就可以统计在某个时间范围内用户签到了多少天，用户自天以后的哪天开始签到。

遗憾的是，start 和 end 参数是字节索引，也就是说指定的位范围必须是 8 的倍数，而不能任意指定因为这个设计，我们无法直接计算某个月内用户签到了多少天，而必须要把这个月所覆盖的字节内容取出来 (getrange 可以取出字符串的子串) 然后在内存里进行统计，这个非常繁琐。

```
127.0.0.1:6379> set w hello  
OK  
127.0.0.1:6379> bitcount w  
(integer) 21  
127.0.0.1:6379> bitcount w 0 0 # 第一个字符中 1 的位数  
(integer) 3  
127.0.0.1:6379> bitcount w 0 1 # 前两个字符中 1 的位数  
(integer) 7  
127.0.0.1:6379> bitpos w 0 # 第一个 0 位  
(integer) 0  
127.0.0.1:6379> bitpos w 1 # 第一个 1 位  
(integer) 1  
127.0.0.1:6379> bitpos w 1 1 1 # 从第二个字符算起，第一个 1 位  
(integer) 9  
127.0.0.1:6379> bitpos w 1 2 2 # 从第三个字符算起，第一个 1 位  
(integer) 17
```

<br/>

<br/>

<br/>

<br/>

<br/>

扫一扫有惊喜：

![imagepng](http://itechor.top/solo/upload/bb791a58c3a84193b7f643b6849482c5\_image.png) (http://ym0214.com)