

如何阅读 Redis 源码?

在这篇文章中， 我将向大家介绍一种我认为比较合理的 Redis 源码阅读顺序， 希望可以给对 Redis 有兴趣并打算阅读 Redis 源码的朋友带来一点帮助。

第 1 步：阅读数据结构实现

刚开始阅读 Redis 源码的时候， 最好从数据结构的相关文件开始读起， 因为这些文件和 Redis 中的其他部分耦合最少， 并且这些文件所实现的数据结构在大部分算法书上都可以了解到， 所以从这些文件开始读是最轻松的、难度也是最低的。

下表列出了 Redis 源码中， 各个数据结构的实现文件：

文件	内容
sds.h 和 sds.c	Redis 的动态字符串实现。
adlist.h 和 adlist.c	Redis 的双端链表实现。
dict.h 和 dict.c	Redis 的字典实现。
redis.h 中的 zskiplist 结构和 zskiplistNode 结构， 以及 t_zset.c 中所有以 zsl 开头的函数， 比如 zslCreate、zslInsert、zslDeleteNode， 等等。	Redis 的跳跃表实现。
hyperloglog.c 中的 hllhdr 结构， 以及所有以 hll 开头的函数。	Redis 的 HyperLogLog 实现。

第 2 步：阅读内存编码数据结构实现

在阅读完和数据结构有关的文件之后， 接下来就应该阅读内存编码（encoding）数据结构了。

和普通的数据结构一样， 内存编码数据结构基本上是独立的， 不和其他模块耦合， 但是区别在于：

- 上一步要读的数据结构， 比如双端链表、字典、HyperLogLog， 在算法书上或者相关的论文上都可以找到资料介绍。
- 而内存编码数据结构却不容易找到相关的资料， 因为这些数据结构都是 Redis 为了节约内存而专门开发出来的， 换句话说， 这些数据结构都是特制（ad hoc）的， 除了 Redis 源码中的文档之外， 基本上找不到其他资料来了解这些特制的数据结构。

不过话又说回来， 虽然内存编码数据结构是 Redis 特制的， 但它们基本都和内存分配、指针操作、位操作这些底层的東西有关， 读者只要认真阅读源码中的文档， 并在有需要时， 画图来分析这些数据结构， 那么要完全理解这些内存编码数据结构的运作原理并不难， 当然这需要花一些功夫。

下表展示了 Redis 源码中， 各个内存编码数据结构的实现文件：

文件	内容
intset.h 和 intset.c	整数集合（intset）数据结构。
ziplist.h 和 ziplist.c	压缩列表（zip list）数据结构。

第 3 步：阅读数据类型实现

在完成以上两个阅读步骤之后， 我们就读完了 Redis 六种不同类型的键（字符串、散列、列表、集合、有序集合、HyperLogLog）的所有底层实现结构了。

接下来， 为了知道 Redis 是如何通过以上提到的数据结构来实现不同类型的键， 我们需要阅读实现各个数据类型的文件， 以及 Redis 的对象系统文件， 这些文件包括：

文件	内容
object.c	Redis 的对象（类型）系统实现。
t_string.c	字符串键的实现。
t_list.c	列表键的实现。
t_hash.c	散列键的实现。
t_set.c	集合键的实现。
t_zset.c 中除 zsl 开头的函数之外的所有函数。	有序集合键的实现。
hyperloglog.c 中所有以 pf 开头的函数。	HyperLogLog 键的实现。

第 4 步：阅读数据库实现相关代码

在读完了 Redis 使用所有底层数据结构， 以及 Redis 是如何使用这些数据结构来实现不同类型的键之后， 我们就可以开始阅读 Redis 里面和数据库有关的代码了， 它们分别是：

文件	内容
redis.h 文件中的 redisDb 结构， 以及 db.c 文件。	Redis 的数据库实现。
notify.c	Redis 的数据库通知功能实现代码。
rdb.h 和 rdb.c	Redis 的 RDB 持久化实现代码。
aof.c	Redis 的 AOF 持久化实现代码。

选读

Redis 有一些独立的功能模块， 这些模块可以在完成第 4 步之后阅读， 它们包括：

文件	内容
redis.h 文件的 pubsubPattern 结构， 以及 pubsub.c 文件。	发布与订阅功能的实现。
redis.h 文件的 multiState 结构以及 multiCmd 结构， multi.c 文件。	事务功能的实现。
sort.c	SORT 命令的实现。
bitops.c	GETBIT、SETBIT 等二进制位操作命令的实现。

第 5 步：阅读客户端和服务器的相关代码

在阅读完数据库实现代码， 以及 RDB 和 AOF 两种持久化的代码之后， 我们可以开始阅读客户端和 Redis 服务器本身的实现代码， 和这些代码有关的文件是：

文件	内容
ae.c， 以及任意一个 ae_*.c 文件（取决于你所使用的多路复用库）。	Redis 的事件处理器实现（基于 Reactor 模式）。
networking.c	Redis 的网络连接库， 负责发送命令回复和接受命令请求， 同时也负责创建/销毁客户端， 以及通信协议分析等工作。
redis.h 和 redis.c 中和单机 Redis 服务器有关的部分。	单机 Redis 服务器的实现。

如果读者能完成以上 5 个阅读步骤的话， 那么恭喜你， 你已经了解了单机的 Redis 服务器是怎样处理命令请求和返回命令回复， 以及是 Redis 怎样操作数据库的了， 这是 Redis 最重要的部分， 也是之后继续阅读多机功能的基础。

选读

Redis 有一些独立的功能模块， 这些模块可以在完成第 5 步之后阅读， 它们包括：

文件	内容
scripting.c	Lua 脚本功能的实现。
slowlog.c	慢查询功能的实现。
monitor.c	监视器功能的实现。

第 6 步：阅读多机功能的实现

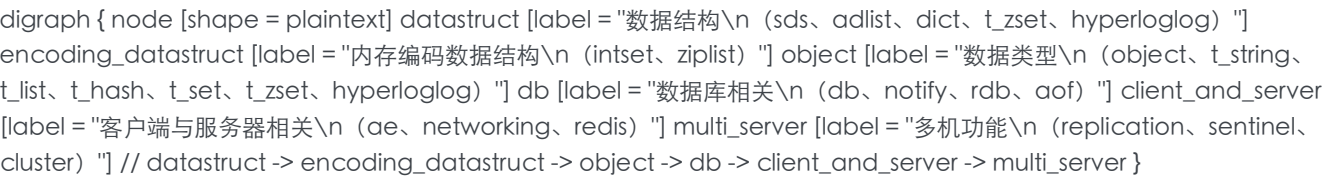
在弄懂了 Redis 的单机服务器是怎样运作的之后， 就可以开始阅读 Redis 多机功能的实现代码了， 和这些功能有关的文件为：

文件	内容
replication.c	复制功能的实现代码。
sentinel.c	Redis Sentinel 的实现代码。
cluster.c	Redis 集群的实现代码。

注意， 因为 Redis Sentinel 用到了复制功能的代码， 而集群又用到了复制和 Redis Sentinel 的代码， 所以在阅读这三个模块的时候， 记得先阅读复制模块， 然后阅读 Sentinel 模块， 最后才阅读集群模块， 这样理解起来就会更得心应手。

如果你连这三个模块都读完了的话， 那么恭喜你， 你已经读完了 Redis 单机功能和多机功能的所有代码了！

下图总结了本文介绍的阅读顺序：



结语

Redis 的设计非常简洁、优美、精巧和高效，任何人只要愿意去阅读它的代码的话，应该都会有所收获的。

希望这篇文章能够给想要阅读 Redis 代码的朋友们带来一些帮助，也欢迎各位随时和我讨论 Redis 源码方面的问题，或者跟我分享各位阅读 Redis 源码的心得和经验。

另外我的 [Redis 源码注释](#) 项目以及《Redis 设计与实现》一书对于理解 Redis 的源代码应该也会有所帮助，有兴趣的朋友可以自行了解该项目/书本。

黄健宏 (huangz)
2014.7.28

留言

2 条评论

blog.huangz.me

1 登录


Recommend

分享

按从新到旧排序



加入讨论...


 **麦客** · 6个月前

在sds.c sdstrim 的注释描述是不是写错了呢？

```
* s = sdsnew("AA...AA.a.a.aHelloWorld :::");
* s = sdstrim(s,"A. :");
* printf("%s\n", s);
```

这个输出不会是Hello World吧，是我理解错了？

^ | v · 回复 · 分享 ·

 **unasm** · 6个月前

不错，一步步来读

^ | v · 回复 · 分享 ·

在 **BLOG.HUANGZ.ME** 上还有.....

这是什么？

我进行翻译的方法 — huangz/blog

20 条评论 · 2年前



huangz — 用 VIM。

第 3 章：套接字编程简介 — Right Track Wrong Train


2 条评论 · 3年前



huangz — 谢谢提醒，已经修复了：
[https://github.com/huangz1990/...](https://github.com/huangz1990/)

关于编写参考手册的思考 — huangz/blog


4 条评论 · 7个月前



huangz — 谢谢关注，是数据库方面的，目前进展顺利，我会争取尽快公布具体消息。

翻译笔记之四：消除人称代词 — huangz/blog

4 条评论 · 10个月前



Hehe — 一般的翻译能做到信和达就可以了，您还要追求雅，洁癖啊

订阅

在您的网站上使用Disqus Add Disqus Add

隐私