

多位业界专家**联合推荐**
来自一线开发者的**实战经验总结**
真正零基础入门，深入浅出全面剖析Redis
任务驱动式学习，**轻松掌握Redis实战知识**

Redis

入门指南

李子骅 编著



 人民邮电出版社
POSTS & TELECOM PRESS

Redis

入门指南

李子骅 编著



人民邮电出版社
北京

958088810

图书在版编目 (CIP) 数据

Redis入门指南 / 李子骅编著. — 北京: 人民邮电出版社, 2013.5
ISBN 978-7-115-31294-5

I. ①R… II. ①李… III. ①数据库—基本知识
IV. ①TP311.13

中国版本图书馆CIP数据核字(2013)第050956号

内 容 提 要

本书是一本 Redis 的入门指导书籍, 以通俗易懂的方式介绍了 Redis 基础与实践方面的知识, 包括历史与特性、在开发和生产环境中部署运行 Redis、数据类型与命令、使用 Redis 实现队列、事务、复制、管道、持久化、优化 Redis 存储空间等内容, 并采用任务驱动的方式介绍了 PHP、Ruby、Python 和 Node.js 这 4 种语言的 Redis 客户端库的使用方法。

本书的目标读者不仅包括 Redis 的新手, 还包括那些已经掌握 Redis 使用方法的人。对于新手而言, 本书的内容由浅入深且紧贴实践, 旨在让读者真正能够即学即用; 对于已经了解 Redis 的读者, 通过本书的大量实例以及细节介绍, 也能发现很多新的技巧。

Redis 入门指南

-
- ◆ 编 著 李子骅
责任编辑 杨海玲
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京鑫正大印刷有限公司印刷
 - ◆ 开本: 800×1000 1/16
印张: 12
字数: 253 千字 2013 年 5 月第 1 版
印数: 1-3 500 册 2013 年 5 月北京第 1 次印刷

ISBN 978-7-115-31294-5

定价: 39.00 元

读者服务热线: (010)67132692 印装质量热线: (010)67129223
反盗版热线: (010)67171154

前言

Redis 如今已经成为 Web 开发社区中最火热的内存数据库之一，而它的诞生距现在不过才 4 年。随着 Web 2.0 的蓬勃发展，网站数据快速增长，对高性能读写的需求也越来越多，再加上半结构化的数据比重逐渐变大，人们对早已被铺天盖地地运用着的关系数据库能否适应现今的存储需求产生了疑问。而 Redis 的迅猛发展，为这个领域注入了全新的思维。

Redis 凭借其全面的功能得到越来越多的公司的青睐，从初创企业到新浪微博这样拥有着几百台 Redis 服务器的大公司，都能看到 Redis 的身影。Redis 也是一个名副其实的多面手，无论是存储、队列还是缓存系统，都有它的用武之地。

本书将从 Redis 的历史讲起，结合基础与实践，带领读者一步步进入 Redis 的世界。

目标读者

本书假定读者是 Redis 的新手，甚至可能连 Redis 是什么都没听说过。本书将会详细地介绍 Redis 是什么以及为什么要使用 Redis，旨在能够让读者从零开始逐步晋升为一个优秀的 Redis 开发者。

本书还包含了很多 Redis 实践方面的知识，对于有经验的 Redis 开发者，大可以直接跳过已经掌握的内容，只阅读感兴趣的部分。每章的引言都简要介绍了本章要讲解的内容，供读者参考。

本书并不需要读者有任何 Redis 的背景知识，不过如果读者有 Web 后端开发经验或 Linux 系统使用经验，阅读本书将会更加得心应手。

组织结构

第 1 章介绍了 Redis 的历史与特性，主要回答两个初学者最关心的问题，即 Redis 是什么和为什么要使用 Redis。

第 2 章讲解了如何安装和运行 Redis。如果你身旁的计算机没有运行 Redis，那么一定不要错过这一章，因为本书后面的部分都需要读者最好能一边阅读一边实践，以提高学习效率。本章中还会介绍 Redis 命令行客户端的使用方法等基础知识，这些都是实践前需要掌握的知识。

第 3 章介绍了 Redis 的数据类型。本章讲解的不仅是每个数据类型的介绍和命令的格式，还会着重讲解每个数据类型分别在实践中如何使用。整个第 3 章会带领读者从零开始，一步步地使用 Redis 构建一个博客系统，旨在帮助读者在学习完本章的内容之后可以直接在自己的项

目 录

第 1 章 简介	1
1.1 历史与发展	1
1.2 特性	2
1.2.1 存储结构	2
1.2.2 内存存储与持久化	3
1.2.3 功能丰富	3
1.2.4 简单稳定	4
第 2 章 准备	7
2.1 安装 Redis	7
2.1.1 在 POSIX 系统中安装	7
2.1.2 在 OS X 系统中安装	8
2.1.3 在 Windows 中安装	9
2.2 启动和停止 Redis	11
2.2.1 启动 Redis	11
2.2.2 停止 Redis	14
2.3 Redis 命令行客户端	14
2.3.1 发送命令	14
2.3.2 命令返回值	15
2.4 配置	16
2.5 多数据库	17
第 3 章 入门	19
3.1 热身	19
3.2 字符串类型	21
3.2.1 介绍	22
3.2.2 命令	22
3.2.3 实践	26
3.2.4 命令拾遗	28
3.3 散列类型	32
3.3.1 介绍	33
3.3.2 命令	34
3.3.3 实践	37
3.3.4 命令拾遗	39

3.4 列表类型	40
3.4.1 介绍	41
3.4.2 命令	41
3.4.3 实践	44
3.4.5 命令拾遗	46
3.5 集合类型	48
3.5.1 介绍	48
3.5.2 命令	49
3.5.3 实践	52
3.5.4 命令拾遗	54
3.6 有序集合类型	57
3.6.1 介绍	57
3.6.2 命令	58
3.6.3 实践	62
3.6.4 命令拾遗	63
第4章 进阶	67
4.1 事务	67
4.1.1 概述	68
4.1.2 错误处理	69
4.1.3 WATCH 命令介绍	70
4.2 生存时间	72
4.2.1 命令介绍	73
4.2.2 实现访问频率限制之一	75
4.2.3 实现访问频率限制之二	76
4.2.4 实现缓存	77
4.3 排序	78
4.3.1 有序集合的集合操作	78
4.3.2 SORT 命令	79
4.3.3 BY 参数	81
4.3.4 GET 参数	83
4.3.5 STORE 参数	84
4.3.6 性能优化	85
4.4 消息通知	85
4.4.1 任务队列	86
4.4.2 使用 Redis 实现任务队列	87
4.4.3 优先级队列	88
4.4.4 “发布/订阅”模式	90
4.4.5 按照规则订阅	91
4.5 管道	92
4.6 节省空间	93
4.6.1 精简键名和键值	94
4.6.2 内部编码优化	94

第 5 章 实践	103
5.1 PHP 与 Redis	103
5.1.1 安装	104
5.1.2 使用方法	104
5.1.3 简便用法	105
5.1.4 实践: 用户注册登录功能	107
5.2 Ruby 与 Redis	111
5.2.1 安装	111
5.2.2 使用方法	111
5.2.3 简便用法	112
5.2.4 实践: 自动完成	112
5.3 Python 与 Redis	116
5.3.1 安装	116
5.3.2 使用方法	116
5.3.3 简便用法	117
5.3.4 实践: 在线的好友	117
5.4 Node.js 与 Redis	123
5.4.1 安装	123
5.4.2 使用方法	123
5.4.3 简便用法	125
5.4.4 实践: IP 地址查询	127
第 6 章 脚本	131
6.1 概览	131
6.1.1 脚本介绍	132
6.1.2 实例: 访问频率限制	132
6.2 Lua 语言	133
6.2.1 Lua 语法	134
6.2.2 标准库	143
6.2.3 其他库	146
6.3 Redis 与 Lua	147
6.3.1 在脚本中调用 Redis 命令	147
6.3.2 从脚本中返回值	148
6.3.3 脚本相关命令	148
6.3.4 应用实例	149
6.4 深入脚本	152
6.4.1 KEYS 与 ARGV	152
6.4.2 沙盒与随机数	153
6.4.3 其他脚本相关命令	154
6.4.4 原子性和执行时间	155

第 7 章 管理	157
7.1 持久化	157
7.1.1 RDB 方式	157
7.1.2 AOF 方式	159
7.2 复制	161
7.2.1 配置	161
7.2.2 原理	163
7.2.3 图结构	164
7.2.4 读写分离	165
7.2.5 从数据库持久化	165
7.3 安全	166
7.3.1 可信的环境	166
7.3.2 数据库密码	166
7.3.3 命名命令	167
7.4 通信协议	167
7.4.1 简单协议	167
7.4.2 统一请求协议	169
7.5 管理工具	170
7.5.1 redis-cli	170
7.5.2 phpRedisAdmin	171
7.5.3 Rdbtools	173
附录 A Redis 命令属性	175
A.1 REDIS_CMD_WRITE	175
A.2 REDIS_CMD_DENYOOM	177
A.3 REDIS_CMD_NOSCRIPT	178
A.4 REDIS_CMD_RANDOM	179
A.5 REDIS_CMD_SORT_FOR_SCRIPT	179
A.6 REDIS_CMD_LOADING	179
附录 B 配置参数索引	181

第 1 章

简介

Redis 是一个开源的高性能键值对数据库。它通过提供多种键值数据类型来适应不同场景下的存储需求，并借助许多高层级的接口使其可以胜任如缓存、队列系统等不同的角色。

本章将分别介绍 Redis 的历史和特性，以使读者能够快速地对 Redis 有一个全面的了解。

1.1 历史与发展

2008 年，意大利的一家创业公司 Merzia^①推出了一款基于 MySQL 的网站实时统计系统 LLOOGG^②，然而没过多久该公司的创始人 Salvatore Sanfilippo 便开始对 MySQL 的性能感到失望，于是他决定亲自为 LLOOGG 量身定做一个数据库，并于 2009 年开发完成，这个数据库就是 Redis。不过 Salvatore Sanfilippo 并不满足只将 Redis 用于 LLOOGG 这一款产品，而是希望让更多的人使用它，于是在同一年 Salvatore Sanfilippo 将 Redis 开源发布，并开始和 Redis 的另一名主要的代码贡献者 Pieter Noordhuis 一起继续着 Redis 的开发，直到今天。

Salvatore Sanfilippo 自己也没有想到，短短的几年时间，Redis 就拥有了庞大的用户群体。Hacker News 在 2012 年发布了一份数据库的使用情况调查^③，结果显示有近 12% 的公司在使用 Redis。国内如新浪微博、街旁和知乎，国外如 GitHub、Stack Overflow、Flickr、暴雪和 Instagram，都是 Redis 的用户。

① 见 <http://merzia.com>。

② 见 <http://lloogg.com>。

③ 见 <http://news.ycombinator.com/item?id=4833188>。

VMware 公司从 2010 年开始赞助 Redis 的开发, Salvatore Sanfilippo 和 Pieter Noordhuis 也分别于同年的 3 月和 5 月加入 VMware, 全职开发 Redis。

Redis 的代码托管在 GitHub 上, 开发十分活跃 (见图 1-1)。截至交稿时, Redis 的最新版本是 2.6.9。

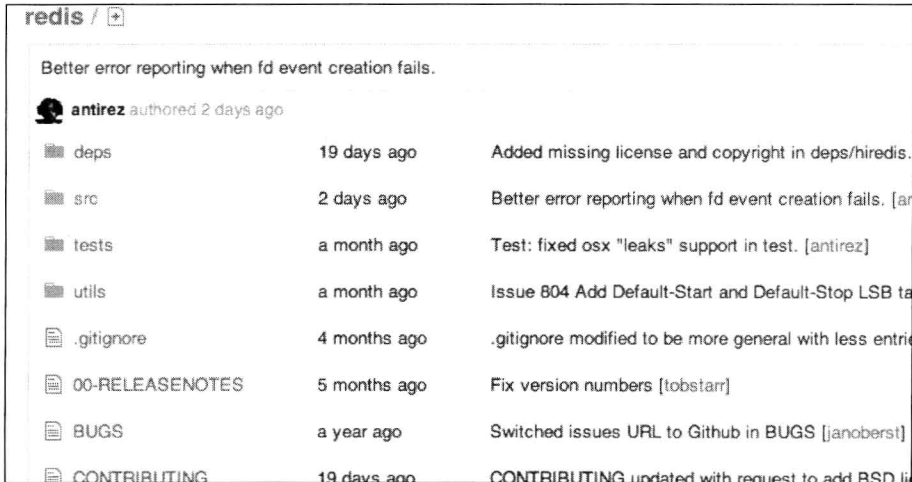


图 1-1 antirez 是 Salvatore Sanfilippo 的 GitHub 用户名, 截图的前两天他还提交过代码

1.2 特性

作为一款个人开发的数据库, Redis 究竟有什么魅力吸引了如此多的用户呢?

1.2.1 存储结构

有过脚本语言编程经验的读者对字典 (或称映射、关联数组) 数据结构一定很熟悉, 如代码 `dict["key"] = "value"` 中 `dict` 是一个字典结构变量, 字符串 `"key"` 是键名, 而 `"value"` 是键值, 在字典中我们可以获取或设置键名对应的键值, 也可以删除一个键。

Redis 是 REMote DIctionary Server (远程字典服务器) 的缩写, 它以字典结构存储数据, 并允许其他应用通过 TCP 协议读写字典中的内容。同大多数脚本语言中的字典一样, Redis 字典中的键值除了可以是字符串, 还可以是其他数据类型。到目前为止 Redis 支持的键值数据类型如下:

- 字符串类型
- 散列类型
- 列表类型
- 集合类型

- 有序集合类型

这种字典形式的存储结构与常见的 MySQL 等关系数据库的二维表形式的存储结构有很大的差异。举个例子，如下所示，我们在程序中使用 `post` 变量存储了一篇文章的数据（包括标题、正文、阅读量和标签）：

```
post["title"] = "Hello World!"
post["content"] = "Blablaba..."
post["views"] = 0
post["tags"] = ["PHP", "Ruby", "Node.js"]
```

现在我们将这篇文章的数据存储在数据库中，并且要求可以通过标签检索出文章。如果使用关系数据库存储，一般会将其中的标题、正文和阅读量存储在一个表中，而将标签存储在另一个表中，然后使用第三个表连接文章和标签表^①。需要查询时还得将三个表进行连接，不是很直观。而 Redis 字典结构的存储方式和对多种键值数据类型的支持使得开发者可以将程序中的数据直接映射到 Redis 中，数据在 Redis 中的存储形式和其在程序中的存储方式非常相近。使用 Redis 的另一个优势是它对不同的数据类型提供了非常方便的操作方式，如使用集合类型存储文章标签，Redis 可以对标签进行如交集、并集这样的集合运算操作。3.5 节会专门介绍如何借助集合运算轻易地实现“找出所有同时属于 A 标签和 B 标签且不属于 C 标签”这样关系数据库实现起来性能不高且较为繁琐的操作。

1.2.2 内存存储与持久化

Redis 数据库中的所有数据都存储在内存中。由于内存的读写速度远快于硬盘，因此 Redis 在性能上对比其他基于硬盘存储的数据库有非常明显的优势，在一台普通的笔记本电脑上，Redis 可以在一秒内读写超过十万个键值。

将数据存储在内存中也有问题，例如，程序退出后内存中的数据会丢失。不过 Redis 提供了对持久化的支持，即将可以内存中的数据异步写入到硬盘中，同时不影响继续提供服务。

1.2.3 功能丰富

Redis 虽然是作为数据库开发的，但由于其提供了丰富的功能，越来越多的人将其用作缓存、队列系统等。Redis 可谓是名副其实的多面手。

Redis 可以为每个键设置生存时间（Time To Live, TTL），生存时间到期后键会自动被删除。这一功能配合出色的性能让 Redis 可以作为缓存系统来使用，而且由于 Redis 支持持久化和丰富的数据类型，使其成为了另一个非常流行的缓存系统 Memcached 的有力竞争者。

^① 这是一种符合第三范式的设计。事实上还可以使用其他方式来实现标签系统，参阅 <http://tagging.pui.-ch/-post/37027745720/tags-database-schemas> 以了解更多相关资料。

讨论 关于 Redis 和 Memcached 优劣的讨论一直是一个热门的话题。在性能上 Redis 是单线程模型，而 Memcached 支持多线程，所以在多核服务器上后者的性能更高一些。然而，前面已经介绍过，Redis 的性能已经足够优异，在绝大部分场合下其性能都不会成为瓶颈。所以在使用时更应该关心的是二者在功能上的区别，如果需要用到高级的数据类型或是持久化等功能，Redis 将会是 Memcached 很好的替代品。

作为缓存系统，Redis 还可以限定数据占用的最大内存空间，在数据达到空间限制后可以按照一定的规则自动淘汰不需要的键。

除此之外，Redis 的列表类型键可以用来实现队列，并且支持阻塞式读取，可以很容易地实现一个高性能的优先级队列。同时在更高层面上，Redis 还支持“发布/订阅”的消息模式，可以基于此构建聊天室^①等系统。

1.2.4 简单稳定

即使功能再丰富，如果使用起来太复杂也很难吸引人。Redis 直观的存储结构使得通过程序与 Redis 交互十分简单。在 Redis 中使用命令来读写数据，命令语句之于 Redis 就相当于 SQL 语言之于关系数据库。例如在关系数据库中要获取 posts 表内 id 为 1 的记录的 title 字段的值可以使用如下 SQL 语句实现：

```
SELECT title FROM posts WHERE id = 1 LIMIT 1
```

相对应的，在 Redis 中要读取键名为 post:1 的散列类型键的 title 字段的值，可以使用如下命令语句实现：

```
HGET post:1 title
```

其中 HGET 就是一个命令。Redis 提供了一百多个命令（如图 1-2 所示），听起来很多，但是常用的却只有十几个，并且每个命令都很容易记忆。读完第 3 章你就会发现 Redis 的命令比 SQL 语言要简单很多。

Redis 提供了几十种不同编程语言的客户端库，这些库都很好地封装了 Redis 的命令，使得在程序中与 Redis 进行交互变得更容易。有些库还提供了可以将编程语言中的数据类型直接以相应的形式存储到 Redis 中（如将数组直接以列表类型存入 Redis）的简单方法，使用起来非常方便。

Redis 使用 C 语言开发，代码量只有 3 万多行。这降低了用户通过修改 Redis 源代码来使之更适合自己的项目需要的门槛。对于希望“榨干”数据库性能的开发者的而言，这无疑是一个很大的吸引力。

Redis 是开源的，所以事实上 Redis 的开发者的并不止 Salvatore Sanfilippo 和 Pieter Noordhuis。截至目前，有将近 100 名开发者为 Redis 贡献了代码。良好的开发氛围和严谨

^① Redis 的贡献者之一 Pieter Noordhuis 提供了一个使用该模式开发的聊天室的例子，见 <https://gist.github.com/348262>。

的版本发布机制使得 Redis 的稳定版本非常可靠，如此多的公司在项目中使用了 Redis 也可以印证这一点。

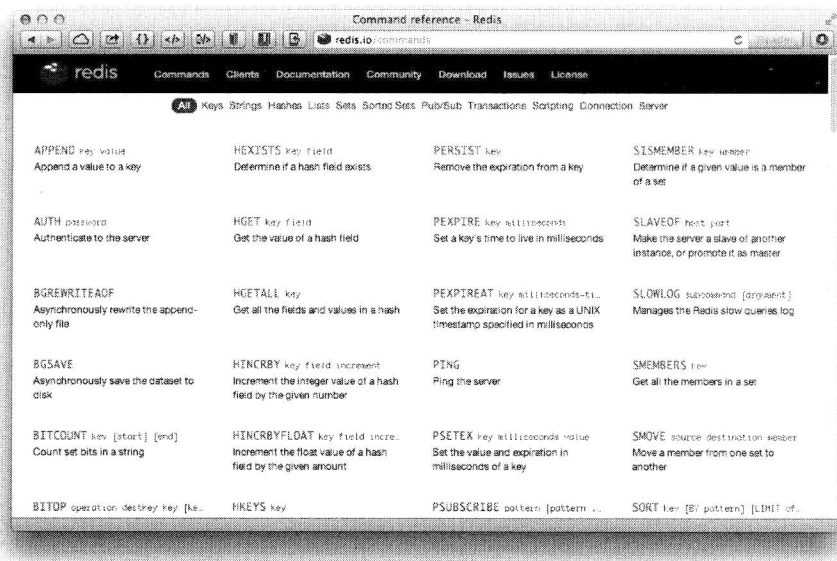


图 1-2 Redis 官网提供了详细的命令文档

第 2 章

准备

“纸上得来终觉浅，绝知此事要躬行。”

——陆游《冬夜读书示子聿》

学习 Redis 最好的办法就是动手尝试它。在介绍 Redis 最核心的内容之前，本章先来介绍一下如何安装和运行 Redis，以及 Redis 的基础知识，使读者可以在之后的章节中一边学习一边实践。

2.1 安装 Redis

安装 Redis 是开始 Redis 学习之旅的第一步。在安装 Redis 前需要了解 Redis 的版本规则以选择最适合自己的版本，Redis 约定次版本号（即第一个小数点后的数字）为偶数的版本是稳定版（如 2.4 版、2.6 版），奇数版本是非稳定版（如 2.5 版、2.7 版），推荐使用稳定版本进行开发和在生产环境使用。

2.1.1 在 POSIX 系统中安装

Redis 兼容大部分 POSIX 系统，包括 Linux、OS X 和 BSD 等，在这些系统中推荐直接下载 Redis 源代码编译安装以获得最新的稳定版本。Redis 最新稳定版本的源代码可以从地址 <http://download.redis.io/redis-stable.tar.gz> 下载。

下载安装包后解压即可使用 `make` 命令完成编译，完整的命令如下：

```
wget http://download.redis.io/redis-stable.tar.gz
```

```
tar xzf redis-stable.tar.gz
cd redis-stable
make
```

Redis 没有其他外部依赖，安装过程很简单。编译后在 Redis 源代码目录的 `src` 文件夹中可以找到若干个可执行程序，最好在编译后直接执行 `make install` 命令来将这些可执行程序复制到 `/usr/local/bin` 目录中以便以后执行程序时不用输入完整的路径。

在实际运行 Redis 前推荐使用 `make test` 命令测试 Redis 是否编译正确，尤其是在编译一个不稳定版本的 Redis 时。

提示 除了手工编译外，还可以使用操作系统中的软件包管理器来安装 Redis，但目前大多数软件包管理器中的 Redis 的版本都较古老。考虑到 Redis 的每次升级都提供了对以往版本的问题修复和性能提升，使用最新版本的 Redis 往往可以提供更加稳定的体验。如果希望享受包管理器带来的便利，在安装前请确认您使用的软件包管理器中 Redis 的版本并了解该版本与最新版之间的差异。<http://redis.io/topics/problems> 中列举了一些在以往版本中存在的已知问题。

2.1.2 在 OS X 系统中安装

OS X 下的软件包管理工具 Homebrew 和 MacPorts 均提供了较新版本的 Redis 包，所以我们可以直接使用它们来安装 Redis，省去了像其他 POSIX 系统那样需要手动编译的麻烦。下面以使用 Homebrew 安装 Redis 为例。

1. 安装 Homebrew

在终端下输入 `ruby -e "$(curl -fsSkL raw.githubusercontent.com/mxcl/homebrew/go)"` 即可安装 Homebrew。

如果之前安装过 Homebrew，请执行 `brew update` 来更新 Homebrew，以便安装较新版的 Redis。

2. 通过 Homebrew 安装 Redis

使用 `brew install` 软件包名可以安装相应的包，此处执行 `brew install redis` 来安装 Redis：

```
$ brew install redis
==> Downloading http://redis.googlecode.com/files/redis-2.6.9.tar.gz
Already downloaded: /Library/Caches/Homebrew/redis-2.6.9.tar.gz
==> make -C /private/tmp/redis-OV9u/redis-2.6.9/src CC=cc
==> Caveats
To have launchd start redis at login:
```



```
ln -sfv /usr/local/opt/redis/*.plist ~/Library/LaunchAgents
Then to load redis now:
  launchctl load ~/Library/LaunchAgents/homebrew.mxcl.redis.plist
Or, if you don't want/need launchctl, you can just run:
  redis-server /usr/local/etc/redis.conf
/usr/local/Cellar/redis/2.6.9: 9 files, 740K, built in 6 seconds
```

OS X 系统从 Tiger 版本开始引入了 `launchd` 工具来管理后台程序，如果想让 Redis 随系统自动运行可以通过以下命令配置 `launchd`：

```
ln -sfv /usr/local/opt/redis/*.plist ~/Library/LaunchAgents
launchctl load ~/Library/LaunchAgents/homebrew.mxcl.redis.plist
```

通过 `launchd` 运行的 Redis 会加载位于 `/usr/local/etc/redis.conf` 的配置文件，关于配置文件会在 2.4 节中介绍。

2.1.3 在 Windows 中安装

Redis 官方不支持 Windows。2011 年微软^①向 Redis 提交了一个补丁，以使 Redis 可以在 Windows 下编译运行，但被 Salvatore Sanfilippo 拒绝了，原因是在服务器领域上 Linux 已经得到了广泛的使用，让 Redis 能在 Windows 下运行相比而言显得不那么重要。并且 Redis 使用了如写时复制等很多操作系统相关的特性，兼容 Windows 会耗费太大的精力而影响 Redis 其他功能的开发。尽管如此微软还是发布了一个可以在 Windows 运行的 Redis 分支^②，但是考虑到其版本更新速度比较慢（截至本书交稿，其最新的版本是基于 Redis 2.4 进行开发的），并不建议使用。

如果想使用 Windows 学习或测试 Redis 可以通过 Cygwin 软件或虚拟机（如 VirtualBox）来完成。Cygwin 能够在 Windows 中模拟 Linux 系统环境。Cygwin 实现了一个 Linux API 接口，使得大部分 Linux 下的软件可以重新编译后在 Windows 下运行。Cygwin 还提供了自己的软件包管理工具，让用户能够方便地安装和升级几千个软件包。借助 Cygwin，我们可以在 Windows 上通过源代码编译安装最新版的 Redis。

1. 安装 Cygwin

从 Cygwin 官方网站（<http://cygwin.com>）下载 `setup.exe` 程序，`setup.exe` 既是 Cygwin 的安装包，又是 Cygwin 的软件包管理器。运行 `setup.exe` 后进入安装向导。前几步会要求选择下载源、安装路径、代理和下载镜像等，可以根据具体需求选择，一般来说一路单击“Next”即可。之后会出现软件包管理界面，如图 2-1 所示。

① 微软开放技术有限公司（Microsoft Open Technologies Inc.），专注于参与开源项目、开放标准工作组以及提出倡议。

② 见 <https://github.com/Microsoft/Redis>。