

# 五.Java基础 (Basic)

---

- [Java基础](#)
- [美团技术团队：深入解析String的intern\(\)方法](#)
- [Java本地方法](#)
- [Java中的Switch对整型、字符型、字符串型的具体实现细节](#)

## Java枚举

- [深入理解Java枚举类型\(enum\)](#)

## Java异常

- [Java 异常处理](#)
- [Java 中的异常和处理详解](#)
- [知乎：如何优雅地处理异常？](#)

## 序列化与反序列化

- [美团技术团队：序列化和反序列化](#)

## 反射

- [知乎：学习java应该如何理解反射？](#)
- [深入解析Java反射（1） - 基础](#)
- [深入解析Java反射（2） - invoke方法](#)
- [并发编程网：Java Reflection教程](#)
- [反射——Java高级开发必须懂的](#)

## 注解

- [全面解析Java注解](#)
- [并发编程网：Java注解教程及自定义注解](#)

## String、StringBuilder和StringBuffer

- [在java中String类为什么要设计成final？ - 知乎](#)
- [Java StringBuilder和StringBuffer源码分析](#)
- Stack Overflow: [String, StringBuffer, and StringBuilder](#)

## hashCode()和equals()

- [Java hashCode\(\) 和 equals\(\)的若干问题解答](#)
- [Java提高篇——equals\(\)与hashCode\(\)方法详解](#)
- [面试官爱问的equals与hashCode](#)

## Java IO

- [Java IO教程](#)
- [文件传输基础——Java IO流](#)
- [Java NIO 系列教程](#)
- [以Java的视角来聊聊BIO、NIO与AIO的区别？](#)

## Java容器 ==>> 推荐阅读《分布式Java应用 基础与实践》第四章

- [Java集合学习指南](#)
- [java 集合类知识汇总](#)
- [Java 容器](#)
- [ArrayList vs. LinkedList vs. Vector](#)
- [ArrayList、LinkedList、Vector、HashSet、TreeSet、HashMap、TreeMap的区别和适用场景](#)
- [Java Collections Framework - ArrayList](#)
- **HashMap与ConcurrentHashMap**
  - [占小狼：深入浅出ConcurrentHashMap1.8](#)
  - [占小狼：谈谈ConcurrentHashMap1.7和1.8的不同实现](#)
  - [占小狼：ConcurrentHashMap的红黑树实现分析](#)
  - [占小狼：老生常谈，HashMap的死循环](#)
  - [HashMap与ConcurrentHashMap在JDK1.7和JDK1.8中的实现](#)
  - [美团技术团队：Java 8系列之重新认识HashMap](#)
- **优先级队列**
  - [PriorityQueue](#)
  - [优先级队列是一种什么样的数据结构](#)
  - [堆排序和 PriorityQueue 源码解析](#)
  - [基于堆实现的优先级队列：PriorityQueue 解决 Top K 问题](#)
  - [jdk源码分析PriorityQueue](#)

## Java 8 ==>> 推荐阅读《Java 8实战》与《Java 8函数式编程》

- [What's New in JDK 8](#)
- [IBM: Java 8 新特性概述](#)
- [Java 8 特性 – 终极手册](#)
- [IBM: Java 8 中的 Streams API 详解](#)

### 值得思考的问题：

- [Java 8接口上的default method设计目的是什么？](#) # 六.Java并发（Concurrency）

### 推荐书籍：

- [《Java并发编程的艺术》](#)

z这本书比《Java并发编程实战》更适合入门一些。作者还创办了并发编程网，里面有很多值得读的文章。
- [《Java并发编程实战》](#)

jdk并发包作者写的书，书很好，但翻译为人诟病很久。
- [《实战Java高并发程序设计》](#)

在图书馆瞎逛的时候看到的一本书，翻了翻，出乎意料地不错，涉及到一些Java8函数式编程的知识。

## 并发合集

- [深入理解java内存模型系列文章](#)
- [深入浅出Java Concurrency](#)
- [死磕java并发](#)
- [Java 并发](#)
- [Java 并发知识合集](#)

## Java并发工具类之LongAdder

- [Java并发工具类之LongAdder原理总结](#)
- [Java8 Striped64 和 LongAdder](#)

## 线程池

- [深入分析java线程池的实现原理](#)
- [线程池的工作原理与源码解读](#)
- [Java并发编程：线程池的使用](#)
- [聊聊并发（三）——JAVA线程池的分析和使用](#)

## 阻塞队列

- [聊聊并发（七）——Java中的阻塞队列](#)
- [【死磕Java并发】-----J.U.C之阻塞队列：BlockingQueue总结](#)
- [【死磕Java并发】-----J.U.C之阻塞队列：ArrayBlockingQueue](#)
- [【死磕Java并发】-----J.U.C之阻塞队列：LinkedBlockingDeque](#)
- [【死磕Java并发】-----J.U.C之阻塞队列：PriorityBlockingQueue](#)
- [【死磕Java并发】-----J.U.C之阻塞队列：SynchronousQueue](#)
- [【死磕Java并发】-----J.U.C之阻塞队列：LinkedTransferQueue](#)

## synchronized和lock

- [【死磕Java并发】-----深入分析synchronized的实现原理](#)
- [深入浅出synchronized](#)
- [Lock与synchronized 的区别](#)
- [并发编程的锁机制：synchronized和lock](#)

## CAS

- [面试必问的CAS，要多了解](#)
- [乐观锁的一种实现方式——CAS](#)
- [【死磕Java并发】-----J.U.C之深入分析CAS](#)

## ThreadLocal

- [深入浅出ThreadLocal](#)
- [Java面试必问，ThreadLocal终极篇](#)
- [【死磕Java并发】——深入分析ThreadLocal](#)

## Java中的锁

- [Java中的锁](#)
- [【死磕Java并发】-----J.U.C之重入锁：ReentrantLock](#)
- [【死磕Java并发】-----J.U.C之读写锁：ReentrantReadWriteLock](#)

## Java秒杀

- [设计一个秒杀系统](#)
- [秒杀架构实践](#)
- [限流算法](#)
- [知乎：秒杀系统设计# 九.框架（Framework）](#)

## 1. Spring总览

---

- [Spring 框架的设计理念与设计模式分析](#)

## 2. IoC

---

### 2.1 IoC介绍

- [知乎：Spring IOC原理总结](#)
- [知乎：Spring IoC有什么好处呢？](#)

### 2.2 IoC容器

- BeanFactory
- ApplicationContext以及WebApplicationContext
- BeanFactory和ApplicationContext中各自Bean的生命周期
- Bean的注入方式和作用域

## AOP

- [Spring AOP 实现原理与 CGLIB 应用](#)
- 动态代理
  - [JDK动态代理代理与Cglib代理原理探究](#)
  - [Java Proxy 和 CGLIB 动态代理原理](#)

## 常用知识点

- [Spring常见面试题](#)
- [知乎：@Autowired和@Resource的区别是什么？](#)
- StringRedisTemplate 类
  - [Spring Doc: Class StringRedisTemplate](#)
  - [如何使用RedisTemplate访问Redis数据结构](#)
- RestTemplate类
  - [Spring Doc: Class RestTemplate](#)
  - [A Guide to the RestTemplate | Baeldung](#)
  - [Spring RestTemplate 中几种常见的请求方式](#)

## Spring Boot

---

- [@SpringBootApplication\(源码分析与启动流程\)](#) ==>> 推荐阅读《Spring Boot揭秘》第三章
- 参考视频教程
  - [SpringBoot开发常用技术整合](#)
  - [Spring Boot聚合工程](#)
  - [Spring Boot热部署](#)
- 参考文档以及开原作品索引
  - [Spring Boot 中文索引](#)

## Thymeleaf

- [Thymeleaf-Reference-Guide中文翻译](#)

## Spring MVC

---

- [Spring MVC中文文档](#)
- [SpringMVC深度探险系列](#)

## Servlet

- [知乎：servlet的本质是什么，它是如何工作的？](#)
- [许令波：Servlet 工作原理解析](#)
- [Servlet生命周期与工作原理](#)

## redirect和forward

- [知乎：web开发过程中redirect和forward有什么区别？](#)
- [请求转发（Forward）和重定向（Redirect）的区别](#)

## Spring MVC常用注解

- [springmvc常用注解标签详解](#)
- [Spring MVC 的常用注解](#)

## 面试相关

- [SpringMVC面试题](#)

## MyBatis

---

- [MyBatis Reference Documentation](#) ===> [中文文档](#)
- [美团技术团队：聊聊MyBatis缓存机制](#)
- [动态SQL](#)
- [MyBatis常见面试题](#)
- MyBatis中的#与\$的区别

## Spring Cloud

---

- 参考文档以及开源作品索引
  - [Spring Cloud 中文索引](#)

## 消息队列

---

RabbitMQ

RocketMQ

kafka

ActiveMQ+ [大话数据库连接池简史，你都用过几个？](#)

- [小白科普：Java EE vs J2EE vs Jakarta EE](#)
- [一个著名的日志系统是怎么设计出来的](#)

## 四.算法(Algorithm)

---

### 排序算法

- [常见排序算法及对应的时间复杂度和空间复杂度](#)
- [常用的八种排序算法Java代码实现](#)
- [图解排序算法\(一\)之3种简单排序\(选择，冒泡，直接插入\)](#)
- [图解排序算法\(二\)之希尔排序](#)
- [图解排序算法\(三\)之堆排序](#)
- [图解排序算法\(四\)之归并排序](#)
- [图解排序算法\(五\)之快速排序——三数取中法](#)

### 递归与尾递归

- [知乎：什么是尾递归？](#)

### 字典树 (Trie)

- [从Trie树（字典树）谈到后缀树](#)
- [Trie 三兄弟——标准 Trie、压缩 Trie、后缀 Trie](#)

### 红黑树

- [红黑树](#)
- [美团技术团队：红黑树深入剖析及Java实现](#)

### KMP

- [为什么java String.contains 没有使用类似KMP字符串匹配算法进行优化？](#)

### 布隆过滤器(Bloom Filter)

- [吴军：数学之美系列二十一 - 布隆过滤器 \(Bloom Filter\)](#)
- [使用BloomFilter布隆过滤器解决缓存击穿、垃圾邮件识别、集合判重](#)
- [布隆过滤器\(Bloom Filter\)的原理和实现](#)

### 一致性哈希

- [一致性哈希算法的理解与实践](#)
- [一致性HASH算法详解](#)

## JDK里的排序算法

- [Java提供的排序算法是怎么实现的?](#)
- [排序算法在jdk源码中的应用](#)
- [java.util.Arrays的排序研究](#)

## Huffman 编码

- [霍夫曼编码](#)
- [HUFFMAN 编码压缩算法](#)

## 并查集

- [并查集\(Union-Find\)算法介绍](#)
- [维基百科：并查集](#)

## 系统设计

---

- [系统设计面试之前需要知道的八件事](#)
- 设计Twitter
  - [如何设计 Twitter（第一部分）](#)
  - [如何设计 Twitter（第二部分）](#)
- [创建照片分享应用](#)
- [创建短网址系统](#)
- [如何设计 Google Docs](#)
- 设计新闻推送系统
  - [设计新闻推送系统（第一部分）](#)
  - [设计新闻推送系统（第二部分）](#)
- [设计 Facebook 聊天功能](#)
- [如何为 Twitter 设计趋势算法](#)
- [设计缓存系统](#)
- [设计推荐系统](#)
- [随机 ID 生成器](#)
- 设计键值存储
  - [设计键值存储（第一部分）](#)
  - [设计键值存储（第二部分）](#)
- [构建网页爬虫](#)
- 设计垃圾回收系统
  - [设计垃圾回收系统（第一部分）](#)
  - [设计垃圾回收系统（第二部分）](#)
- 设计电商网站
  - [设计电商网站（第一部分）](#)
  - [设计电商网站（第二部分）](#)
- [设计点击计数器](#)
- 设计 Youtube
  - [设计 Youtube（第一部分）](#)

- [设计 Youtube \(第二部分\)](#)

# 一.计算机网络 (Computer Networking)

## 推荐书籍

- [《图解HTTP》](#)

图解系列感觉都还不错，对新手很友好。还有一本老厚老厚[《HTTP权威指南》](#)，暂时还没有时间看。

- [《图解TCP/IP》](#)

如果对网络零基础，可以作为阅读以下几本书的铺垫。

- [《计算机网络：自顶向下方法》](#)

豆瓣上有人提供了本书的相应公开课,不过只看了书，公开课没看，[国立清华大学公开课地址](#)，[原贴地址](#)

- [《TCP/IP详解 卷1：协议》](#)

计算机网络的权威书籍。了解到作者生平故事的时候，总觉得英年早逝，真的很可惜。

- [《计算机网络》](#)

这本书其实是我最早看的关于计算机网络的书籍，是国内一些大学的教材，尽管自己也被国内很多教材荼毒过，但这本书还是不错的。

## 计算机网络总览

- [计算机网络](#)
- [HTTP](#)

## TCP、UDP

- [通俗大白话来理解TCP协议的三次握手和四次分手](#)
- [TCP 为什么是三次握手，而不是两次或四次？](#)
- [理解 TCP 和 UDP](#)
- [TCP长连接和心跳那些事](#)

## 计算机网络体系结构

- [OSI七层协议模型、TCP/IP四层模型和五层协议体系结构之间的关系](#)

## HTTP

- HTTP 请求方法
  - [MDN: HTTP 请求方法](#)
  - [w3school HTTP 方法: GET 对比 POST](#)
- HTTP 状态码
  - [w3school: HTTP 状态消息](#)
  - [MDN: HTTP response codes](#)
- HTTP 缓存
  - [MDN: HTTP 缓存 FAQ](#)
  - [google developers: HTTP 缓存](#)
  - [MDN: Cache-Control](#) ==>> [英文](#)
- HTTP内容协商



- [MDN: 内容协商](#) ==> [英文: Content negotiation](#)
- [谈谈 HTTP/2 的协议协商机制](#)
- **HTTP内容安全策略( CSP )**
  - [内容安全策略\( CSP \)](#) ==> [Content Security Policy \(CSP\)](#)
  - [阮一峰: Content Security Policy 入门教程](#)
- **HTTP的发展: 0.9/1.0/1.1/2.0**
  - [MDN: HTTP的发展](#)
  - [\[译\] HTTP 的进化 - 0.9、1.0、1.1、Keep-Alive、Upgrade 和 HTTPS](#)
  - [阮一峰: HTTP 协议入门](#)
- [HTTP/1.x 的连接管理](#)
- **Identifying resources on the Web**
  - [URI、URL、URN 的联系和区别](#)
  - [Identifying resources on the Web](#)
- **浏览器发送一次网址请求的过程**
  - [小白科普: 从输入网址到最后浏览器呈现页面内容, 中间发生了什么?](#)
  - [知乎: 当你在浏览器中输入 baidu.com 并且按下回车后发生了什么?](#)
  - [An attempt to answer the age old interview question "What happens when you type google.com into your browser and press enter?"](#)
- **代理**
  - [图解正向代理、反向代理、透明代理](#)

## SSH

- [SSH协议详解](#)
- [SSH 原理与运用 \(一\): 远程登录](#)
- [SSH原理与运用 \(二\): 远程操作与端口转发](#)

## SSL与TLS

- [阮一峰: SSL/TLS协议运行机制的概述](#)
- [阮一峰: 图解SSL/TLS协议](#)
- [维基百科: 传输层安全性协议](#)

## 集线器、交换机和路由器

- [路由器和交换机的不同之处有哪些?](#)
- [如何跟小白解释路由器和交换机的区别?](#)
- [集线器和交换机的区别?](#)

## 全双工、半双工和单工

- [串口通讯的单工、半双工和全双工的定义、区别及应用](#)

## CDN ==> 推荐阅读《深入分析Java Web技术内幕 修订版》相应章节

- [知乎: CDN是什么? 使用CDN有什么优势?](#)

## Cookie与Session ==>> 推荐阅读《深入分析Java Web技术内幕》相应章节

- [知乎: Cookie和Session有什么区别?](#)

## 加解密算法

- [AES算法](#)
- [SHA系列算法](#)
- [MD5算法](#)
- [如何破解MD5算法](#)
- [Base64算法](#)

# 八.设计模式 (Design Patterns)

- [设计模式合集](#)

## 为什么要创建这个仓库?

我在学习编程的时候，经常会搜索一些资料，但无论是Google还是百度，搜索出来的资料往往都不系统，资料的质量也参差不齐。直到现在，我都认为阅读书籍是最系统的学习方式。计算机科学的知识体系太庞大了，如果一些知识学过，却在之后一段时间不再触及，很容易就忘掉，但我们不可能时时刻刻都能随手拿到一本相应的书来作为参考，因此系统地整理一些自己看过的博文合集显得十分必要了。

随着仓库维护时间越来越长，我发现仓库的内容不仅仅对自己有用，对别人也有一定的参考价值。在拿了一些star后，我渐渐加上了一些自己学习心得，主要是关于一些自己读过的书籍。希望对你们有参考价值。

其实我觉得每个人都可以建一个这样的仓库，整理知识的过程也是一个学习的过程。通过反复阅读与复习，慢慢知识就成了自己的一部分。

## 阅读需知:

- 推荐使用[chrome插件:简悦](#)来阅读。
- 如果你在中国境内，部分文章可能需要科学上网才能阅读；当然也有可能是链接失效，你可以提交issue，感谢你的贡献。
- 仓库维护时间越久，文章目录越来越长，为了优化阅读体验，拆分了大章节，增加了每个大章节的链接。

							
计算机科学	Java	Web	架构	中间件	分布式	工具	未分类
<a href="#">计算机网络</a>	<a href="#">Java基础</a>	<a href="#">框架</a>	<a href="#">设计模式</a>	<a href="#">Zookeeper</a>	<a href="#">分布式基础理论</a>	<a href="#">IDEA系列教程</a>	视频 <a href="#">Git书籍《Pro Git》</a>
<a href="#">操作系统</a>	<a href="#">Java并发</a>		<a href="#">系统设计</a>	<a href="#">消息队列</a>		<a href="#">Postman系列教程</a>	<a href="#">Docker — 从入门到实践</a>
<a href="#">数据库</a>	<a href="#">JVM</a>		<a href="#">代码规范</a>	<a href="#">Web Server</a>			<a href="#">其他</a>
<a href="#">算法</a>	<a href="#">科普系列</a>			<a href="#">搜索引擎</a>			

# 一.计算机网络 (Computer Networking)

## 推荐书籍

- [《图解HTTP》](#)

图解系列感觉都还不错，对新手很友好。还有一本老厚老厚[《HTTP权威指南》](#)，暂时还没有时间看。

- [《图解TCP/IP》](#)

如果对网络零基础，可以作为阅读以下几本书的铺垫。

- [《计算机网络：自顶向下方法》](#)

豆瓣上有人提供了本书的相应公开课,不过只看了书，公开课没看，[国立清华大学公开课地址](#)，[原贴地址](#)

- [《TCP/IP详解 卷1：协议》](#)

计算机网络的权威书籍。了解到作者生平故事的时候，总觉得英年早逝，真的很可惜。

- [《计算机网络》](#)

这本书其实是我最早看的关于计算机网络的书籍，是国内一些大学的教材，尽管自己也被国内很多教材荼毒过，但这本书还是不错的。

## 计算机网络总览

- [计算机网络](#)
- [HTTP](#)

## TCP、UDP

- [通俗大白话来理解TCP协议的三次握手和四次分手](#)
- [TCP 为什么是三次握手，而不是两次或四次？](#)
- [理解 TCP 和 UDP](#)

## 计算机网络体系结构

- [OSI七层协议模型、TCP/IP四层模型和五层协议体系结构之间的关系](#)

## HTTP

- HTTP 请求方法
  - [MDN: HTTP 请求方法](#)
  - [w3school HTTP 方法: GET 对比 POST](#)
- HTTP 状态码
  - [w3school: HTTP 状态消息](#)
  - [MDN: HTTP response codes](#)
- HTTP 缓存
  - [MDN: HTTP 缓存 FAQ](#)
  - [google developers: HTTP 缓存](#)
  - [MDN: Cache-Control](#) ==>> [英文](#)
- HTTP内容协商

- [MDN: 内容协商](#) ==> [英文: Content negotiation](#)
  - [谈谈 HTTP/2 的协议协商机制](#)
- **HTTP内容安全策略( CSP )**
  - [内容安全策略\( CSP \)](#) ==> [Content Security Policy \(CSP\)](#)
  - [阮一峰: Content Security Policy 入门教程](#)
- **HTTP的发展: 0.9/1.0/1.1/2.0**
  - [MDN: HTTP的发展](#)
  - [\[译\] HTTP 的进化 - 0.9、1.0、1.1、Keep-Alive、Upgrade 和 HTTPS](#)
  - [阮一峰: HTTP 协议入门](#)
- [HTTP/1.x 的连接管理](#)
- **Identifying resources on the Web**
  - [URI、URL、URN 的联系和区别](#)
  - [Identifying resources on the Web](#)
- **浏览器发送一次网址请求的过程**
  - [小白科普: 从输入网址到最后浏览器呈现页面内容, 中间发生了什么?](#)
  - [知乎: 当你在浏览器中输入 baidu.com 并且按下回车后发生了什么?](#)
  - [An attempt to answer the age old interview question "What happens when you type google.com into your browser and press enter?"](#)
- **代理**
  - [图解正向代理、反向代理、透明代理](#)

## SSH

- [SSH协议详解](#)
- [SSH 原理与运用 \(一\) : 远程登录](#)
- [SSH原理与运用 \(二\) : 远程操作与端口转发](#)

## SSL与TLS

- [阮一峰: SSL/TLS协议运行机制的概述](#)
- [阮一峰: 图解SSL/TLS协议](#)
- [维基百科: 传输层安全性协议](#)

## 集线器、交换机和路由器

- [路由器和交换机的不同之处有哪些?](#)
- [如何跟小白解释路由器和交换机的区别?](#)
- [集线器和交换机的区别?](#)

## 全双工、半双工和单工

- [串口通讯的单工、半双工和全双工的定义、区别及应用](#)

## CDN

- [知乎: CDN是什么? 使用CDN有什么优势?](#)

## Cookie、Session、Token

- [知乎：Cookie和Session有什么区别？](#)
- [Cookie、Session、Token](#)

## 加解密算法

- [AES算法](#)
- [SHA系列算法](#)
- [MD5算法](#)
- [如何破解MD5算法](#)
- [Base64算法](#)

# 二.操作系统（Operating System）

---

## 操作系统基础知识(基于CSAPP)

---

### 基础知识

- [【不周山之读薄 CSAPP】零 系列概览](#)
- [壹 数据表示 - 不同的数据是如何存储与表示的](#)
- [贰 机器指令与程序优化 - 控制流、过程调用、跳转](#)
- [叁 内存与缓存 - 内存层级与缓存机制](#)
- [肆 链接 - 不同的代码如何协同](#)
- [伍 异常控制流 - 不同进程间的切换与沟通](#)
- [陆 系统输入输出 - 怎么把不同的内容发送到不同的地方](#)
- [柒 虚拟内存与动态内存分配 - 现代计算机中内存的奥秘](#)
- [捌 网络编程 - 从最原始套接字彻底理解网络编程](#)
- [玖 并行与同步 - 协同工作中最重要的两个问题](#)

### 实验部分

- [实验概览](#)
- [I Data Lab - 位操作，数据表示](#)
- [II Bomb Lab - 汇编，栈帧与 gdb](#)
- [III Attack Lab - 漏洞是如何被攻击的](#)
- [IV Cache Lab - 实现一个缓存系统来加速计算](#)
- [V Shell Lab - 实现一个 shell](#)
- [VI Malloc Lab - 实现一个动态内存分配](#)
- [VII Proxy Lab - 实现一个多线程带缓存的代理服务器](#)

## Linux

---

### Linux基础

- [《The Linux Command Line》](#)
  - [对应的中文在线版](#)
- [实验楼：Linux基础](#)
- [Linux命令实例练习](#)

- [《鸟哥的Linux私房菜在线阅读》](#)
- [Linux 命令大全](#)

## Linux shell

- [实验楼：高级 bash 脚本编程指南](#)
- [正则表达式基础](#)
- [《Linux Shell脚本攻略》](#)

## Vim

- [Vim编辑器基本使用](#)
- [陈皓：简明 VIM 练级攻略](#)

## Linux内核

# 三.数据库（Database）

---

## 数据库理论

---

### 事务

- [『浅入深出』MySQL 中事务的实现](#)
- [彻底理解数据库事务](#)
- [深入分析事务的隔离级别](#)
- [知乎：数据库事务原子性、一致性是怎样实现的？](#)
- [知乎：乐观锁和 MVCC 的区别？](#)
- [浅谈数据库并发控制 - 锁和 MVCC](#)
- [InnoDB存储引擎MVCC实现原理](#)

### 范式

- [解释一下关系数据库的第一第二第三范式？](#)

## MySQL

---

- [MySQL](#)

### MySQL索引

- [MYSQL-索引](#)
- [MySQL索引背后的数据结构及算法原理](#)

### MySQL优化

- [MySQL 对于千万级的大表要怎么优化？](#)
- [MYSQL性能优化的最佳20+条经验](#) ==> [英文原文:Top 20+ MySQL Best Practices](#)
- [MySQL 性能优化神器 Explain 使用分析](#)
- [视频：性能优化之MySQL优化](#)

## MySQL的锁

- [对mysql乐观锁、悲观锁、共享锁、排它锁、行锁、表锁概念的理解](#)
- [数据库的锁机制](#)
- [深入理解乐观锁与悲观锁](#)

## Redis

---

关于Redis的学习，命令与基础知识部分，Redis官网其实蛮详细了。

- [英文官网](#)
- [Redis中文网](#)

### 书籍推荐：

- [《Redis开发与运维》](#)

这本书主要是关于基础知识的讲解，比较全面，其实还有一本[《Redis实战》](#)，我翻了翻，没怎么细看。

- [《Redis设计与实现》](#)

Redis底层是用C语言写的，这本书剖析了Redis的内部源码实现，配图也很多，有很多经典巧妙的数据结构，我觉得是对算法学习很好的补充。

- [《Redis 深度历险：核心原理与应用实践》](#)

这本书有对应的小册，在掘金官网上可以购买，我只看了小册，不过书和小册应该基本一致。

### 基础知识

- [字符串 \(strings\)](#)
- [散列 \(hashes\)](#)
- [列表 \(lists\)](#)
- [集合 \(sets\)](#)
- [有序集合 \(sorted sets\)](#)
- [bitmaps](#)
- [hyperloglogs](#)
- [地理空间 \(geospatial\)](#)
- [复制 \(replication\)](#)
- [Lua脚本 \(Lua scripting\)](#)
- [LRU驱动事件 \(LRU eviction\)](#)
- [事务 \(transactions\)](#)
- [磁盘持久化 \(persistence\)](#)
- [Redis哨兵 \(Sentinel\)](#)
- [分区 \(Cluster\)](#)

### 底层源码解析

- [合集](#)
- [Redis源码解析](#)

## 分布式锁

- [点我达技术：Redis分布式锁进化史](#)

## 缓存

- [美团技术团队：缓存那些事](#)

# 四.算法(Algorithm)

---

## 排序算法

- [常见排序算法及对应的时间复杂度和空间复杂度](#)
- [常用的八种排序算法Java代码实现](#)
- [图解排序算法\(一\)之3种简单排序\(选择，冒泡，直接插入\)](#)
- [图解排序算法\(二\)之希尔排序](#)
- [图解排序算法\(三\)之堆排序](#)
- [图解排序算法\(四\)之归并排序](#)
- [图解排序算法\(五\)之快速排序——三数取中法](#)

## 递归与尾递归

- [知乎：什么是尾递归？](#)

## 字典树 (Trie)

- [从Trie树（字典树）谈到后缀树](#)
- [Trie 三兄弟——标准 Trie、压缩 Trie、后缀 Trie](#)

## 红黑树

- [红黑树](#)
- [美团技术团队：红黑树深入剖析及Java实现](#)

## KMP

- [为什么java String.contains 没有使用类似KMP字符串匹配算法进行优化？](#)

## 布隆过滤器(Bloom Filter)

- [吴军：数学之美系列二十一 - 布隆过滤器 \(Bloom Filter\)](#)
- [使用BloomFilter布隆过滤器解决缓存击穿、垃圾邮件识别、集合判重](#)
- [布隆过滤器\(Bloom Filter\)的原理和实现](#)

## 一致性哈希

- [一致性哈希算法的理解与实践](#)
- [一致性HASH算法详解](#)



## JDK里的排序算法

- [Java提供的排序算法是怎么实现的？](#)
- [排序算法在jdk源码中的应用](#)
- [java.util.Arrays的排序研究](#)

## Huffman 编码

- [霍夫曼编码](#)
- [HUFFMAN 编码压缩算法](#)

## 并查集

- [并查集\(Union-Find\)算法介绍](#)
- [维基百科：并查集](#)

## 动态规划

- [知乎：什么是动态规划？动态规划的意义是什么？](#)
- [漫画：什么是动态规划？](#)

## 贪心

- 

# 五.Java基础（Basic）

---

- [Java基础](#)
- [美团技术团队：深入解析String的intern\(\)方法](#)
- [Java本地方法](#)
- [Java中的Switch对整型、字符型、字符串型的具体实现细节](#)

## Java枚举

- [深入理解Java枚举类型\(enum\)](#)

## Java异常

- [Java 异常处理](#)
- [throw和throws有什么区别？](#)
- [Java 中的异常和处理详解](#)
- [知乎：如何优雅地处理异常？](#)

## 序列化与反序列化

- [美团技术团队：序列化和反序列化](#)

## 反射

- [知乎：学习java应该如何理解反射？](#)
- [深入解析Java反射（1） - 基础](#)
- [深入解析Java反射（2） - invoke方法](#)
- [并发编程网：Java Reflection教程](#)
- [反射——Java高级开发必须懂的](#)

## 注解

- [全面解析Java注解](#)
- [并发编程网：Java注解教程及自定义注解](#)

## String、StringBuilder和StringBuffer

- [在java中String类为什么要设计成final? - 知乎](#)
- [Java StringBuilder和StringBuffer源码分析](#)
- Stack Overflow: [String, StringBuffer, and StringBuilder](#)

## hashCode()和equals()

- [Java hashCode\(\) 和 equals\(\)的若干问题解答](#)
- [Java提高篇——equals\(\)与hashCode\(\)方法详解](#)
- [面试官爱问的equals与hashCode](#)

## Java IO

- [Java IO教程](#)
- [文件传输基础——Java IO流](#)
- [Java NIO 系列教程](#)
- [以Java的视角来聊聊BIO、NIO与AIO的区别?](#)

## Java容器 ==> 推荐阅读《分布式Java应用 基础与实践》第四章

- [Java集合学习指南](#)
- [java 集合类知识汇总](#)
- [Java 容器](#)
- [ArrayList vs. LinkedList vs. Vector](#)
- [ArrayList、LinkedList、Vector、HashSet、TreeSet、HashMap、TreeMap的区别和适用场景](#)
- [Java Collections Framework - ArrayList](#)
- **HashMap与ConcurrentHashMap**
  - [Java7/8 中的 HashMap 和 ConcurrentHashMap 全解析](#)
  - [占小狼：深入浅出ConcurrentHashMap1.8](#)
  - [占小狼：谈谈ConcurrentHashMap1.7和1.8的不同实现](#)
  - [占小狼：ConcurrentHashMap的红黑树实现分析](#)
  - [占小狼：老生常谈，HashMap的死循环](#)
  - [HashMap与ConcurrentHashMap在JDK1.7和JDK1.8中的实现](#)
  - [美团技术团队：Java 8系列之重新认识HashMap](#)
- **优先级队列**
  - [PriorityQueue](#)
  - [优先级队列是一种什么样的数据结构](#)
  - [堆排序和 PriorityQueue 源码解析](#)
  - [基于堆实现的优先级队列：PriorityQueue 解决 Top K 问题](#)
  - [jdk源码分析PriorityQueue\]](#)

## Java 8 ==>> 推荐阅读《Java 8实战》与《Java 8函数式编程》

- [What's New in JDK 8](#)
- [IBM: Java 8 新特性概述](#)
- [Java 8 特性 - 终极手册](#)
- [IBM:Java 8 中的 Streams API 详解](#)

### 值得思考的问题:

- [Java 8接口上的default method设计目的是什么?](#)

## 六.Java并发 (Concurrency)

---

### 推荐书籍:

- [《Java并发编程的艺术》](#)

z这本书比《Java并发编程实战》更适合入门一些。作者还创办了并发编程网，里面有很多值得读的文章。

- [《Java并发编程实战》](#)

jdk并发包作者写的书，书很好，但翻译为人诟病很久。

- [《实战Java高并发程序设计》](#)

在图书馆瞎逛的时候看到的一本书，翻了翻，出乎意料地不错，涉及到一些Java8函数式编程的知识。

### 并发合集

- [深入理解java内存模型系列文章](#)
- [深入浅出 Java Concurrency](#)
- [死磕Java并发](#)
- [Java 并发](#)
- [Java 并发知识合集](#)

### Java并发工具类之LongAdder

- [Java并发工具类之LongAdder原理总结](#)
- [Java8 Striped64 和 LongAdder](#)

### 线程池

- [深入分析java线程池的实现原理](#)
- [线程池的工作原理与源码解读](#)
- [Java并发编程：线程池的使用](#)
- [聊聊并发（三）——JAVA线程池的分析和使用](#)

### 阻塞队列

- [聊聊并发（七）——Java中的阻塞队列](#)
- [【死磕Java并发】-----J.U.C之阻塞队列：BlockingQueue总结](#)
- [【死磕Java并发】-----J.U.C之阻塞队列：ArrayBlockingQueue](#)
- [【死磕Java并发】-----J.U.C之阻塞队列：LinkedBlockingDeque](#)
- [【死磕Java并发】-----J.U.C之阻塞队列：PriorityBlockingQueue](#)
- [【死磕Java并发】-----J.U.C之阻塞队列：SynchronousQueue](#)

- [【死磕Java并发】-----J.U.C之阻塞队列：LinkedTransferQueue](#)

## synchronized和lock

- [【死磕Java并发】-----深入分析synchronized的实现原理](#)
- [深入浅出synchronized](#)
- [Lock与synchronized 的区别](#)
- [并发编程的锁机制：synchronized和lock](#)

## CAS

- [面试必问的CAS，要多了解](#)
- [乐观锁的一种实现方式——CAS](#)
- [【死磕Java并发】-----J.U.C之深入分析CAS](#)

## ThreadLocal

- [深入浅出ThreadLocal](#)
- [Java面试必问，ThreadLocal终极篇](#)
- [【死磕Java并发】——深入分析ThreadLocal](#)

## Java中的锁

- [Java中的锁](#)
- [【死磕Java并发】-----J.U.C之重入锁：ReentrantLock](#)
- [【死磕Java并发】-----J.U.C之读写锁：ReentrantReadWriteLock](#)

## Java秒杀

- [设计一个秒杀系统](#)
- [秒杀架构实践](#)
- [限流算法](#)
- [知乎：秒杀系统设计](#)

# 七.Java虚拟机 (JVM)

---

## 推荐书籍

- [《深入理解Java虚拟机》](#)

学过JVM的应该都看过这本书，比较详细，但这本书第二版是2013年出的，HotSpot在具体实现上有了一些变化，比如说内存结构Perm区的继任者Metaspace，以及一些新的垃圾收集器也出现了，这部分内容需要再自己选择性地去补充学习。

- [《实战Java虚拟机》](#)

相比较上一本，这一本书感觉更基础，两者可以相互补充。

## 合集

- [JVM一览](#)
- [Jvm知识汇总篇](#)

## JVM内存结构

- [JAVA8: 永久代\(PermGen\)和元空间\(Metaspace\)](#)
- [JVM内存结构 VS Java内存模型 VS Java对象模型](#)
- [JVM内存的那些事](#)

## 垃圾收集 (GC)

- [深入理解 java 垃圾回收机制](#)
- [不同的垃圾回收器的比较](#)
- [了解CMS\(Concurrent Mark-Sweep\)垃圾回收器](#)
- [并发编程网：深入理解G1垃圾收集器](#)
- [占小狼：G1垃圾收集器介绍](#)
- [占小狼：Java GC的那些事 \(1\)](#)
- [占小狼：Java GC的那些事 \(2\)](#)
- [Java GC系列 \(1\)：Java垃圾回收简介](#)
- [Java GC系列 \(2\)：Java垃圾回收是如何工作的？](#)
- [Java GC系列 \(3\)：垃圾回收器种类](#)
- [Java GC系列 \(4\)：垃圾回收监视和分析](#)
- [知乎：怎么在面试时回答Java垃圾回收机制 \(GC\) 相关问题？](#)
- [美团技术团队：Java Hotspot G1 GC的一些关键技术](#)

## 类加载

- [深入探讨 Java 类加载器](#)
- [深入分析Java ClassLoader原理](#)

## 性能调优

- [并发编程网：JVM实用参数系列](#)
- [美团技术团队：从实际案例聊聊Java应用的GC优化](#)

# 八.架构与设计

---

## 设计模式

---

- [设计模式合集](#)

## 系统设计

---

- [系统设计面试之前需要知道的八件事](#)
- 设计Twitter
  - [如何设计 Twitter \(第一部分\)](#)
  - [如何设计 Twitter \(第二部分\)](#)
- [创建照片分享应用](#)
- [创建短网址系统](#)
- [如何设计 Google Docs](#)
- 设计新闻推送系统
  - [设计新闻推送系统 \(第一部分\)](#)

- [设计新闻推送系统（第二部分）](#)
- [设计 Facebook 聊天功能](#)
- [如何为 Twitter 设计趋势算法](#)
- [设计缓存系统](#)
- [设计推荐系统](#)
- [随机 ID 生成器](#)
- 设计键值存储
  - [设计键值存储（第一部分）](#)
  - [设计键值存储（第二部分）](#)
- [构建网页爬虫](#)
- 设计垃圾回收系统
  - [设计垃圾回收系统（第一部分）](#)
  - [设计垃圾回收系统（第二部分）](#)
- 设计电商网站
  - [设计电商网站（第一部分）](#)
  - [设计电商网站（第二部分）](#)
- [设计点击计数器](#)
- 设计 Youtube
  - [设计 Youtube（第一部分）](#)
  - [设计 Youtube（第二部分）](#)

## 九.框架（Framework）

---

### Spring

---

- [Spring 框架的设计理念与设计模式分析](#)

### IoC

- [知乎：Spring IOC原理总结](#)
- [知乎：Spring IoC有什么好处呢？](#)
- BeanFactory、ApplicationContext以及WebApplicationContext
- BeanFactory和ApplicationContext中各自Bean的生命周期
- Bean的注入方式和作用域

### AOP

- [Spring AOP 实现原理与 CGLIB 应用](#)
- [探秘Spring AOP](#)
- 动态代理
  - [JDK动态代理代理与Cglib代理原理探究](#)
  - [Java Proxy 和 CGLIB 动态代理原理](#)

## 常用知识点

- [Spring常见面试题](#)
- [知乎：@Autowired和@Resource的区别是什么？](#)
- StringRedisTemplate 类
  - [Spring Doc: Class StringRedisTemplate](#)
  - [如何使用RedisTemplate访问Redis数据结构](#)
- RestTemplate类
  - [Spring Doc: Class RestTemplate](#)
  - [A Guide to the RestTemplate | Baeldung](#)
  - [Spring RestTemplate 中几种常见的请求方式](#)
- EntityManager
  - [EntityManager相关核心概念](#)
  - [EntityManager相关](#)

## Spring源码

- [死磕Spring](#)
- [Spring源码阅读](#)
- [Spring 实现原理与源码解析系统](#)

## Spring Boot

---

- [@SpringBootApplication\(源码分析与启动流程\)](#) ==>> 推荐阅读《Spring Boot揭秘》第三章
- 参考视频教程
  - [SpringBoot开发常用技术整合](#)
  - [Spring Boot聚合工程](#)
  - [Spring Boot热部署](#)
- 参考文档以及开原作品索引
  - [Spring Boot 中文索引](#)

## Thymeleaf

- [Thymeleaf-Reference-Guide中文翻译](#)

## Spring MVC

---

- [SpringMVC深度探险系列](#)

## Servlet

- [知乎：servlet的本质是什么，它是如何工作的？](#)
- [许令波：Servlet 工作原理解析](#)
- [Servlet生命周期与工作原理](#)

## redirect和forward

- [知乎：web开发过程中redirect和forward有什么区别？](#)
- [请求转发（Forward）和重定向（Redirect）的区别](#)

## Spring MVC常用注解

- [springmvc常用注解标签详解](#)
- [Spring MVC 的常用注解](#)

## 面试相关

- [SpringMVC面试题](#)

## MyBatis

---

- [MyBatis Reference Documentation](#) ===> [中文文档](#)
- [美团技术团队：聊聊MyBatis缓存机制](#)
- [动态SQL](#)
- [MyBatis常见面试题](#)
- MyBatis中的#与\$的区别

## Spring Cloud

---

- 参考文档以及开源作品索引
  - [Spring Cloud 中文索引](#)

# 十.中间件

---

## 1.Web Server

---

Tomcat

Nginx

## 2.消息队列

---

RabbitMQ

RocketMQ

kafka

ActiveMQ

# 十一.分布式

---

## 1.分布式基础理论

---

- [美团技术团队：分布式系统互斥性与幂等性问题的分析与解决](#)
- [分布式基础理论](#)
- CAP理论：
  - 论文：[Brewer's Conjecture and the Feasibility of Consistent, Available, Partition-Tolerant Web](#)
  - [CAP 定理的含义](#)



- [分布式理论\(一\) - CAP定理](#)
- **BASE理论:**
  - [分布式理论\(二\) - BASE理论](#)
- **2PC协议:**
  - [分布式理论\(三\) - 2PC协议](#)
- **3PC协议:**
  - [分布式理论\(四\) - 3PC协议](#)
- **一致性算法Paxos:**
  - [分布式理论\(五\) - 一致性算法Paxos](#)
- **一致性协议Raft:**
  - [分布式理论\(六\) - 一致性协议Raft](#)
- **分布式事务**
  - [事务相关基础知识](#)
  - [分布式事务](#)
  - [IBM developerWorks : JTA 深度历险 - 原理与实现](#)
- **happens-before**
  - 论文: [Time, Clocks, and the Ordering of Events in a Distributed System](#)

## 十二.搜索引擎

---

### Elasticsearch

---

### Apache Solr

---

## 十二.其他

---

### IDEA

---

- [IntelliJ-IDEA-Tutorial](#)
- [IntelliJ IDEA神器使用技巧](#)

### Postman

---

- [Postman系列教程](#)

### 压力测试

---

- Apache bench
- JMeter

### Swagger

---

### Git

---

- [使用git提交到github,每次都要输入用户名和密码的解决方法](#)
- [GitHub & Git](#)
- [《Pro Git》](#)

## 任务调度

---

- [Java定时任务调度工具详解之Timer篇](#)
- [Java定时任务调度工具详解之Quartz篇](#)
- [Quartz官方文档中文版](#)
- [Quartz Documentation](#)

## Docker

---

- [Docker入门教程](#)
- [视频：Docker入门](#)

## RESTful

---

- [怎样用通俗的语言解释REST，以及RESTful?](#)

## 杂

---

- [美团技术团队：常见性能优化策略的总结](#)
- [美团技术团队：性能优化模式](#)

## 七.Java虚拟机 (JVM)

---

### 推荐书籍

- [《深入理解Java虚拟机》](#)

学过JVM的应该都看过这本书，比较详细，但这本书第二版是2013年出的，HotSpot在具体实现上有了一些变化，比如说内存结构Perm区的继任者Metaspace，以及一些新的垃圾收集器也出现了，这部分内容需要再自己选择性地去补充学习。

- [《实战Java虚拟机》](#)

相比较上一本，这一本书感觉更基础，两者可以相互补充。

### 合集

- [JVM一览](#)
- [Jvm知识汇总篇](#)

### JVM内存结构

- [JAVA8：永久代\(PermGen\)和元空间\(Metaspace\)](#)
- [JVM内存结构 VS Java内存模型 VS Java对象模型](#)
- [JVM内存的那些事](#)

### 垃圾收集 (GC)

- [深入理解java 垃圾回收机制](#)
- [不同的垃圾回收器的比较](#)
- [了解CMS\(Concurrent Mark-Sweep\)垃圾回收器](#)
- [并发编程网：深入理解G1垃圾收集器](#)
- [占小狼：G1垃圾收集器介绍](#)
- [占小狼：Java GC的那些事 \(1\)](#)
- [占小狼：Java GC的那些事 \(2\)](#)

- [Java GC系列（1）：Java垃圾回收简介](#)
- [Java GC系列（2）：Java垃圾回收是如何工作的？](#)
- [Java GC系列（3）：垃圾回收器种类](#)
- [Java GC系列（4）：垃圾回收监视和分析](#)
- [知乎：怎么在面试时回答Java垃圾回收机制（GC）相关问题？](#)
- [美团技术团队：Java Hotspot G1 GC的一些关键技术](#)

## 类加载

- [深入探讨Java类加载器](#)
- [深入分析Java ClassLoader原理](#)

## 性能调优

- [并发编程网：JVM实用参数系列](#)
- [美团技术团队：从实际案例聊聊Java应用的GC优化](#)
- [JVM发生OOM的8种原因、及解决办法](#)
- [JVM调优（学习篇）](#)

## Web Server

---

### Tomcat

### Nginx

- [Zookeeper教程](#)
- [gitbook: Zookeeper深入浅出](#)
- [《从Paxos到Zookeeper》](#)

## 十二.其他

---

### IDEA

---

- [IntelliJ-IDEA-Tutorial](#)
- [IntelliJ IDEA神器使用技巧](#)

### Postman

---

- [Postman系列教程](#)

### 压力测试

- Apache bench
- JMeter

### Swagger

---

### Git

---

- [使用git提交到github,每次都要输入用户名和密码的解决方法](#)
- [GitHub & Git](#)
- [《Pro Git》](#)

## 任务调度

---

- [Java定时任务调度工具详解之Timer篇](#)
- [Java定时任务调度工具详解之Quartz篇](#)
- [Quartz官方文档中文版](#)
- [Quartz Documentation](#)

## Docker

---

- [Docker入门教程](#)
- [视频: Docker入门](#)

## RESTful

---

- [怎样用通俗的语言解释REST, 以及RESTful?](#)

## 杂

---

- [美团技术团队: 常见性能优化策略的总结](#)
- [美团技术团队: 性能优化模式](#)

# 十一.分布式

---

## 1.分布式基础理论

---

- [美团技术团队: 分布式系统互斥性与幂等性问题的分析与解决](#)
- [分布式基础理论](#)
- **CAP理论:**
  - 论文: [Brewer's Conjecture and the Feasibility of Consistent, Available, Partition-Tolerant Web](#)
  - [CAP 定理的含义](#)
  - [分布式理论\(一\) - CAP定理](#)
- **BASE理论:**
  - [分布式理论\(二\) - BASE理论](#)
- **2PC协议:**
  - [分布式理论\(三\) - 2PC协议](#)
- **3PC协议:**
  - [分布式理论\(四\) - 3PC协议](#)
- **一致性算法Paxos:**
  - [分布式理论\(五\) - 一致性算法Paxos](#)
- **一致性协议Raft:**
  - [分布式理论\(六\) - 一致性协议Raft](#)
- **分布式事务**
  - [事务相关基础知识](#)
  - [分布式事务](#)
  - [IBM developerWorks : JTA 深度历险 - 原理与实现](#)
  - [深入理解「分布式事务」](#)
- **happens-before**

- 论文: [Time, Clocks, and the Ordering of Events in a Distributed System](#)

## 十二.搜索引擎

---

### Elasticsearch

---

### Apache Solr# 二.操作系统 (Operating System)

---

### 操作系统基础知识(基于CSAPP)

---

#### 基础知识

- [【不周山之读薄 CSAPP】零 系列概览](#)
- [壹 数据表示 - 不同的数据是如何存储与表示的](#)
- [贰 机器指令与程序优化 - 控制流、过程调用、跳转](#)
- [叁 内存与缓存 - 内存层级与缓存机制](#)
- [肆 链接 - 不同的代码如何协同](#)
- [伍 异常控制流 - 不同进程间的切换与沟通](#)
- [陆 系统输入输出 - 怎么把不同的内容发送到不同的地方](#)
- [柒 虚拟内存与动态内存分配 - 现代计算机中内存的奥秘](#)
- [捌 网络编程 - 从最原始套接字彻底理解网络编程](#)
- [玖 并行与同步 - 协同工作中最重要的两个问题](#)

#### 实验部分

- [实验概览](#)
- [I Data Lab - 位操作, 数据表示](#)
- [II Bomb Lab - 汇编, 栈帧与 gdb](#)
- [III Attack Lab - 漏洞是如何被攻击的](#)
- [IV Cache Lab - 实现一个缓存系统来加速计算](#)
- [V Shell Lab - 实现一个 shell](#)
- [VI Malloc Lab - 实现一个动态内存分配](#)
- [VII Proxy Lab - 实现一个多线程带缓存的代理服务器](#)

## Linux

---

### Linux基础

- [《The Linux Command Line》](#)
  - [对应的中文在线版](#)
- [实验楼: Linux基础](#)
- [Linux命令实例练习](#)
- [《鸟哥的Linux私房菜在线阅读》](#)
- [Linux 命令大全](#)

## Linux shell

- [实验楼：高级 bash 脚本编程指南](#)
- [正则表达式基础](#)
- [《Linux Shell脚本攻略》](#)

## Vim

- [Vim编辑器基本使用](#)
- [陈皓：简明 VIM 练级攻略](#)

## Linux内核# 三.数据库 (Database)

# 数据库理论

---

## 事务

- [『浅入深出』MySQL 中事务的实现](#)
- [彻底理解数据库事务](#)
- [深入分析事务的隔离级别](#)
- [知乎：数据库事务原子性、一致性是怎样实现的？](#)
- [知乎：乐观锁和 MVCC 的区别？](#)
- [浅谈数据库并发控制 - 锁和 MVCC](#)
- [InnoDB存储引擎MVCC实现原理](#)

## 范式

- [解释一下关系数据库的第一第二第三范式？](#)

# MySQL

---

- [MySQL](#)

## MySQL索引

- [MYSQL-索引](#)
- [MySQL索引背后的数据结构及算法原理](#)

## MySQL优化

- [MySQL 对于千万级的大表要怎么优化？](#)
- [MYSQL性能优化的最佳20+条经验](#) ===> [英文原文:Top 20+ MySQL Best Practices](#)
- [MySQL 性能优化神器 Explain 使用分析](#)
- [SQL优化方案](#)

## MySQL的锁

- [对mysql乐观锁、悲观锁、共享锁、排它锁、行锁、表锁概念的理解](#)
- [数据库的锁机制](#)
- [深入理解乐观锁与悲观锁](#)

# Redis

---

关于Redis的学习，命令与基础知识部分，Redis官网其实蛮详细了。

- [英文官网](#)
- [Redis中文网](#)

## 书籍推荐：

- [《Redis开发与运维》](#)

这本书主要是关于基础知识的讲解，比较全面，其实还有一本[《Redis实战》](#)，我翻了翻，没怎么细看。

- [《Redis设计与实现》](#)

Redis底层是用C语言写的，这本书剖析了Redis的内部源码实现，配图也很多，有很多经典巧妙的数据结构，我觉得是对算法学习很好的补充。

- [《Redis 深度历险：核心原理与应用实践》](#)

这本书有对应的小册，在掘金官网上可以购买，我只看了小册，不过书和小册应该基本一致。

## 基础知识

- [字符串 \(strings\)](#)
- [散列 \(hashes\)](#)
- [列表 \(lists\)](#)
- [集合 \(sets\)](#)
- [有序集合 \(sorted sets\)](#)
- [bitmaps](#)
- [hyperloglogs](#)
- [地理空间 \(geospatial\)](#)
- [复制 \(replication\)](#)
- [Lua脚本 \(Lua scripting\)](#)
- [LRU驱动事件 \(LRU eviction\)](#)
- [事务 \(transactions\)](#)
- [磁盘持久化 \(persistence\)](#)
- [Redis哨兵 \(Sentinel\)](#)
- [分区 \(Cluster\)](#)

## 底层源码解析

- [合集](#)
- [Redis源码解析](#)

## 分布式锁

- [点我达技术：Redis分布式锁进化史](#)

## 缓存

- [美团技术团队：缓存那些事](#)

