如何系统学习网络编程（2024年最新）

有什么推荐的项目。

尤其是学习 C++ 的同学，因为学习 Java 的同学不少是有现成的框架做一些 Web 项目。

所以在这给大家推荐一类项目：网络编程。

在这分享一下我学习过程中整理的书籍、面经、博客等，不是在网上那种打包下载的，而是自己需要学到某个方向知识的时候，去挨个找的，最后汇总而成。

有需自取: 计算机必看经典书单(含下载方式)

#

一、什么是网络编程

说白了，就是 Socket 编程，任何语言都一样，使用语言自带的 Socket API 即可。

其实现在绝大多数应用、浏览器基本都是 C/S 模式，也就是一个客户端，一个服务端。

最常见的是通过 HTTP 协议进行客户端和服务端的数据传输，需要更高效数据传输或者自定义协议的时候一般就会使用 socket 编程。

其实 socket 编程编码起来非常简单，就跟读写文件差不多，双方通过 listen、connect、accept 等函数建立一个连接，也可以叫做文件描述符 sockfd。

接下来就可以使用 write、read 等文件读写函数对其进行操作，典型的 TCP 编程流程如下:

图源: 趣谈Linux操作系统

可以说网络编程是互联网的基石，不过实际工作中大多数同学是接触不到网络编程这一层的，因为都是现成的 RPC 框架。

框架已经帮我们做了协议解析、网络数据传输、解封包这些底层操作。

但是我觉得网络编程技能还是很重要的，大家可以在学校期间学习掌握。

#

二、学习资料

#

2.1 网络编程入门

接下来给大家推荐一些不错的学习路线和资料：

首先，建议不要直接上来就去看《Unix网络编程》，我知道这本书被誉为网络编程圣经，甚至 UNP 和 APUE 被称为鹅厂敲门砖。

这本书虽然非常经典，但是由于细节太多，太厚，需要时间比较长，所以不建议初学者一开始就去看。

先去随便找菜鸟教程或者 man 手册熟悉常用的 socket api，用自己熟悉的编程语言，Java、C、Python 等，先写几个网络聊天室、文件传输这种 demo，然后再去深入学习细节。

#

2.2 网络编程重点知识

Linux 下网络编程核心的包括系统编程和网络 IO 两个部分：

进程间通信方式： 信号量、管道、共享内存、socket 等

多线程编程：互斥锁、条件变量、读写锁、线程池等

五大 IO 模型：同步、异步、阻塞、非阻塞、信号驱动

高性能 IO 两种模式：Reactor 和 Proactor（ Linux 下由于缺少异步 IO 支持，基本没有 Proactor

IO 复用机制：epoll、select、poll（ C10K 问题）

#

2.3 推荐的书

#

《Unix网络编程》

#

《Unix环境高级编程》

这两本是砖头书，虽然是网络编程和 Unix 系统编程方面的无出其右的圣经，但主要用途还是垫显示器（逃

#

《Linux高性能服务器编程》

我强烈推荐，这本书前半部分基本是在重复计网基础知识，但是后面几章关于高性能服务器程序框架、高性能IO、IO复用、定时器、多线程编程、线程池和进程池还是讲得非常全面到位的，值得一看，看完基本上对于整个网络编程就有了框架。

#

《Linux多线程服务器端编程》

这本书同样强烈推荐，这是陈硕大佬写的书，说实话第一部分：C++ 多线程系统编程都直接把我看蒙了，没有想到 C++ 里要做到线程安全这么难，第一章我看了两三遍才看懂吧。。。 这是难得的讲解 C++ 多线程编程的书。

并且在书中，陈硕大佬用了一章讲解了 Muduo 网络库设计与实现，Muduo 比较适合学完基础的网络编程后继续进阶学习如何设计和写一个网络库，是一个高质量的 Reactor 网络库，采用 one loop per thread + thread pool 实现，代码比较简洁，书和源码搭配着看作为学习网络编程方面来说是非常不错。

#

三、推荐的项目

学完网络编程就可以写点小项目练手了，这里列举几个项目：

#

3.1 HTTP 服务器

这个似乎成了 Linux C/C++ 人手一个的项目了？ 这里推荐一个做为参考： https://github.com/imarvinle/WebServer

(opens new window)

Web Server 可以扩展写的地方还是挺多的，比如可以加入代理功能，这部分我在留学生 lab 中写过，但是没有集成到这个里面来，可以加入日志库，可以添加 CGI 支持等等。

#

3.2 网络库

这个也算是造轮子了， 可以就采用 one loop per thread + thread pool 这种模式，建议先去看懂 Muduo 源码，然后自己再写一个类似的，这个过程就算是抄，你也可以学到不少东西的，先看，再模仿、修改。

地址: https://github.com/chenshuo/muduo

(opens new window)

#

3.3 RPC

实现一个 PRC 需要考虑到序列化、网络传输、服务发现等，比较有名的有 grpc、brpc 等。

brpc 是一个基于 protobuf 接口的 RPC 框架，现在已经进入顶级开源社区 Apache。

它的官方仓库文档就非常完善了：

小而美的 RPC 框架也可以看下 搜狗 RPC：https://github.com/sogou/srpc

(opens new window)

这个底层基于搜狗另外一个开源项目： https://github.com/sogou/workflow

(opens new window)

并且代码量都不是特别大，基本在万行左右，适合个人学习。

#

3.4 网络聊天室

简单版的就可以直接在局域网内实现群聊、单聊等。

更进一步可以考虑一下如何不通过服务器中转消息实现 P2P 聊天，类似 QQ，这里会涉及到 UDP 打洞、NAT 转换等知识，还是很有意思的，我大二用 Java 搞过。

不管 Java 还是 C++，这些基础的网络编程概念都是共通的。

#

3.5 联机游戏

其实就是将一些单机小游戏考虑做成联机的，我当时是做的联机版五指棋，关键就是将游戏数据、状态通过网络进行同步。 比如 联机坦克大战，这些做起来应该会挺有意思的。

作者: 编程指北

链接: https://csguide.cn/roadmap/basic/network\_programming\_learning\_path.html#%E4%B8%89%E3%80%81%E6%8E%A8%E8%8D%90%E7%9A%84%E9%A1%B9%E7%9B%AE

来源: https://csguide.cn

著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。