



# 西南科技大学

Southwest University of Science and Technology

信息工程学院

## 实 习 报 告

课 程 名 称	物联网工程专业毕业实习
学 生 姓 名	李 钦 源
学 号	5120164270
专 业 班 级	物联 1603
实 习 地 点	西南科技大学
实 习 时 间	2020.4.1-2020.5.1

西南科技大学信息工程学院制

# 西南科技大学信息工程学院

## 《物联网工程专业毕业实习》评分表（2020）

评定指标	评定项目	评定标准	考核等级	分值范围	得分
实习过程 (50分)	职业素养 实习态度	严格遵守实习纪律，无迟到、早退等现象，认真听讲，虚心学习，做好实习笔记，能学会理解和评价物联网工程实践对环境、社会可持续发展的影响，能认真维护团队利益，集体意识强。	合格	12—20 分	19
		实习不认真，有缺席现象。	不合格	小于 12 分	
	实习任务 自主学习 完成情况	能积极主动了解熟悉相关技术、法律法规、企业质量保障体系，能独立思考，提出问题并展开讨论	合格	18—30 分	28
		实习中敷衍了事，对产品、技术、标准知之甚少，缺乏自主学习。	不合格	小于 18 分	
实习报告 (50分)	实习报告 内 容	实习报告内容详实，有自己的独立思考，能主动提出问题并进行分析，能认识所见企业产品的生产加工对社会、环境等方面的影响。	优秀	30—35 分	33
		实习报告内容比较详实，能对问题进行分析总结，能认识所见企业产品的生产加工对社会、环境等方面的影响。	良好	25—30 分	
		实习报告内容较为完整，能认识所见企业产品的生产加工对社会、环境等方面的影响。	中	20—25 分	
		实习报告内容不完整，也没有自己的思考。	差	小于 20 分	
	实习报告 撰写质量	实习报告结构合理，表达清晰，书写工整，有自己独立思考。	优秀	10—15 分	13
		表达较清晰，书写较工整。	一般	5—10 分	
		实习报告内容欠详实。	差	小于 5 分	
过程指导导师评分老师签名：詹曦			总分	93	
2020 年 5 月 10 日					

## 一、实习目的意义

代理服务器是一种介于客户端和服务端之间充当信息中转的服务器，而 HTTP 代理服务器则是介于 web 浏览器和 HTTP 服务器之间的服务器。目前，HTTP 代理服务器是 Internet 链路级网关所提供的一种重要安全功能，具有突破 IP 访问限制、提高访问速度、用户权限验证以及自我保护等功能。因此，实现代理服务器对加强网络安全，突破网络封锁均具有重要意义。通过实现一个简易 HTTP 代理服务器的设计，一方面可以加强对代理服务器及其工作原理的理解，另一方面也可以锻炼网络编程能力，为实现更加完善和复杂的代理服务器奠定基础。本次实习内容就是在指导老师要求下设计出一款具有基本功能的 HTTP 代理服务器。

具体任务内容有：

1. 熟悉 HTTP 代理服务器的工作原理及相关 RFC 文档。
2. 对 HTTP 代理服务器的基本功能进行需求分析和设计。
3. 使用 C++ 程序设计语言编程实现 HTTP 代理服务器程序。
4. 对 3 中实现的 HTTP 代理服务器进行测试。

## 二、实习计划

1. 完成 HTTP 代理服务器基本框架：2020 年 4 月 1 号至 2020 年 4 月 5 号
2. 编写 HTTP 代理服务器程序：2020 年 4 月 5 号至 2020 年 4 月 25 号
3. 调试功能与 Debug：2020 年 4 月 25 号至 2020 年 5 月 1 号

### 三、实习日志

实习日志			
序号	日期	工作内容	完成情况
1	2020 年 4 月 1 号	确定工作内容	已完成
2	2020 年 4 月 4 号	安装环境	已完成
3	2020 年 4 月 7 号	确定代理服务器你基本框架	已完成
4	2020 年 4 月 10 号	编写 DEMO 程序	已完成
5	2020 年 4 月 12 号	编写程序配置文件部分	已完成
6	2020 年 4 月 15 号	编写系统任务基本框架	已完成
7	2020 年 4 月 18 号	编写代理服务器任务监听部分代码	已完成
8	2020 年 4 月 21 号	编写代理服务器请求解析部分代码	已完成
9	2020 年 4 月 25 号	完成代理服务器剩余模块	已完成
10	2020 年 4 月 27 号	测试代理服务器功能	已完成
11	2020 年 4 月 30 号	完成代理服务器设计报告编写	已完成

### 四、实习内容

本次实习最主要的内容就是设计与实现 HTTP 代理服务器，包含 HTTP 代理服务器拥有的基本功能，下面列出几个设计初期拟定的施行方案：

1、在 Windows 平台使用 C++编程语言进行编程，网络编程库使用 Windows 下的 C 语言的库函数，并设计可以直观表达各项功能的界面，通过 GUI（Graphical User Interface）面板来编辑基本参数，控制的各项功能。

2、在 Linux 平台上进行 C++编程，使用可跨平台运行的 Qt 库来实现代理服务器的基本功能。同时使用 Qt 库制作一个控制界面，通过 GUI 完成代理服务器人机交互。

3、在 Linux 平台上进行 C++编程，使用基于 Linux 的原生 C/C++网络编程库来开发代理服务器的各项功能。使用配置文件和运行命令的方式来完成代理服务器的控制。

4、在 Linux 上进行编程，使用可跨平台的 Asio 网络库来开发服务器的网络连接部分，

程序的其他重要模块也使用基于 Boost 的库进行开发，不制作 GUI 界面，所有服务器控制都基于配置文件和控制台命令来完成，花费更多时间来提升服务器的稳定性和安全性，实现更多有用的功能。

通过查阅相关资料，对现在网络上普遍流行的代理服务器的设计结构进行学习，下面将对上面提出的几个拟定方案进行选择。

首先排除第一个方案，因为在 Windows 上进行 C++ 服务器类编程开发都比较困难，开发周期会大大加长，而且基于 Windows 开发的服务器性能和安全性方面都不是很高。第二个方案使用 Linux 作为编程和运行平台，因为 Linux 对服务器性能和安全性方面的保护是非常到位的，其次 Linux 下的 C++ 编程更加方便可以使用的资源更加多，而且 QT 库也是非常优秀的跨平台 C++ 库，所以这个方案相对与方案一有很大提升。但是往往决定一款服务器程序性能好坏的因素，GUI 界面的优劣并不在其中，而是其长期运行的稳定性，处理客户请求的效率和并发性等因素。所以设计代理服务器程序更应该着重考虑它在后台的运行情况和资源消耗。方案三就是考虑到了这一点，舍弃了复杂的 GUI 界面设计，花更多时间来提升后台性能，开发更多实用的功能。方案四是在方案三的基础上进一步提高，使用先进的 Boost 库来开发程序。由于 C++ 官方标准库中没有对网络部分进行补充，所以在编写 C++ 程序的网络连接部分时总是要使用最基本的 C 语言库中的语义，这是相当复杂且很不安全的，所以 Boost 库中有 Asio 对 C++ 网络编程部分进行了第三方补充。Asio 库的使用不仅简单容易上手，而且稳定性非常高。Boost 库中还存有其他方便好用的库，也可以添加到代理服务器的编程中，以降低开发难度。综上，采用方案四作为程序开发的最终方案。

下面列出最终采用方案的系统构成为：

开发平台：Linux

编程语言：C++

编译器：GCC

编辑器：qtcreator

使用的第三方库：Boost

程序类型：后台服务程序

根据以上预设计方案，想要完整开发一款 HTTP 代理服务器，其中需要涉及的相关技术包括：Linux 系统使用方法、C++ 的新特性、Boost 编程库的使用、HTTP 协议的基本内容、HTTP 代理服务器的运行流程等。

一个高性能且运行稳定的 HTTP 代理服务器必须拥有一个优秀的网络模型作为程序开发

的基础，下面对几种常见的网络模型进行优劣分析：

### 1、迭代模型

优点：设计简单，系统稳定，适合系统模型开发时，对方案可行性验证时使用。

缺点：因为迭代模型每次只能处理一个任务，当任务较多时工作效率将会非常低，所以实际开发时一般不会选择使用。

### 2、并发模型

优点：多线程处理，有效提高了任务处理速度。当并发量比较低时，稳定性高。

缺点：不适合大量连接，当并发量提高时，资源消耗显著提高，系统会因为过度使用线程而崩溃甚至出现死机。

### 3、分发模型

优点：适合大量用户接入，将通信处理与业务逻辑区分开来，使得系统结构逻辑严密，可扩展性强。

缺点：模型复杂度高，设计难度大，并且容易因为线程逻辑而导致的内存调用错误，使得系统崩溃。

### 4、线程池模型

优点：合理设置初始线程数，可以有效利用多个 CPU 资源，极大提高任务处理速度，系统稳定较高。

缺点：与分发模型类似，因为涉及到多线程资源共享问题，如果程序流程设计不当，容易导致程序内存调用错误。为了解决资源同步共享问题，使用互斥锁等线程同步机制也会浪费一些系统资源。

根据对以上 4 种网络模型的优缺点分析，最后得出使用线程池模型可以很好的设计出一个高性能且稳定的 HTTP 代理服务器。并且 Asio 网络库与线程池模型的兼容性比较好，减少了程序开发过程中可能会遇到的复杂问题。

HTTP 代理服务器的信息输入主要包括配置文件、控制台选项、来自客户端的请求报文和来自服务器的响应报文。配置文件和控制台属于程序基本配置类，它们都可以通过 ProgramOptions 库进行处理，从而产生可以被程序读取的键值对变量。来自客户端的请求报文在通过 Asio 网络库读取为字节流后，可以用专门处理 HTTP 请求报文的程序段进行解析，从中可以提取出客户端请求的路径和目标服务器地址等重要信息。来自上游服务器的响应报文也需要通过专门处理 HTTP 响应报文的程序段进行解析，得出本次请求的响应结果，然后再将这个结果返回给客户端的浏览器。

HTTP 代理服务器的信息输出主要包括发向客户端浏览器的响应报文和发向上游服务器的请求报文。其中发给上游服务器的请求报文是通过来自客户端请求报文解析处理后重新组装而成的，发送请求报文是为了向上游服务器请求客户端需要的内容。发送给客户端的响应报文是通过来自上游浏览器响应报文解析处理后重新组装而成的，响应报文是为了响应来自客户端的请求。

图 1 对 HTTP 代理服务器可能存在的输入输出内容进行了展示。

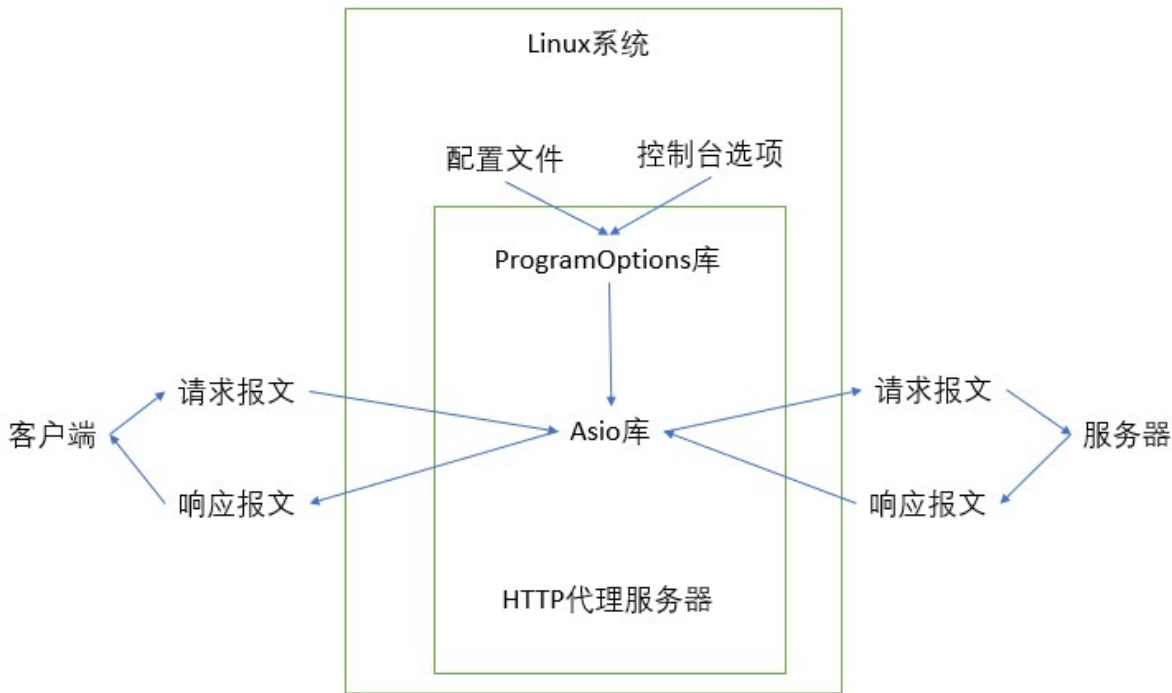


图 1 HTTP 代理服务器输入输出框架

程序主体部分主要包含配置文件、请求监听和请求处理这几部分，下面对程序主体进行介绍。

配置文件包含：基本配置文件、过滤配置表、反向代理配置表三个文件。

基本配置用于保存程序最基本的配置，例如监听端口和监听地址等。

过滤配置表存储的是代理服务器监听请求时，需要过滤的地址和 IP。

反向代理配置表属于代理服务器反向代理部分功能的文件，用于记录反向代理名称与地址的对应关系。

请求监听部分的作用是监听来自客户端的连接请求，HTTP 代理服务器需要先开启 TCP 端口监听服务功能，这样只要有来自 TCP 客户端是连接，经过 TCP 三次握手确认后，请求监听部分代码就会把确认过的连接转交给请求报文接收任务处理线程进行处理，然后开启下一次监听。

这部分代码使用的是 Asio 库中 acceptor 类，其原理与 C 语言 TCP 套接字监听相同。这段代码首先从配置文件中解析出需要监听地址与端口，然后打开 acceptor 实例并绑定地址与端口，最后开始监听来自客户端的请求。当有客户端连接服务器时，系统就会接受连接请求。

请求报文处理部分代码包含两个部分，第一部分为报文接收阶段，第二部分为报文处理阶段。接收部分使用的是 Asio 库函数 `async_read` 进行读取，`async_read` 能异步接收来自客户端的请求报文，当有数据需要接收时，`async_read` 的处理函数就被唤醒，没有数据时就保持休眠状态，这样就可以将系统资源交给急需处理的函数。 `async_read` 每次读取一定数量的报文内容，然后将这部分内容交给请求报文解析函数进行解析，如果解析函数返回 Good，即表示请求报文已经处理完全了，如果返回 Bad，则表示请求报文内容有误，并放弃了本次解析。

请求报文解析部分使用的一个单独的 HTTP 协议报文解析类进行的管理。解析函数会逐个字符的去读取报文内容，然后按照状态机的方式对报文的处理状态进行严格管理，一旦发现一个不符合报文规范的字符，解析函数就会返回 Bad，表示请求报文内容有误，本次请求被拒绝。其中请求报文读取到的内容会被存放到 HTTP 协议内容结构体内，方便后续调用。

当来自客户端的请求报文解析成功后，就可以从请求报文中得到请求方法和目标服务器地址两个关键的数据。请求方法用于选择客户端请求服务器时使用的连接方法，包括 GET、POST、CONNECT 等，其中 GET 和 POST 方法连接处理流程一样，而 CONNECT 方法则需要建立 HTTPS 安全链接。目标服务器地址用于保存客户端想要连接的服务器的实际地址。

连接上位机部分代码，首先使用 Asio 库中的 `query` 解析类对远端服务器的地址和端口进行解析，得到服务器的实际 IP，然后使用 `async_connect` 异步连接方法来连接远端服务器。当连接成功的任务处理函数被执行后，处理函数就会判断连接是否使用了 CONNECT 方法，如果不是就发送组装好的请求报文给远端服务器，如果是则发送安全链接响应报文给客户端。

当远端服务器收到请求报文后就会对报文内容做出响应，返回响应报文，HTTP 代理服务器在收到响应报文后只需要将响应报文继续返回给客户端即可，这样一次完整的 HTTP 请求代理就算结束了。

## 五、实习总结

### 5.1 实习中所涉及的技术问题

#### 1、HTTP 协议



HTTP 协议主要是用于 Web 客户端与 Web 服务器之间进行数据交换而被定义出来的。编写 HTTP 代理服务器首先需要弄明白的就是 HTTP 协议的基本结构，标准的 HTTP 协议包含请求报文与响应报文两部分内容，请求报文首先包含一个请求行，请求行中有请求的类型、请求的内容和协议版本，然后就是多行请求头，最后是请求的消息体。响应报文的结构与请求报文类似，首先是一个状态行包含响应码、响应消息和协议版本，然后是多行响应头，最后是响应的消息内容。

HTTP 协议工作时的基本流程为：首先 Web 浏览器向 Web 服务器发送包含请求内容与目标信息的请求报文，然后 Web 服务器解析请求报文内容，根据请求内容向 Web 浏览器返回响应报文。由于 HTTP 代理服务器的请求格式也符合 HTTP 协议，因此代理服务器在收到客户端请求时的第一步也是按照协议格式解析出请求的内容，以此才能判断客户端想要请求的内容以及目标服务器等重要信息。

## 2、Linux 操作系统

该操作系统的内核是由林纳斯·托瓦兹在 1991 年 10 月 5 日首次发布的，这个系统完全是开源自由的且非常安全可靠，所以目前世界上大部分的服务器程序都是由 Linux 系统在运行。本次设计开发所使用的系统是 Linux 系统发行版中的 ArchLinux 系统，HTTP 代理服务器的代码编写、运行和调试都将在这个系统中完成，代码编写完成后可直接发布到 Linux 云服务器中运行起来，这样系统便能起到代理服务器的作用。刚上手使用 Linux 系统一定要学会使用控制台和几个基本指令。Linux 的控制台是系统的灵魂所在，只要学会使用控制台，基本就可以操作系统的所有功能了。

## 3、C++编程语言

C++是一种通用程序设计语言，其使用非常广泛。目前 C++已经更新了很多版本了，最新版本为 C++20 是第六个 C++标准，而这次的设计只需要学习到 C++的第三个标准即 C++11。下面列出几个编程中常用的新特性：move 移动语义、可变参数模板、初始化列表、auto 关键字、lambda 表达式、类型获取等等。C++11 中向 C++标准库更新了很多非常实用的功能，极大的提高了 C++编程开发的效率。

## 4、Boost 库

Boost 库是 C++的第三方代码库，提供免费同行评议。Boost 库与 C++标准库的兼容性非常强，所以其中有很多优秀的设计都被选入使用到了 C++标准库之中。由于 C++标准库中不提供关于网络编程方面的接口，要使用网络编程就必须使用原始的 C 语言提供的库函数，这是非常不安全而且繁琐的。所以本次设计重点使用的是 Boost 库中的 Asio 库，Asio 提供了

一个便携式的网络接口和底层 I/O 操作，包括套接字的封装、计时器、主机名解析、套接字流、串行端口、文件描述符等操作。这些操作在 HTTP 代理服务器的编写中都是非常有用的，不仅提高了代码的安全性，而且减少了代码的编写难度。同时程序还会用到另一个库 Program Options，这个库以键值对的方式提供了更加方便的控制台选项设计功能和配置文件读取功能。对于无界面服务器程序的开发具有很大的帮助。

## 5.2 实习中所涉及的环境与法律问题

因为代理服务器可以作为网络之间的连通工具，让客户可以突破网络限制，访问到原本不能访问的内容，所以也能通过代理服务器访问到非法内容。如果使用代理服务器的方式不正确，让其他人知道了公司代理服务器的访问入口，从而盗取公司重要资料，这将是非常危险的。所以使用代理服务器时必须注意法律及注意事项等问题，避免风险的发生。

## 5.3 社会责任与团队合作问题

团队合作一定要讲求效率。做好工作日程安排，并严格地按照日程安排进行工作。这一方面是因为团队目标的完成需要效率的支撑，另一方面还因为“一鼓作气，再而衰，三而竭”。作为队员，有时候无法改变团队的做事风格，自己应该合理安排时间，做好自己的事情，不至于因为团队的事情而耽误了自己的事情。本次实习任务让我明白要想做好一个任务，只有在团队成员的共同努力下才能完成。团队成员之间合理分工合作不仅可以提高做事效率，而且还能提高产品质量。

## 5.4 技术学习中的收获

通过本次实习设计的锻炼，其中收获最大的就是 Linux 系统使用及 C++编程能力的进一步提升。

首先是 Linux 系统使用上面的收获。因为现在主流的电脑系统是 Windows，占据了 90% 以上的电脑市场，所以很多人对于 Linux 系统的电脑都很陌生。虽然 Linux 系统用户基数和系统使用体验上相较于 Windows 确实差了很多，尤其是在软件界面方面，但这也正是 Linux 系统优势所在。Linux 系统中的大部分软件都没有可视化界面，仅仅依靠控制台程序来操作和执行，抛弃掉界面化的使用方式，系统减少渲染图形的消耗，这使得运行 Linux 系统上的软件在速度方面有着天生的优势。而且，不管是微软的 Windows 还是苹果的 macOS，都是需要付费的，而且是比较昂贵的。而 Linux 是免费的，是开源的，我们可以随时取得其源代码，根据我们不同的需求进行定制，这对于用户特别是程序开发人员是非常重要的。由于 Linux 的内核大部分是用 C 语言编写的，并采用了可移植的 Unix 标准应用程序接口，所以它支持如 i386、Alpha、AMD 和 Sparc 等系统平台，以及从个人电脑到大型主机，甚至

包括嵌入式系统在内的各种硬件设备。因此很多优秀的高性能的服务器软件都是运行在 Linux 系统之上。现在很多公司都是使用 Linux 系统来管理系统程序和重要文件，因此学会 Linux 系统的使用无疑是以后参与工作后非常重要的一步。

然后就是关于 C++编程语言方面的收获。C 语言现在是绝大部分大学的学生，入学后掌握的第一门编程语言。而大学教育中的 C 语言知识讲解很多都只是皮毛，并没有深入学习其更高级的使用。C++语言作为C语言面向对象的升级与提升，不仅很好的继承了C语言精髓，而且在面向对象方面也是具有独到的特色，所以想要真正学习好 C++编程语言的使用并非是一件容易的事。学习这门语言只有在掌握足够理论知识的前提上，进行更多的实践，才能将学习到的知识融会贯通。

## 六、收获体会及建议意见

回顾实习生活，感触很深，收获颇丰。经过实习我更加认识到实践是检验真理的唯一标准，只学不实践，那么所学的就等于零，理论应当与实践相结合。另一方面，实践可为以后找工作打基础。我感觉实践是大学生活的第二课堂，是知识常新和发展的源泉，是检验真理的试金石，也是大学生锻炼成长的有效途径。一个人的知识和本事仅有在实践中才能发挥作用，才能得到丰富、完善和发展。大学生成长，就要勤于实践，将所学的理论知识与实践相结合一齐，在实践中继续学习，不断总结，逐步完善，有所创新，并在实践中提高自我的各方面知识、本事、技术等因素融合成的综合素质和本事，为自我事业的成功打下良好的基础。

在短短的一个月实习时间里受益匪浅。不仅开拓了我认识知识的眼界，而且让我懂得了很多为人处世的道理。以后真正进入工作岗位也应该时刻保持这样的学习态度，把所学的知识充分运用到实践中，并在实践中继续学习掌握新的知识不断提高自我。

虽然本次设计的 HTTP 代理服务器在功能使用上并不是很完善，对于内容过滤和反向代理方面都只是一个测试阶段功能，在实际使用中的表现也不尽人意，同时还有一些原本计划设计的功能，却因为种种原因而没有实现的，但是我一定会在将来继续学习新的知识，使用新知识来进一步完善和提高本次设计中很多不足的地方。