
华中科技大学计算机学院

《计算机通信与网络》实验报告

班级_____ 姓名_____ 学号_____

项目	Socket 编程 (40%)	数据可靠传输协议设计 (20%)	CPT 组网 (20%)	平时成绩 (20%)	总分
得分					

教师评语：

教师签名：

给分日期：

目 录

实验一 SOCKET 编程实验	1
1.1 环境	1
1.2 系统功能需求	1
1.3 系统设计	2
1.4 系统实现	4
1.5 系统测试及结果说明	6
1.6 其它需要说明的问题	9
1.7 参考文献	9
心得体会与建议	10
2.1 心得体会	10
2.2 建议	11

实验一 Socket 编程实验

1.1 环境

1.1.1 开发平台

处理器：Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz 1.80 GHz

操作系统：Windows 10 20H2

机带主存：8.00 GB

开发平台：Visual Studio 2019

开发语言：C/C++

1.1.2 运行平台

处理器：Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz 1.80 GHz

操作系统：Windows 10 20H2

运行软件：Visual Studio 2019

1.2 系统功能需求

基本需求：

1.可配置 Web 服务器的监听地址、监听端口和主目录（不得写在代码里面，不能每配置一次都要重编译代码）；

2.能够单线程处理一个请求。当一个客户（浏览器,输入 URL: http://202.103.2.3/index.html）连接时创建一个连接套接字；

3.从连接套接字接收 http 请求报文，并根据请求报文的确定用户请求的网页文件；

4.从服务器的文件系统获得请求的文件。 创建一个由请求的文件组成的 http 响应报文；

5.经 TCP 连接向请求的浏览器发送响应，浏览器可以正确显示网页的内容。

高级要求：

1.能够传输包含多媒体（如图片）的网页给客户端，并能在客户端正确显示；

2.在服务器端的屏幕上输出请求的来源（IP 地址、端口号和 HTTP 请求命令行）；

3.在服务器端的屏幕上能够输出对每一个请求处理的结果；

4.对于无法成功定位文件的请求，根据错误原因，作相应错误提示，并具备一定的异常情况处理能力。

1.3 系统设计

系统架构设计：

根据系统需求，要求基于 Socket 搭建客户端与服务器间的数据交互信道。考虑用户与 Web 服务端通过 HTTP 协议进行数据通信，选择采用可靠、面向连接的流类型 Socket 编程框架搭建 Web 服务器，基于流类型 Socket 的系统架构如图 1.1 所示。采用阻塞式 I/O 作为传输方式，方便搭建过程查错与传输过程分析。

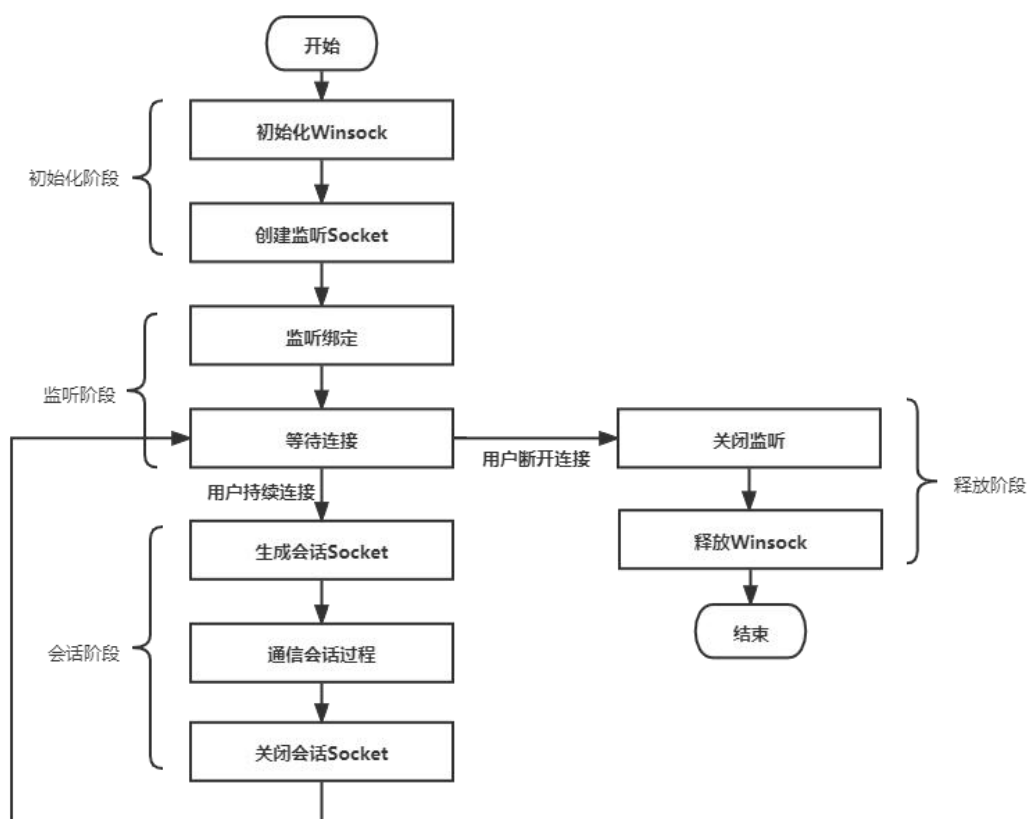


图 1.1 基于流类型 Socket 的系统架构

模块功能设计：

该系统框架主要分为 4 个阶段，即初始化阶段、监听阶段、会话阶段、释放阶段，如图 1.1 所示。每个阶段分属若干过程，用于完成各阶段任务。

初始化阶段模块：

下分 Winsock 初始化，监听 Socket 创建两个过程。使用 WSStartup 初始化 Winsock，检查 Winsock 版本号，该过程若出现错误，使用 WSAGetLastError 显示错误，退出系统。使用 socket 初始化一个监听 Socket，若初始化失败则使用 WSAGetLastError 显示错误，退出系统。

监听阶段模块：

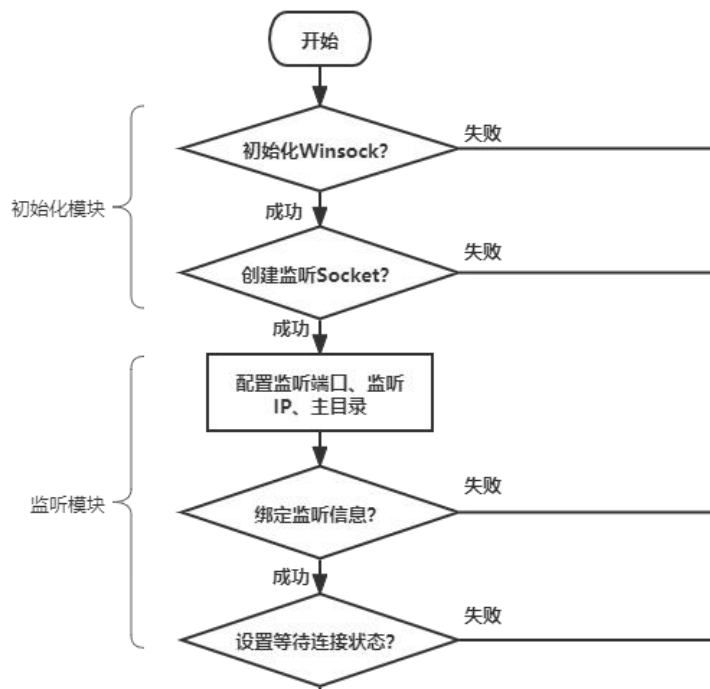
下分配置监听设置，监听绑定，等待连接状态设置三个过程，用于配置监听端口，便于后续的请求接收。配置监听设置过程包括输入监听 IP、监听端口以及提供用户访问的主目录，写入 `sockaddr_in` 结构体中。监听绑定过程，使用 `bind` 函数，将监听信息结构体与监听 `Socket` 绑定。设置等待连接状态则采用 `listen` 函数，使监听 `Socket` 处于等待连接状态，指示当接收到连接请求时，可以建立连接。同样，若过程中出现错误，使用 `WSAGetLastError` 显示错误，退出系统。

会话阶段模块：

下分建立连接，请求数据处理，响应数据处理，关闭会话 `Socket` 四个过程，对收到的请求做出相应的回应，若客户端持续连接，则继续等待下一请求。使用 `accept` 函数与客户端建立连接，生成会话 `Socket`。使用 `recv` 函数接收客户端的请求报文，并处理请求报文，生成请求文件路径。根据生成的文件路径，做出合适的回应，如查找失败，禁止访问，查找成功等。若查找成功，则将相应的文件以二进制形式打开，存放到缓冲区中，作为主体体，使用 `send` 发送响应报文给客户端。若为其他情况，则产生相应的状态码，如 404，403，并发送响应报文。发送成功后，关闭会话 `Socket`。

释放阶段模块：

下分关闭监听，释放 `Winsock` 两个过程。当客户端断开连接后，使用 `closesocket` 关闭监听 `Socket`。使用 `WSACleanup` 释放 `Winsock`，释放分配给该系统的资源。



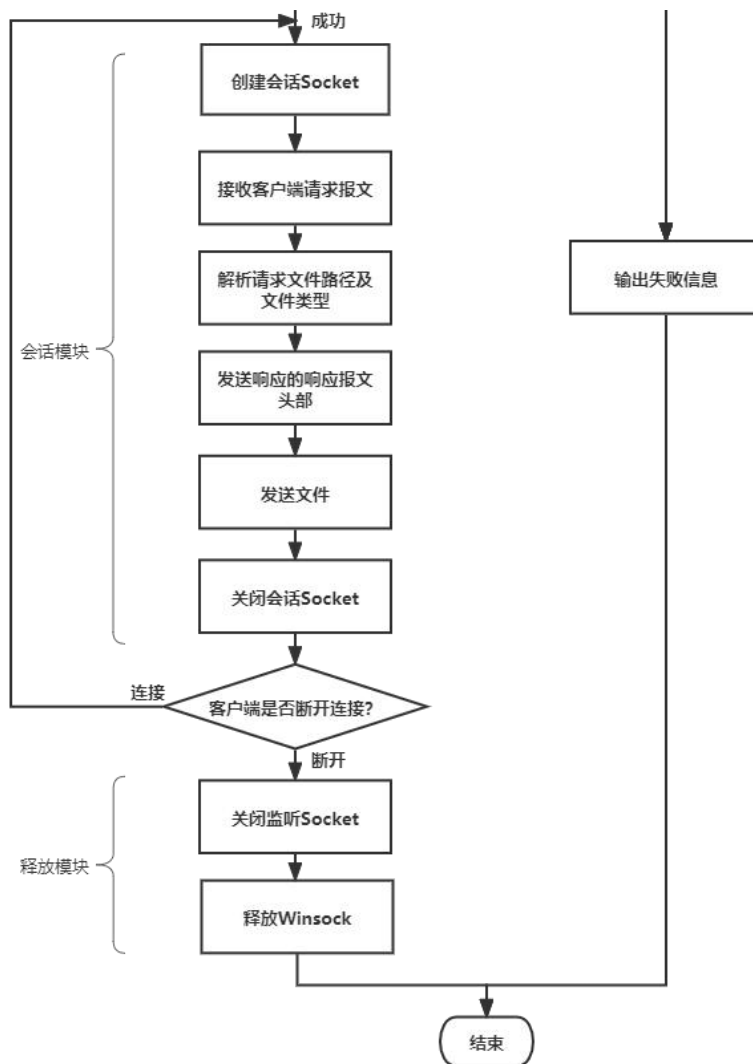


图 1.2 服务器端具体设计流程图

具体流程设计：

根据上述 4 个模块设计，对其进行框架组合，具体流程如图 1.2 所示。具体流程中包含各模块的过程细节，以及对错误的处理流程。

1.4 系统实现

在系统实现中，特别以下功能做具体的介绍。

1. 配置监听端口、监听 IP 及主目录

监听信息由一个结构体 `sockaddr_in` 指示，结构体中包含 3 个关键元素。`short` 类型的 `sin_family`，指示地址族，设为 `AF_INET`；`u_short` 类型的 `sin_port`，指示监听端口，使用 `htons` 函数将主机字节顺序转换为网络字节顺序，Windows 主机字节按小端存储，而网络字节是大端存储，可设为 HTTP 进程通用端口号 80；`in_addr` 结构体，指示监听 IP，使用 `inetaddr` 将 IP 类型的字符串转化为 `unsigned long` 形式的值，进行结构体赋值，可设为 “127.0.0.1”。

2. 解析请求文件路径，即获取 URL

根据 HTTP 的请求报文格式例子，如图 1.3 所示，请求的文件以第一行中的第一个‘/’为起点，至遇见第一个空格为止，为以主目录为根节点的相对地址，与我们输入的主目录结合，构成请求报文的请求文件路径。

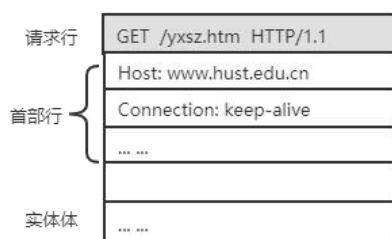


图 1.3 HTTP 请求报文格式样例

3. 构造响应报文

响应报文分为状态行、首部行、空行与实体体四个部分，构造状态行与首部行的流程如图 1.4 所示。

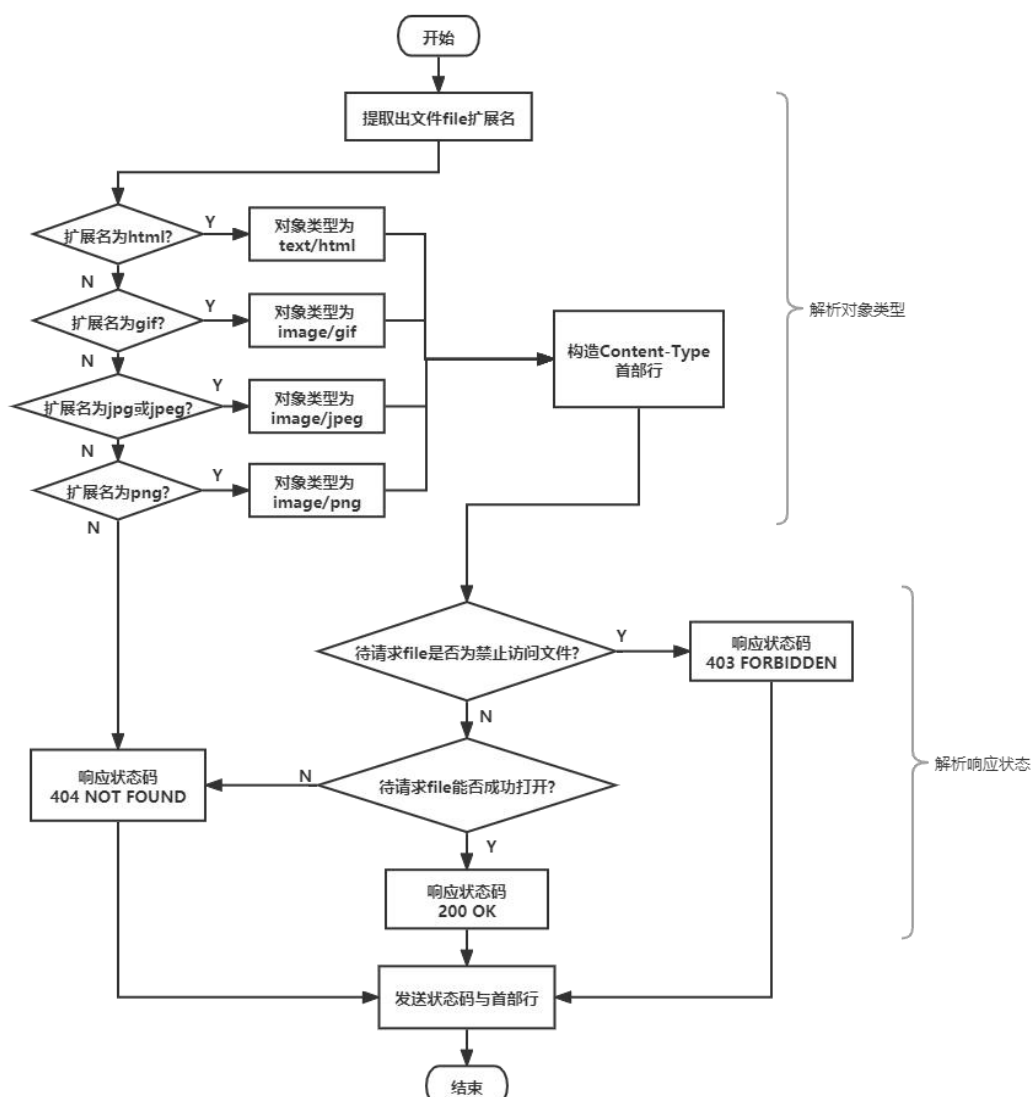


图 1.4 状态行与首部行的构造流程

对状态行与首部行进行构造。首先解析请求文件类型，支持 html、gif、jpg、jpeg、png5

种文件类型。提取出文件结尾的扩展名，将其转化为正式对象类型，如 `text/html` 等。将其与字符串 “Content-Type: ” 组合成对象类型首部行。

分析状态，设置一个禁止访问的文件，若客户端请求该文件，则禁止其访问，状态码设置为 403，则对应的状态行设置为 “HTTP/1.1 403 FORBIDDEN\r\n”。注意，不能忘记末尾的 ‘\r\n’ 回车与换行符。若打开文件失败，则表明无法在主目录下寻找到请求文件，状态行设置为 “HTTP/1.1 404 NOT FOUND\r\n”。打开成功，则表明查找成功，允许访问，状态行设置为 “HTTP/1.1 200 OK\r\n”。

而实体体的发送，由于传输内容包含图片等多媒体信息，需要以二进制形式 “rb” 打开文件，进行实体体的构造。发送时，通过 `fseek` 和 `ftell` 使用，获取二进制文件长度。通过 `fseek` 将文件指针定位到文件尾，再通过 `ftell` 函数计算指针偏移的字节数，从而得到文件长度，再将文件指针移到文件头，读取数据，构造实体体。

注意，在首部行与主体体间，还需发送空行，即 “\r\n”，以满足 HTTP 响应报文格式。

1.5 系统测试及结果说明

硬件测试环境：

处理器：Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz 1.80 GHz

机带主存：8.00 GB

测试结果与分析：

基础要求：

1. 可配置监听地址、监听端口和主目录，将监听信息与监听 Socket 绑定，且能够在监听端口上进行监听。测试输入与结果如图 1.5 所示。

```
Winsock初始化成功!
Socket创建成功!
-----
请输入监听端口号: 5050
请输入监听IP地址: 127.0.0.1
请输入主目录: C:/index
-----
Socket绑定成功!
设置等待连接状态成功!
```

图 1.5 配置监听信息及端口监听设置测试

2. 收到客户端请求时能创建连接套接字，能够响应客户端的请求，并定位相应的 html 文件，能够构造并发送可被客户端解析的响应报文。

通过 360 极速浏览器输入 URL: 127.0.0.1:5050/hello.html, 访问本地 hello.html 文件，

访问页面如图 1.6 所示。



图 1.6 响应客户请求测试

通过浏览器的网络工具，查看接收到的响应报文格式，如图 1.7 所示。状态为 200，而对象类型为 text/html，为发送的响应头部。

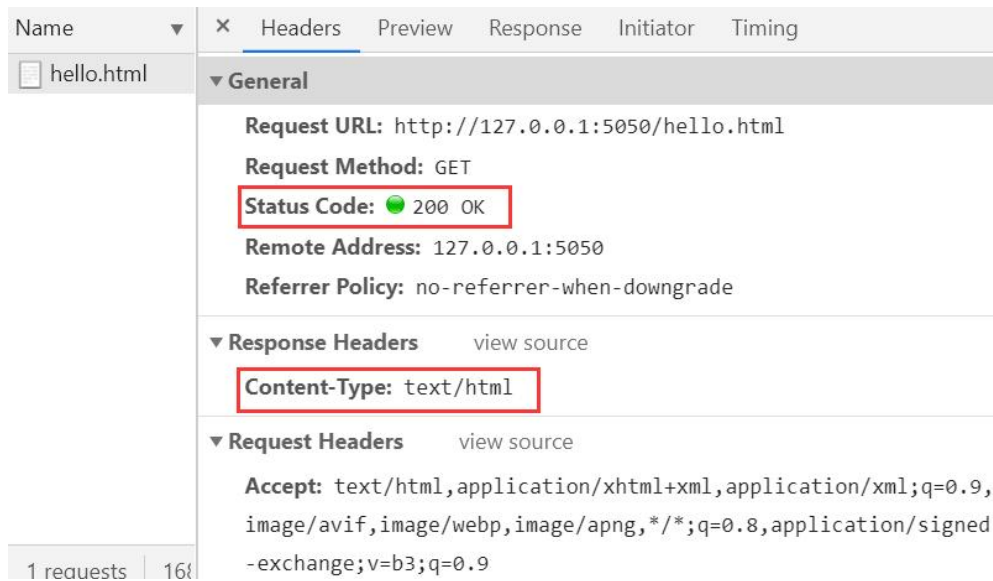


图 1.7 客户端网络工具查看响应报文

高级要求：

1. 在服务器端的屏幕上输出每个请求的来源（IP 地址、端口号和 HTTP 请求命令行）。

对于上述的测试过程，在服务器端窗口，输出请求客户端的 IP 地址与端口号，并且产生请求报文的长度，请求行与首部行，如图 1.8 所示。

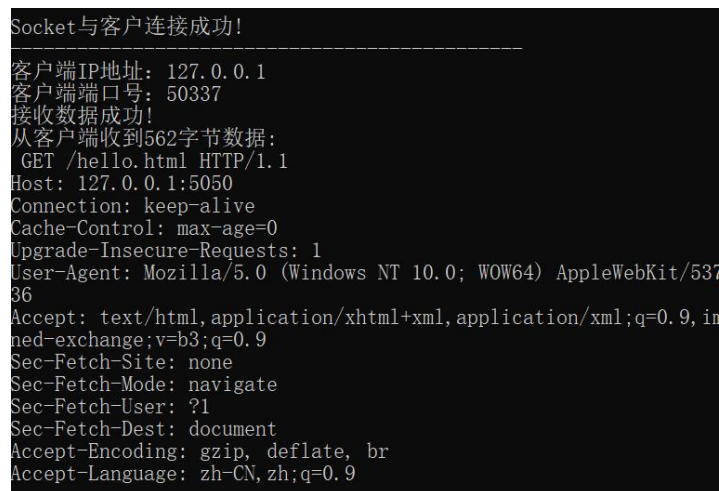


图 1.8 输出客户端信息及请求报文测试

其中，Connection 方式为 keep-alive，表示持续连接模式，客户端还未断开连接。

2. 在服务器端的屏幕上能够输出对每一个请求处理的结果，支持多种类型文件的输出。在上述测试中，响应状态为 200，请求成功，相应的结果输出如图 1.9 所示。

```
-----
请求文件名: /hello.html
完整路径: C:/index/hello.html
200 客户端请求文件查找成功!
-----
```

图 1.9 请求成功的结果输出

除了支持 html 文件请求外，还支持 gif、jpg 等类型的文件请求。在客户端输入 URL：172.0.0.1:5050/class.jpg。成功输出本地文件 class.jpg，如图 1.10 所示。

请求行

方法	sp	URL	sp	版本	cr	lf
----	----	-----	----	----	----	----

首部行

首部字段名:	sp	值	cr	lf
:				
首部字段名:	sp	值	cr	lf

空行

cr	lf
----	----

实体主体

--

请求行
(GET, POST, HEAD 命令)

回车符
换行符

GET /index.html HTTP/1.1\r\n

Host: www-net.cs.umass.edu\r\n

User-Agent: Firefox/3.6.10\r\n

Accept: text/html,application/xhtml+xml\r\n

Accept-Language: en-us,en;q=0.5\r\n

Accept-Encoding: gzip,deflate\r\n

Accept-Charset: ISO-8859-1,utf-8;q=0.7\r\n

Keep-Alive: 115\r\n

Connection: keep-alive\r\n

\r\n

单独一行回车、换行表示报文首部结束

图 1.10 多文件请求的结果输出

3. 对于无法成功定位文件的请求，能够根据错误原因，作相应错误提示。

输入一个主目录下无法查询到的文件对应的 URL：127.0.0.1:5050/myweb.html。服务器端输出错误信息，404 NOT FOUND，如图 1.11 所示。

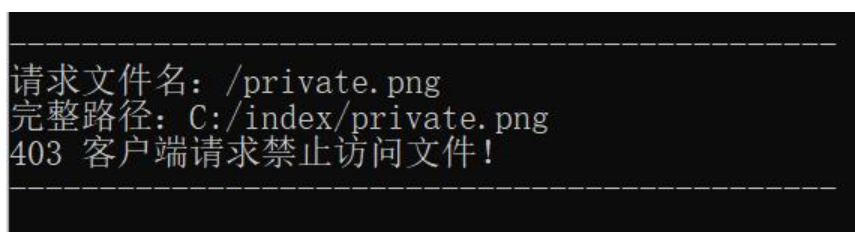
```
-----
请求文件名: /myweb.html
完整路径: C:/index/myweb.html
404 客户端请求文件查找失败!
-----
```

图 1.11 定位失败的结果输出

8

4. 支持一定的异常处理能力。

设置 private.png 为不可访问文件,当客户端输入相应 URL:172.0.0.1:5050/private.png 来访问该文件时,响应 403 FORBIDDEN,在客户端界面输出处理结果,如图 1.12 所示。



```
-----  
请求文件名: /private.png  
完整路径: C:/index/private.png  
403 客户端请求禁止访问文件!  
-----
```

图 1.12 异常处理的结果输出

1.6 其它需要说明的问题

在刚接触套接字概念时,要弄清监听 Socket 与会话 Socket 的区别,并且要养成良好的命名习惯,对两种套接字进行区分。监听 Socket 用于等待连接请求,而会话 Socket 用于与客户端通话。

在进行主目录的配置时,需以'/'作为分割线。原因在于客户端输入的 URL 一般以'/'为界线,为了方便进行路径的合并,在输入主目录时,同样也以'/'进行路径填写。

1.7 参考文献

- [1] (美)詹姆斯·F.库罗斯,(美)基思·W.罗斯著;陈鸣译. 计算机网络:自顶向下方法第7版. 北京:机械出版社,2018.5
- [2] 模块一 Socket 编程实验指导手册.pdf

心得体会与建议

2.1 心得体会

实验一中，对 Web 服务器端的架构让我了解了 Socket 的流程框架，并逐渐掌握了数据在应用层面上的传输，Socket 相当于下层对应用层开放的一个接口，运输层、网络层等下级层面对编程者来说是透明的，十分方便。且在最近一段时间里，也了解了 CoAP 这一应用层协议，使用的是面向无连接的数据报 Socket，过程就更加简洁了。Socket 在应用层面上起到了桥梁的作用。

除此之外，也让我了解了理论运用实践的乐趣。在课堂上学习的理论知识，如果没有实践依托，难免显得晦涩难懂，且映像不深刻。通过实验实践，通过搭建一个 Web 服务器这样一个有趣的过程，可以了解理论在实践中发挥的作用，例如 Socket 套接字，若仅仅只停留在理论上，只是了解套接字的结构，不加以运用的话，很难体会到 Socket 对应用开发的便捷，很难理解各大应用层数据传输的原理细节。

在实验二里，则掌握了运输层的数据可靠传输原理。从停等传输的示例开始，设计流水线传输，循序渐进，流水线传输中两个基础传输方式 GBN 和 SR，最终实现 TCP 的数据传输原理。在虚拟网络环境中，通过查看接收方和发送方的数据传输信息，以及实现输出的两端窗口信息，可以十分详细的查看数据在不可靠网络信道中对数据丢失、数据出错的处理情况。很详细的体会 3 种流水线传输方式的区别。这是单纯的理论学习所无法达到的效果。例如，对计时器的设置和重传机制来说，GBN 和 TCP 只设置一个计时器，而 SR 为每个发送但未接受的数据设置计时器。GBN 当出现计时器超时，需要重传窗口的所有数据，而 SR 仅重传超时的那个数据，TCP 则重传窗口中的首个未确认数据，且有重传机制。这些机制在理论中较为难理解，通过实验中的虚拟网络传输信息的显示，可以很容易发现这些机制的作用，让人映像深刻。

最后则是实验三的组网实验，这个实验我认为是最有趣的实验。我们作为组网者，为下属区域分配 IP，深刻理解网络层的 IP 协议的运作方式，还涉及到虚拟局域网的设置，同样也加深到链路层面的知识。在实验过程中，深切的体会到网关，虚拟网技术以及路由器的具体作用，与生活联系紧密。也了解了路由器配置中的 192.168.0.0 网络号的作用，是局域网的通用网络号，这与显示中的路由器配置十分的相似。通过对虚拟局域网的配置，体会交换机在网络核心中的作用。在现实中，交换机对用户是透明的，只有在实验中才真正明白其作用之大。

通过这三组实验，从应用层到运输层，再到网络层、链路层，自顶向下的了解计算机网络的运作方式，全面的理解了计算机网络在现实中发挥的巨大作用。

2.2 建议

实验中，给出了基础的示例，方便同学理解任务要求，十分有效。且实验任务书介绍的十分详细。不过实验中的输出结果可以适当规范以下，比如实验二中的窗口，可以对窗口的格式做出一个具体的要求，例如 SR 中要对已确认的窗口进行标记并且在窗口中体现。在检查时也可以少花些时间来一步步查找过程，甚至还要对原窗口进行修改，拖延了检查进度。

另外，希望类似实验三这样有趣的实验能多一些，期待课程组的后续安排~