# 简易HTTP服务器实现

## 设计背景

随着网络技术的不断发展，http（web）服务器已经成为了人们生活中不可或缺的一部分。我们无时不刻的访问者http服务器，如百度和谷歌等，每天都享受着http服务器给我们带来的便利。如此重要的程序却很少有人知道他背后的工作原理，尽管现在开源的服务器已经十分成熟如，apache，tomcat，nginx。所以本次实验通过设计一个简易的http服务器来学习其原理。

## 开发环境

操作系统：ubuntu 18.04 x86\_64

编译器环境：gcc 10

IDE：vscode

## 设计思路

服务器作为运行在后台的程序，被动地等待客户端发送过来的http请求，然后进行相应的处理。由于http服务器需要同时处理多个客户端的请求，所以本次http服务器采用POSIX多线程设计，使用accept等待客户端的连接，使用pthread\_create对每一个客户端的请求分别创建一个线程来进行处理，当然也可以使用fork创建进程。设计http服务器本质上是对网络编程的考验，http服务器和传统的tcp网络程序类似，只是对客户端发过来的数据包进行了一系列的分析和处理，然后做出相应的操作。首先服务器需要创建套接字，然后设置sockaddr的数据结构（端口和IP），然后使用bind函数进行套接字和sockaddr的绑定，随后使用listen初始化tcp连接队列和backend（最大连接数），最后使用无限循环加上accept来处理客户端的连接。对于每次accept接收到的连接，分别使用pthread\_create创建线程进行处理，随后把结果返回给对应的客户端。

## 程序代码

### 代码结构如下

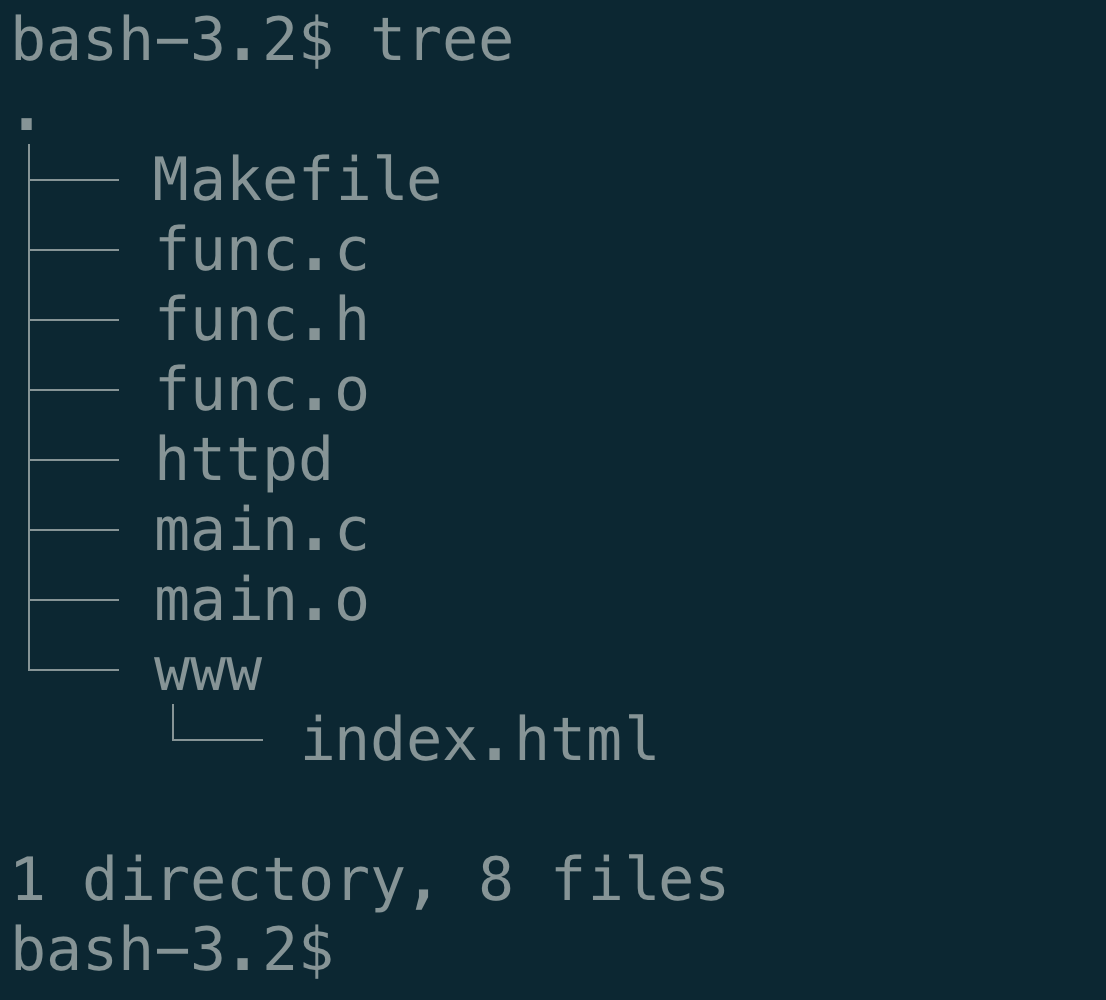


表 1项目结构

### 服务端关键代码

使用while（1）无限循环等待客户端连接，使用pthread\_create创建线程，每个线程使用HandleAcceptRequest函数进行处理。



表 2等待客户端连接

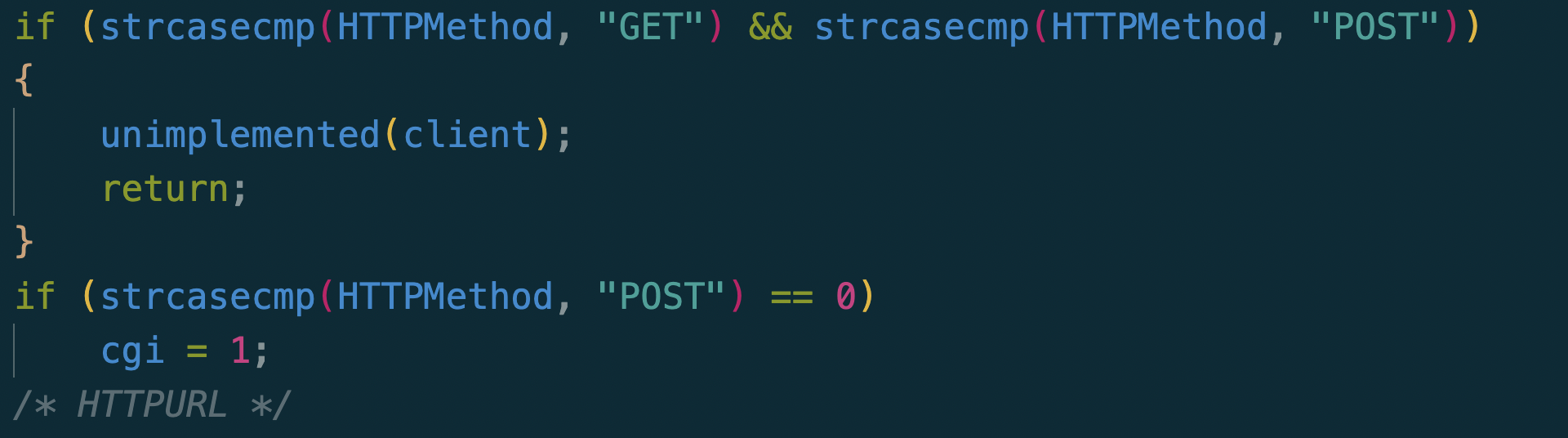


表 3处理POST/GET请求

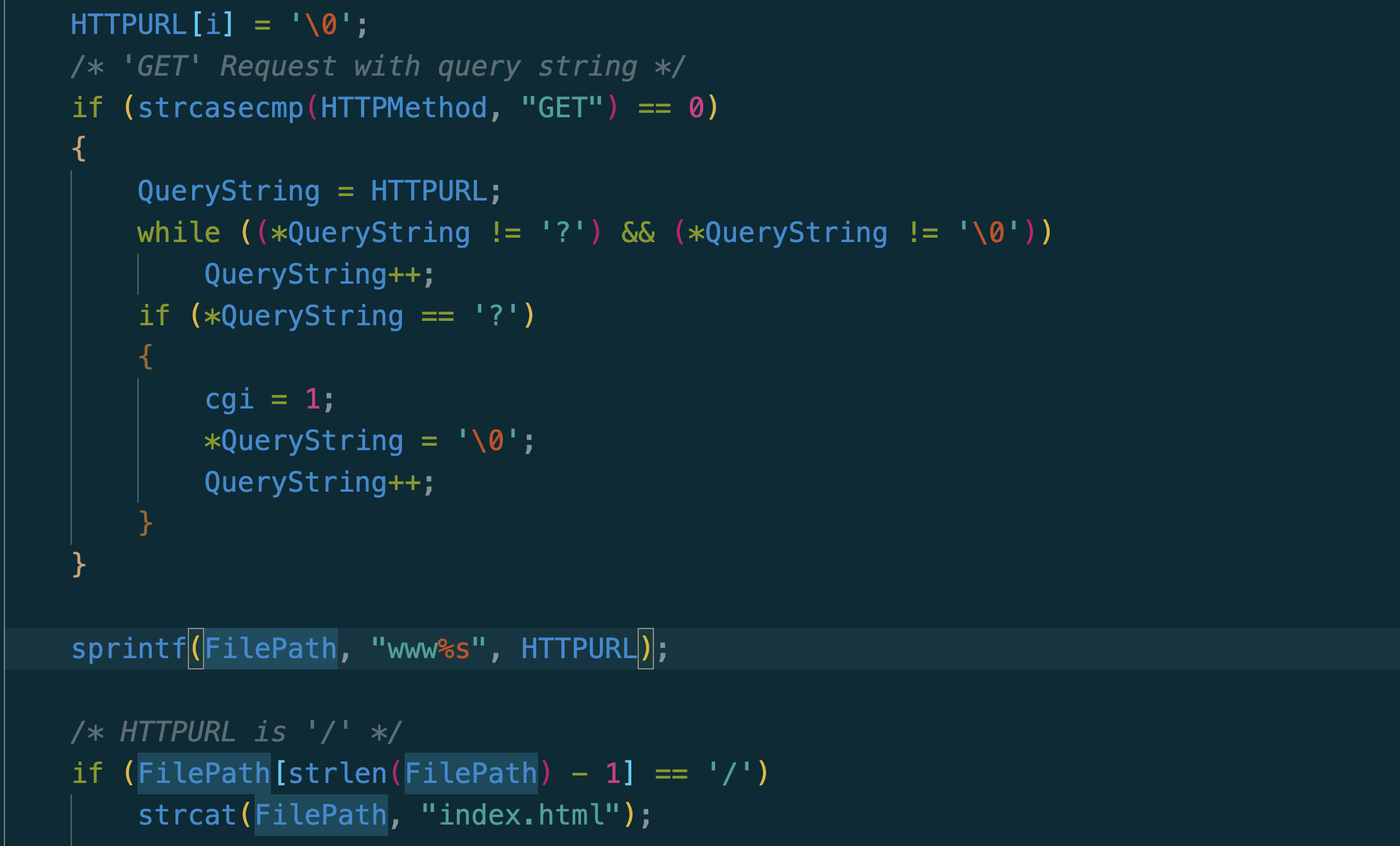


表 4加载index.html

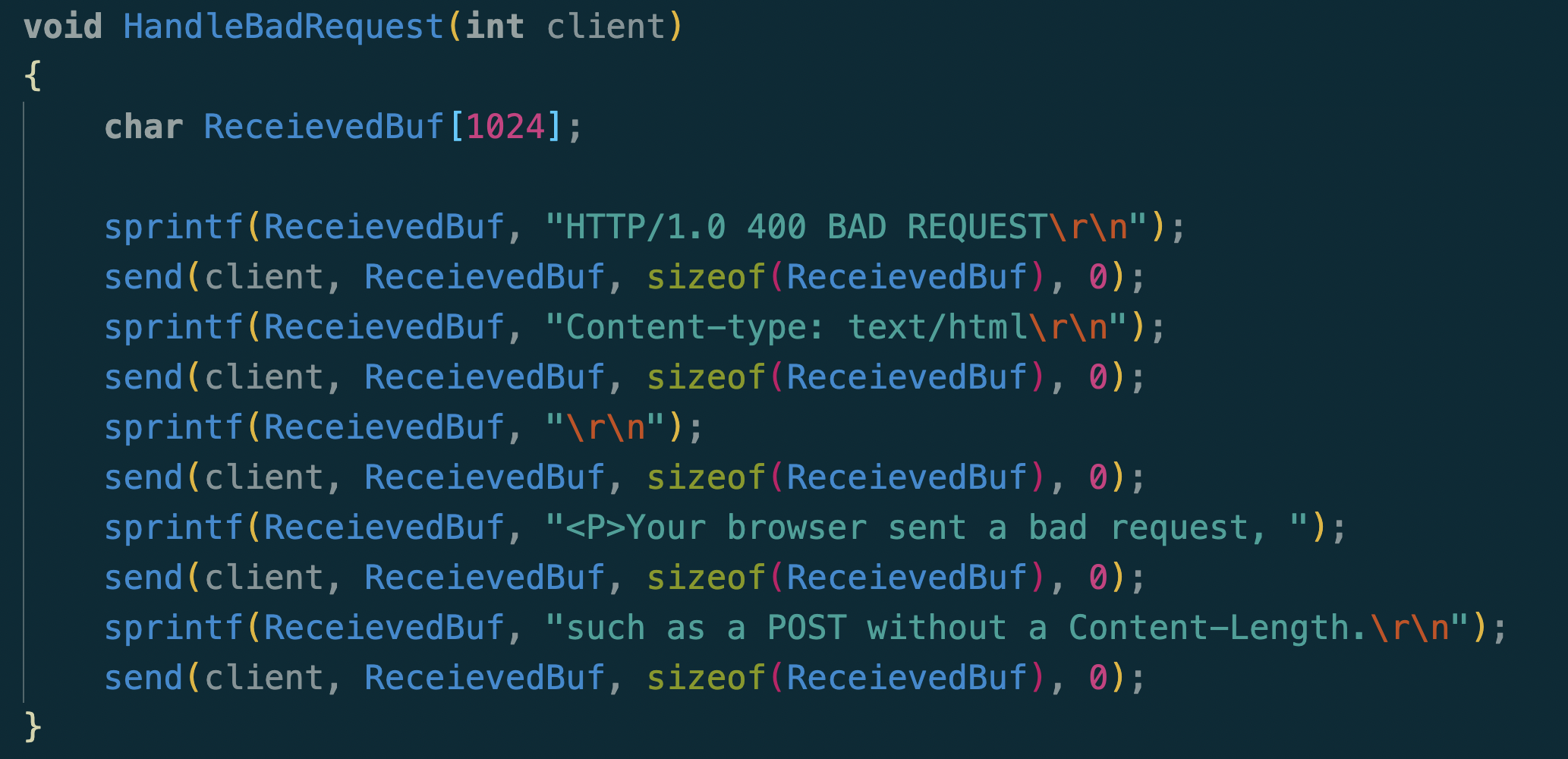


表 5处理错误请求

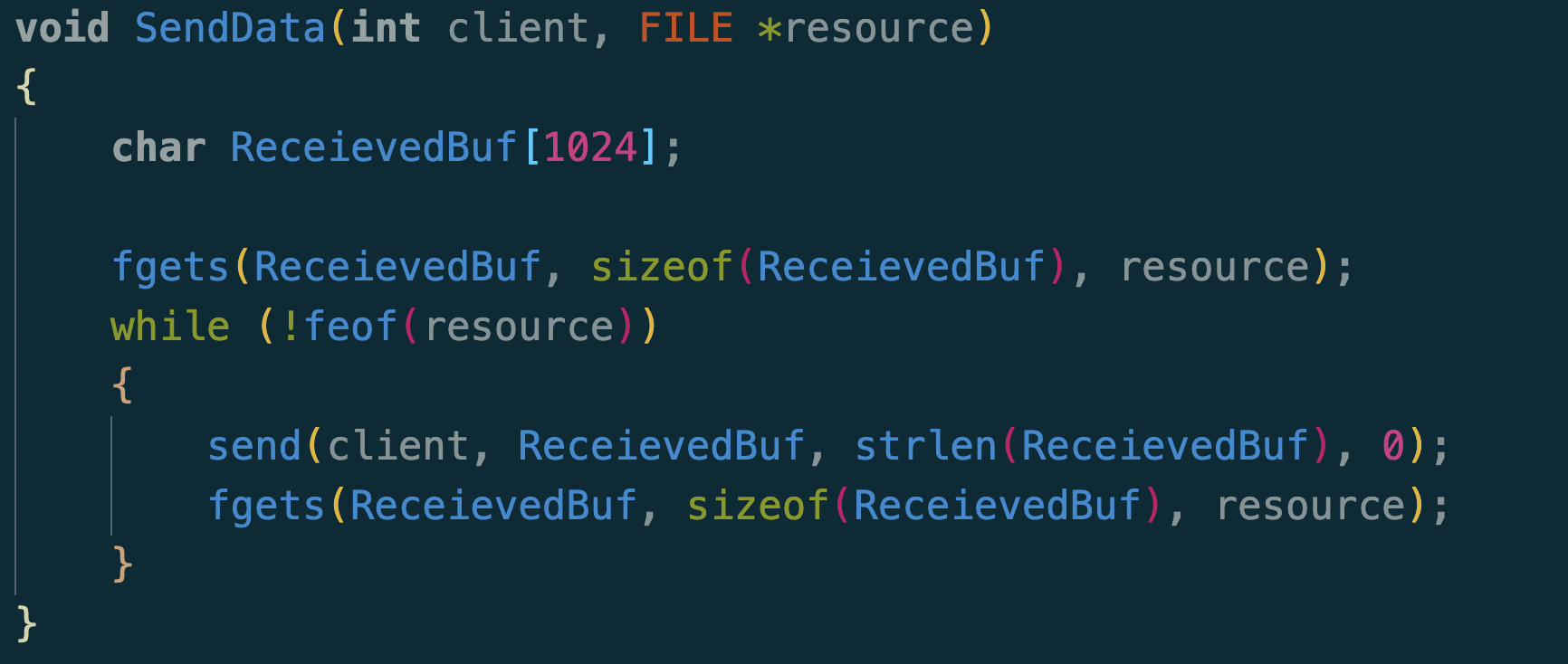


表 6发送数据

### 程序依赖图

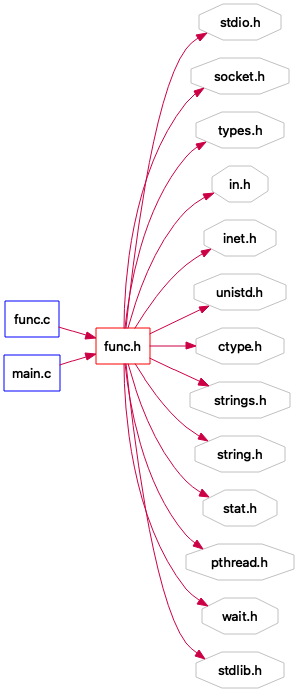


表 7程序依赖图

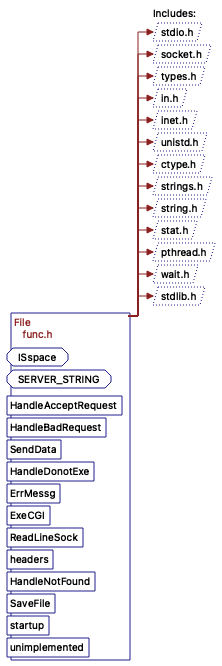


表 8函数声明

### 程序流程图

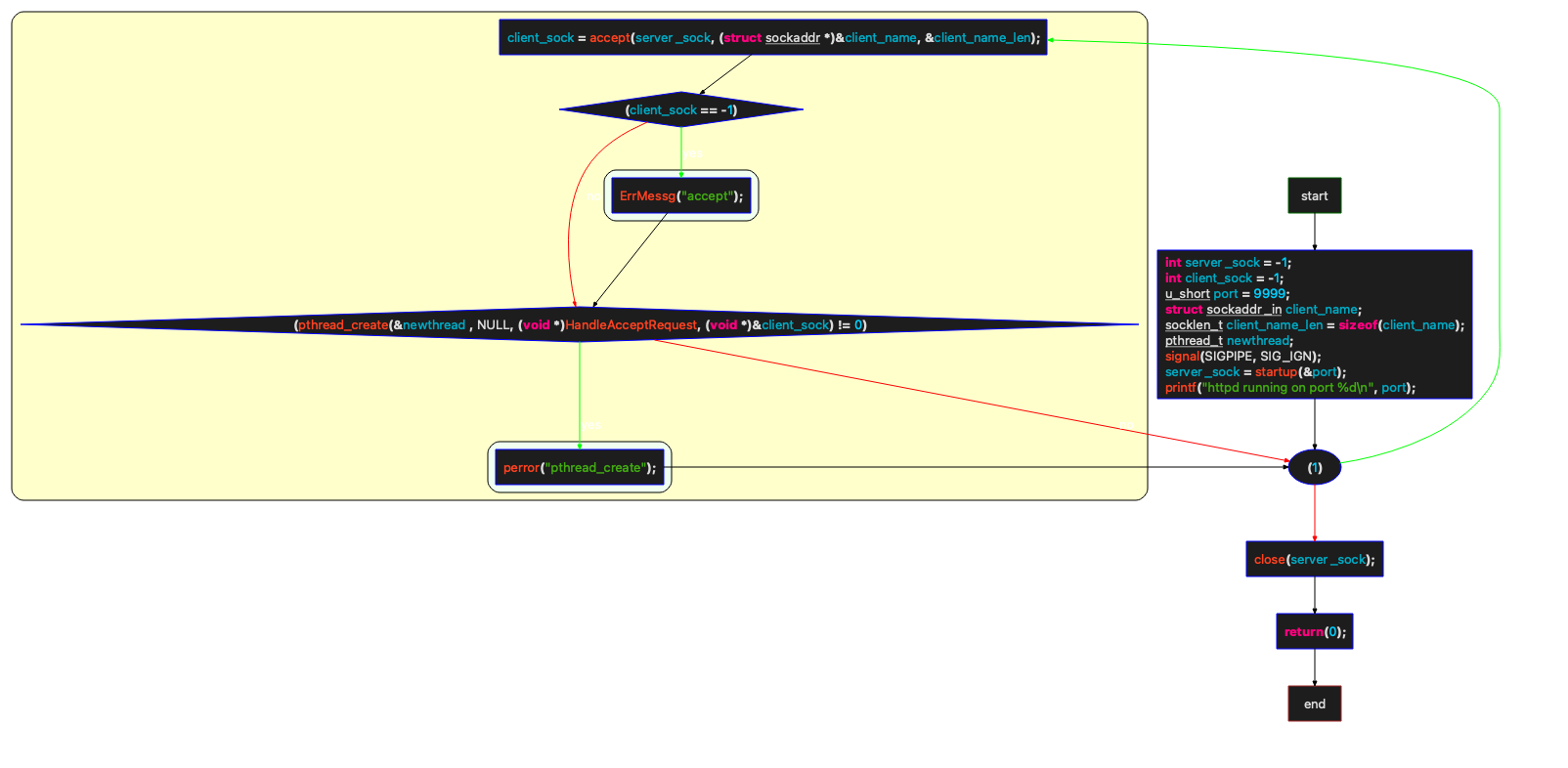


表 9main函数流程图



表 10HandleAccept流程图

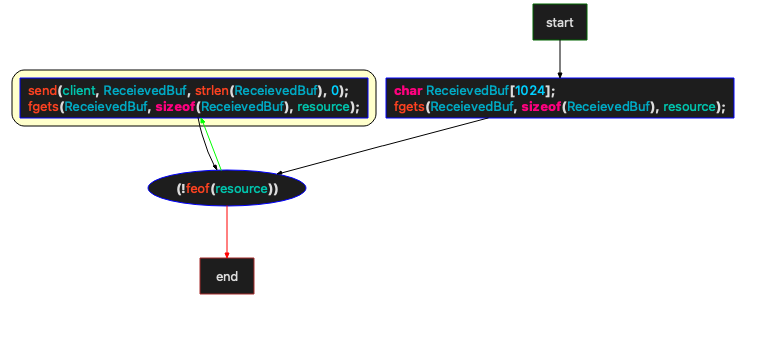


表 11SendData函数

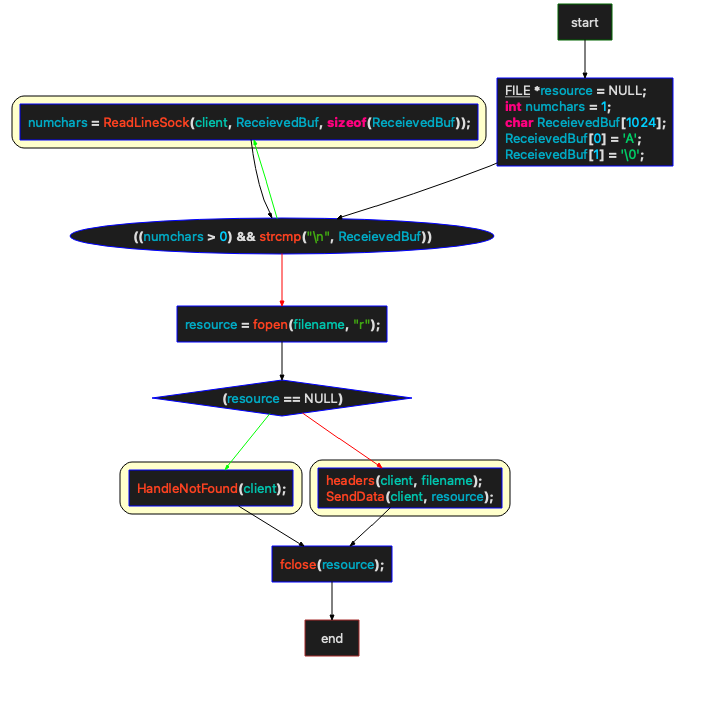


表 12RecvFile函数

## 运行结果

### 使用Make编译程序

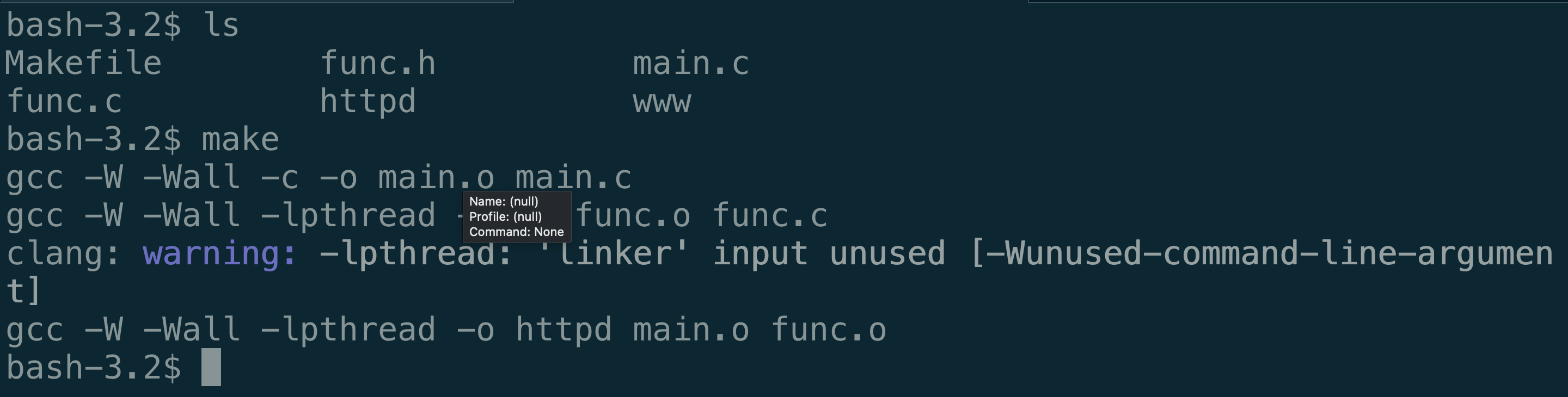


表 13编译程序

### 运行程序

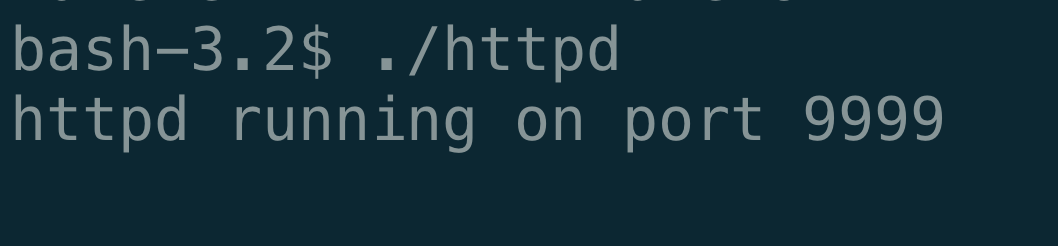


表 14运行程序

### 验证

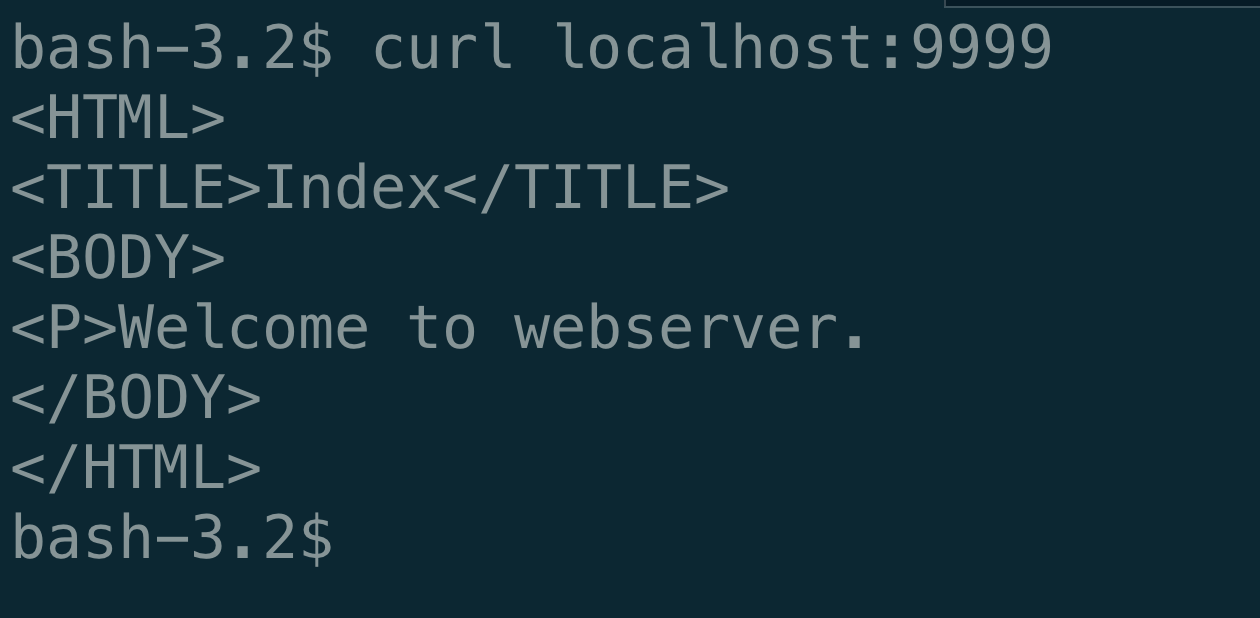


表 15curl验证

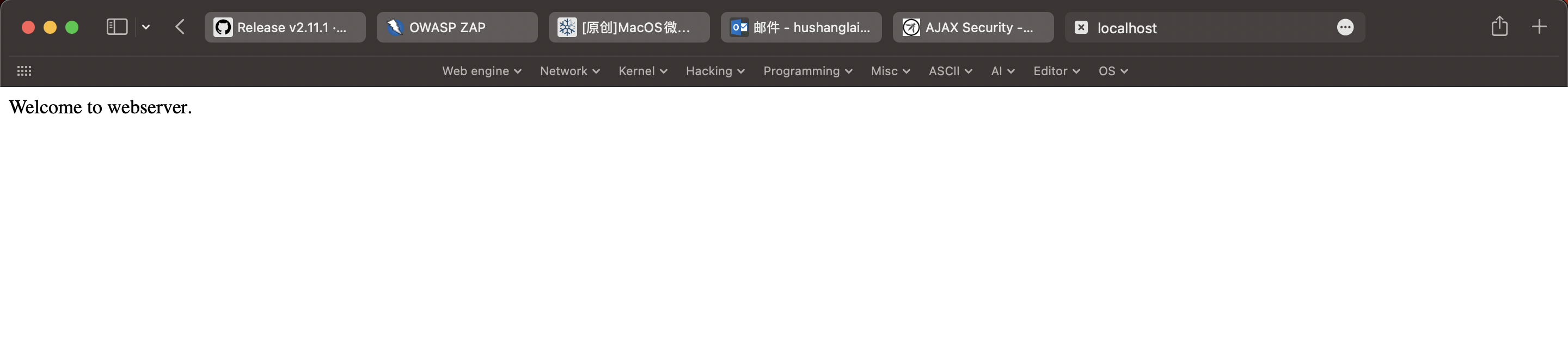


表 16浏览器验证

## 实验总结

通过本次试验，熟练掌握了Linux下面的网络编程。学习了HTTP服务器的工作原理和通用网络编程框架即，使用while无限循环和accept函数等待客户端连接，然后使用pthread\_create函数创建线程处理客户发送过来的http报文，对每个报文做出相应的处理。