天津大学
Tianjin University

Socket网络编程

2024.5.15

天津大学
Tianjin University

目录

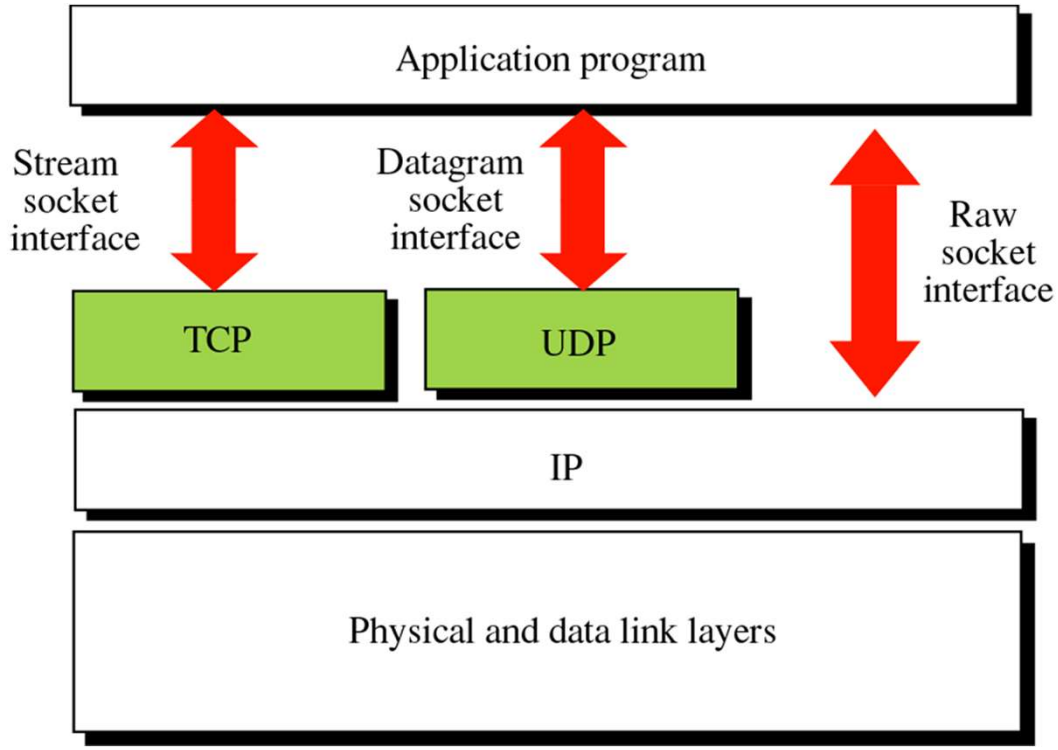
CONTENT

天津大学
Tianjin University

01 Socket简介

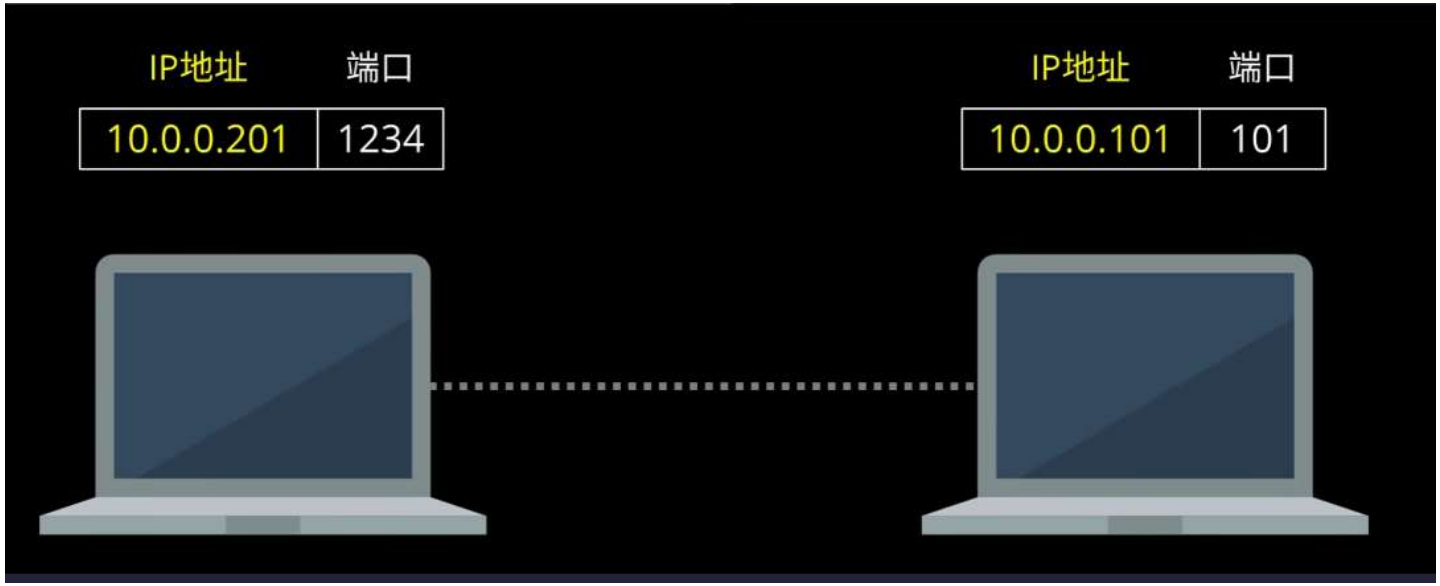
Socket的基本概念

- Socket是一组API，专为不同主机间的通信设计。它位于TCP/IP协议栈之上，提供了一种方式，让我们能够在网络中发送和接收数据。



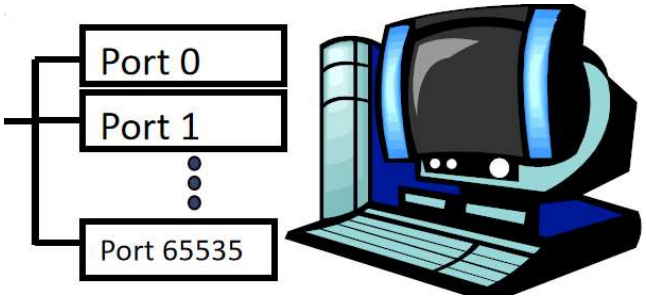
Socket的工作原理

- Socket = IP地址+端口号



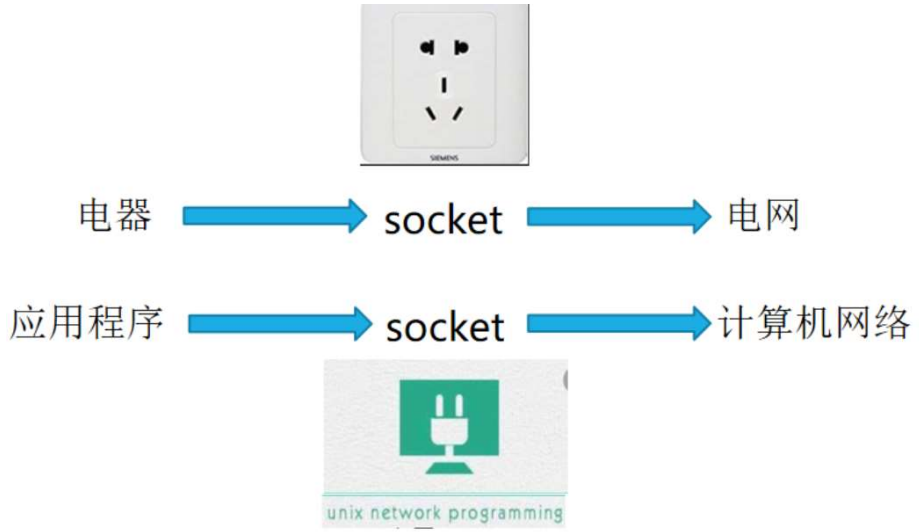
关于端口号

- 每个主机有65536个端口号（2^16）
- 其中一些端口号被保留给常用的程序和服务，被称为“熟知端口号
  20 21: FTP
  23: Telnet
  80: HTTP
  443：HTTPs
- 在写其它的服务程序时应该避开这些端口，通常从1024开始



实现通信

- 通过socket接口应用程序可以：
  - 向网络上的应用发送数据
  - 接收其它应用发来的数据



两种类型的Socket

- SOCK_STREAM
  - 基于TCP协议
  - 可靠数据传输
  - 面向连接
  - 全双工
- SOCK_DGRAM
  - 基于UDP协议
  - 不可靠数据传输
  - 面向消息(没有连接)
  - 能够发送或接收数据

天津大学
Tianjin University

02 使用Socket API进行网络编程

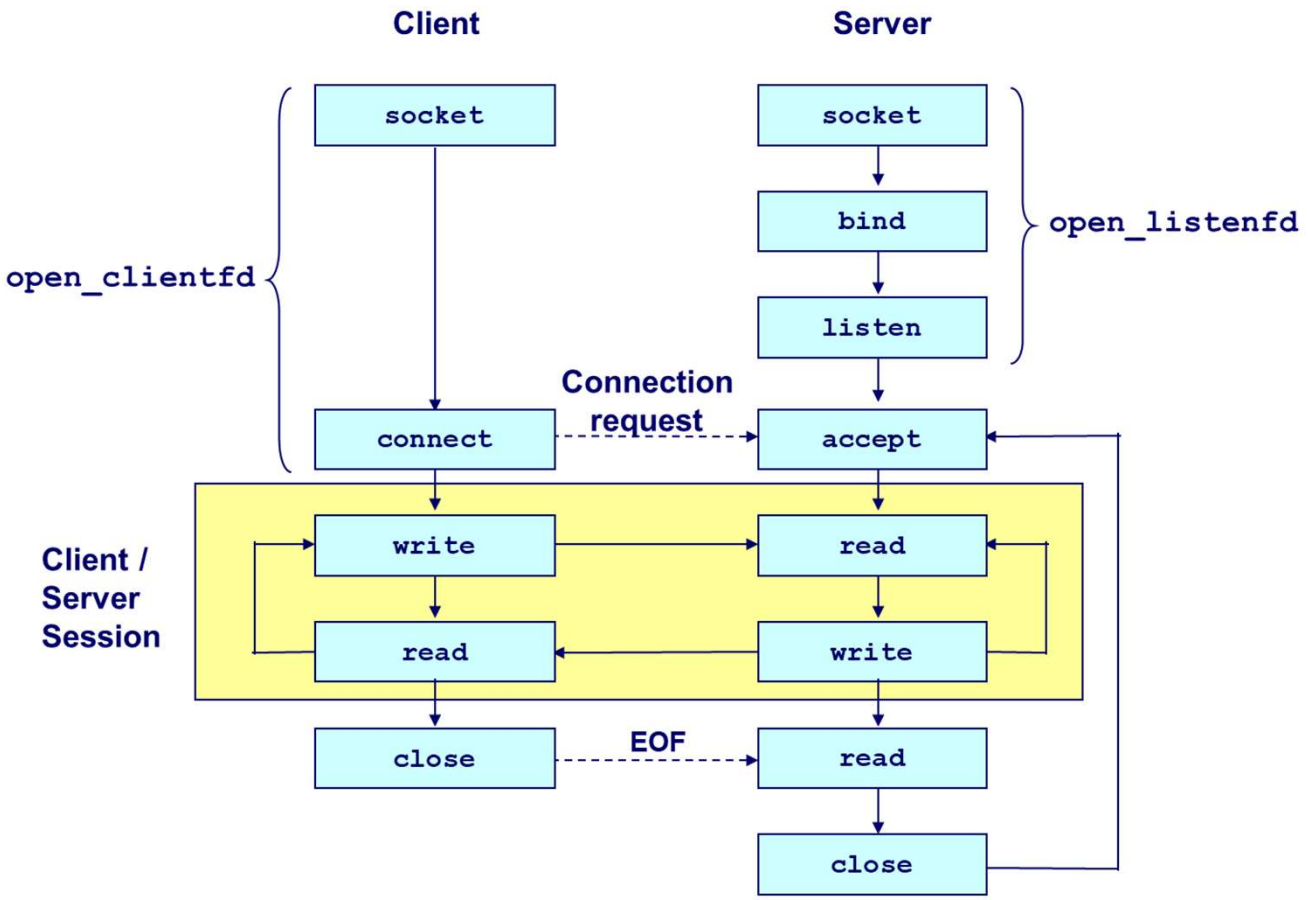
如何使用socket API

1. 初始化
   - socket()方法
   - 设置远程主机在网络中的哪个位置(IP address, hostname)
   - 设置应该把数据传送到远程主机的哪个程序/服务(port)
2. 收发数据
   - 和其他的/O操作类似
   - send <---> write
   - read <---> recv
3. 关闭
   - close()方法
   - 退出server程序
   - 解除端口占用

如何使用socket API

1. 初始化
   - socket()方法
   - 设置远程主机在网络中的哪个位置(IP address, hostname)
   - 设置应该把数据传送到远程主机的哪个程序/服务(port)
2. 收发数据
   - 和其他的/O操作类似
   - send <---> write
   - read <---> recv
3. 关闭
   - close()方法
   - 退出server程序
   - 解除端口占用

TCP socket


Client
Server
socket
socket
bind
listen
open_listenfd
open_clientfd
Connection
request
connect
accept
Client /
Server
Session
write
read
read
write
close
EOF
read
close

初始化(创建、绑定与监听)

1. 创建
   - sock = socket(addr_family, type)
     - addr_family: ip地址类型
       - AF_INET　　IPv4
       - AF_INET6　　IPv6
     - type: socket类型
       - SOCK_STREAM　　TCP
       - SOCK_DGRAM　　UDP
2. 绑定
   - bind(sock, addr, addrlen)
     - sock：socket函数调用返回的套接字
     - addr：地址结构体，调用bind之后这个地址与参数sockfd指定的套接字关联
     - addrlen: addr的长度

初始化(创建、绑定与监听)

3. 监听
   - listen(sock, backlog)
     - scok： socket函数调用返回的套接字
     - backlog: 指定排列在队列中的最大的连接数量,不得小于1.

- 至此,服务端已经完成初始化操作，客户端的初始化只需要调用socket()方法，不需要进行绑定和监听

收发数据(建立连接)

1. 客户端:发送连接请求
   - connect(sock, server_addr, server_ addrlen)
     - scok： socket函数调用返回的套接字
     - server_addr：服务端的地址
     - server_ addrlen: 服务端地址的长度
- 服务端被动等待连接，客户端主动发起连接
2. 服务端:接受连接请求
   - conn = accept(sock, client_addr, clinet_ addrlen)
     - conn: 新的socket对象，用来在新建的连接上发送和接收数据
     - client_addr：客户端的地址
     - client_ addrlen: 客户端地址的长度
- accept方法是阻塞的，并且可以重复调用

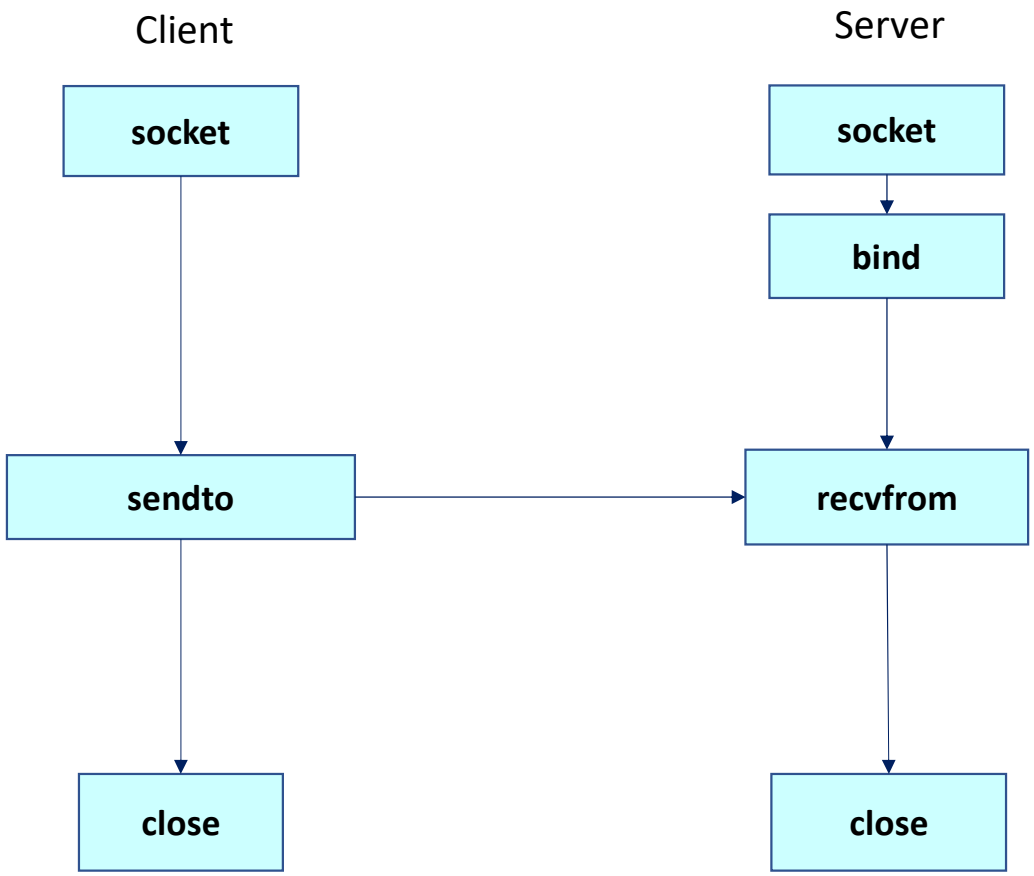
收发数据

1. 发送数据
   - count = send(sock, data, data_len, flag)
     - sock:发送数据的套接字
     - data：要发送的数据
     - data_len：数据的字节数
     - flag：一组指定调用方式的标志，一般设置为0
2. 接收数据
   - count = recv(sock, buf, buf_size, flag)
     - sock: 接收数据的套接字
     - buf：接收数据的缓存
     - buf_size：缓存大小
     - flag：一组指定调用方式的标志，一般设置为0

关闭

1. close(sock)
   - 关闭socket
   - 关闭之后该socket对象的所有操作将不可用

UDP socket


Client
Server
socket
socket
bind
sendto
recvfrom
close
close

收发数据

1. 发送数据
   - sendto(sock, buf, len, flag, dest_addr, addrlen)
     - sock:是由socket()调用返回的并且未作连接的套接字描述符（套接字号）
     - buf：发送缓冲区，往往是使用者定义的数组，该数组装有要发送的数据
     - len：发送缓冲区的大小，单位是字节
     - flag：一组指定调用方式的标志，一般设置为0
     - dest_addr:指向接收数据的主机地址信息的结构体，也就是该参数指定数据要发送到哪个主机哪个进程

收发数据

2. 接收数据
   - recvfrom(sock, buf, buf_size, flag, src_addr, addrlen)
     - sock:是由socket()调用返回的并且未作连接的套接字描述符（套接字号）
     - buf：接收缓冲区，往往是使用者定义的数组，该数组装有接收到的数据
     - len：接收缓冲区的大小，单位是字节
     - flag：一组指定调用方式的标志，一般设置为0
     - src_addr:指向发送数据的主机地址信息的结构体，也就是我们可以从该参数获取到数据是谁发出的

Socket API总结

- Sockets
  - socket setup
  - I/O
    - send & recv
    - sendto & recvfrom
  - Close
- TCP
  - Client: socket()->connect()->send()/recv()->close()
  - Server: socket()->bind()->listen()->accept()->send()/recv()->close()
- UDP
  - Client: socket()->sendto->close()
  - Server: socket()->bind()->recvfrom()->close()

天津大学
Tianjin University

03 Socket API 网络编程基础代码

基础代码目录说明

- 网络客户端和服务器代码
  - .../echo_client.c - Simple echo network client
  - .../echo_server.c - Simple echo network server
- 编译规则文件
  - .../Makefile - Contains rules for make
- 示例驱动和解析器代码
  - .../example.c - Example driver for parsing
  - .../lexer.l - Lex/Yacc related logic
  - .../parse.y
  - .../parse.c
  - .../parse.h
- 示例HTTP请求
  - .../sample_request_simple - Example HTTP requests
  - .../sample_request_realistic

天津大学
Tianjin University

04 Socket编程实验作业

任务要求

- 搭建编程环境
- 学习分词方法lex和yacc
- 实现简单的echo web server：
  - Server收到client的带多请求行的消息后，能够正确解析出来，并且返回响应消息（response message）。分以下3种情况处理：
    - Echo: 如果收到客户端发来的是GET, HEAD和POST方法，则echo回去，即重新封装（encapsulation）消息并返回给客户端。
    - 没实现：如果收到客户端发来的是除GET, HEAD和POST以外的其它方法，服务器并没有实现，则需要返回响应消息“HTTP/1.1 501 Not Implemented\r\n\r\n”。
    - 格式错误：如果收到的客户端消息的格式错误，应能够识别出来，并返回错误代码为400的HTTP响应消息“HTTP/1.1 400 Bad request\r\n\r\n”。

具体要求以**实践指导书**为准

任务要求

- 实验安排:
  - 项目开始
    - 项目于5月15日正式开始

- 工作进度报告
  - 项目开始后，须在规定时间在智慧树平台提交工作进度报告

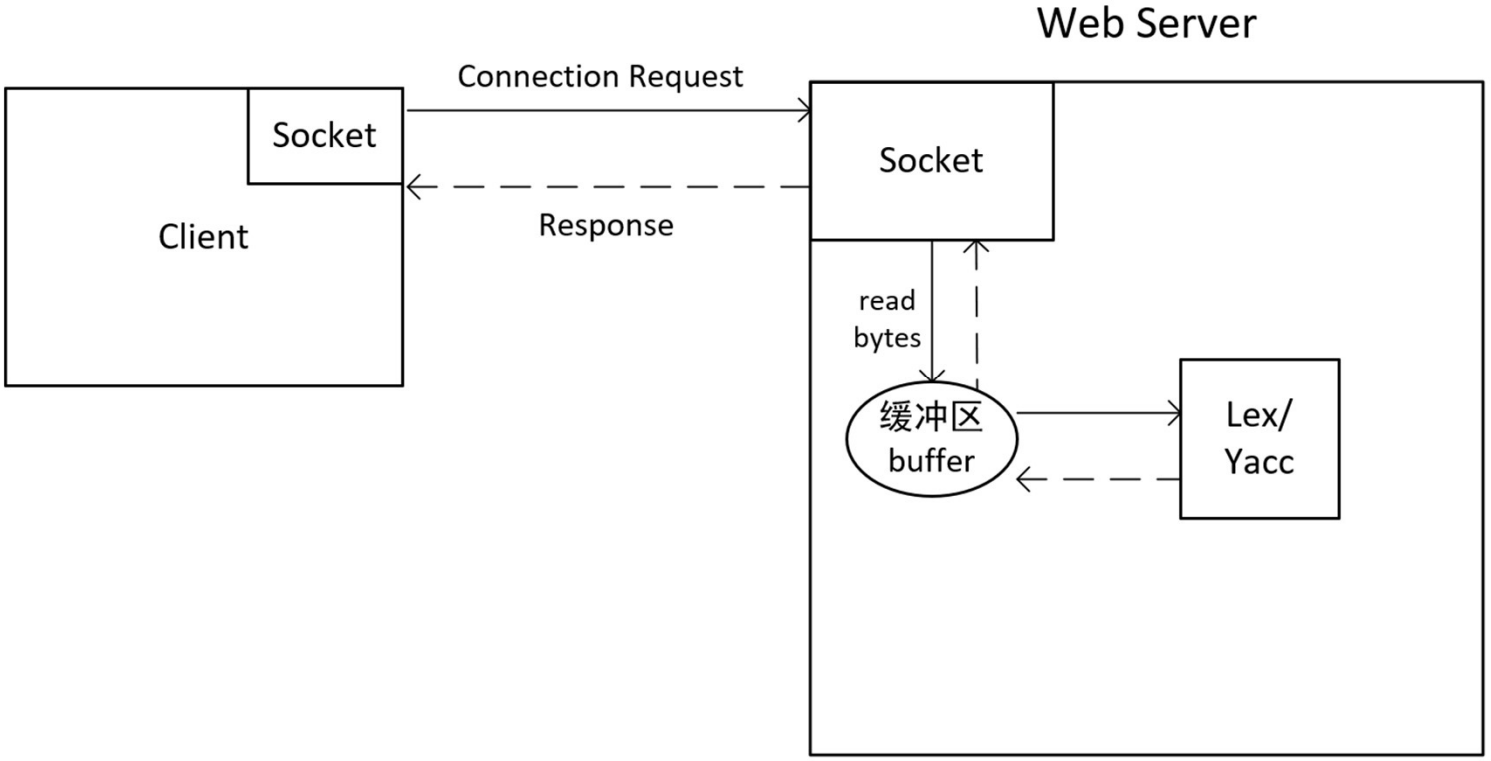
任务要求

- 关键时间节点：
  - 5月15日：项目启动
  - 5月26日：提交第1周进度报告。
  - 6月2日：提交第2周进度报告。
  - 6月9日：提交第3周进度报告。
  - 6月16日：提交第4周进度报告。
  - 6月23日：提交实验报告和最终源码。

解析消息

Web Server 工作流程:

1. 创建Socket套接字，在网络中监听来自客户端的连接请求
2. 将请求读取到缓冲区buffer中
3. 应用Lex/Yacc规则
   buffer中的数据将被送入Lex/Yacc工具处理。
4. 解析结果处理
   服务器对解析后的请求数据进行要求进行处理，并写入到响应消息中
5. 响应
   将响应消息发送回客户端



Lex/Yacc概述

- Lex：定义词法分析器
  - Lex是一个程序生成器，用于创建词法分析器。
- Yacc：定义语法分析器
  - Yacc提供了一种机制来处理基于规则的语法分析，它使用从Lex传递过来的tokens来构建语法树。

```
%{
定义词法单元，可以用 #define 表示
%}
声明部分
%%
转换规则
%%
辅助函数
```

Lex/Yacc 格式介绍

任务要求

- 主要需要修改的文件:

  echo_server.c：接收到消息后需并解析后需按要求响应

  echo_client.c：将输入参数修改为文件形式

  parse.c、parse.y：能够利用lex和yacc正确解析消息（难点。需要了解lex和yacc，并按照标准正确解析）

任务要求

测试结果：

- 正确GET请求

samples > request_get

```
GET / HTTP/1.1
Host: www.cs.cmu.edu
Connection: keep-alive
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Accept-Encoding: gzip, deflate, sdch
Accept-Language: en-US,en;q=0.8
```

```
root@2073dface794:/home/project-1# ./echo_client 127.0.0.1 9999 ./samples/request_get
Sending GET / HTTP/1.1
Host: www.cs.cmu.edu
Connection: keep-alive
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/39.0.2171.99 Safari/537.36
Accept-Encoding: gzip, deflate, sdch
Accept-Language: en-US,en;q=0.8


Received GET / HTTP/1.1
Host: www.cs.cmu.edu
Connection: keep-alive
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/39.0.2171.99 Safari/537.36
Accept-Encoding: gzip, deflate, sdch
Accept-Language: en-US,en;q=0.8
```

任务要求

测试结果：

- 正确POST请求

samples > request_post

```
1  POST / HTTP/1.1
2  Host: www.cs.cmu.edu
3  Connection: keep-alive
4  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5  User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
6  Accept-Encoding: gzip, deflate, sdch
7  Accept-Language: en-US,en;q=0.8
```

```
root@2073dface794:/home/project-1# ./echo_client 127.0.0.1 9999 ./samples/request_post
Sending POST / HTTP/1.1
Host: www.cs.cmu.edu
Connection: keep-alive
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/39.0.2171.99 Safari/537.36
Accept-Encoding: gzip, deflate, sdch
Accept-Language: en-US,en;q=0.8


Received POST / HTTP/1.1
Host: www.cs.cmu.edu
Connection: keep-alive
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/39.0.2171.99 Safari/537.36
Accept-Encoding: gzip, deflate, sdch
Accept-Language: en-US,en;q=0.8
```

任务要求

测试结果：

- 正确HEAD请求

samples > request_head

```
1  HEAD / HTTP/1.1
2  Host: www.cs.cmu.edu
3  Connection: keep-alive
4  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5  User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
6  Accept-Encoding: gzip, deflate, sdch
7  Accept-Language: en-US,en;q=0.8
```

```
root@2073dface794:/home/project-1# ./echo_client 127.0.0.1 9999 ./samples/request_head
Sending HEAD / HTTP/1.1
Host: www.cs.cmu.edu
Connection: keep-alive
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/39.0.2171.99 Safari/537.36
Accept-Encoding: gzip, deflate, sdch
Accept-Language: en-US,en;q=0.8


Received HEAD / HTTP/1.1
Host: www.cs.cmu.edu
Connection: keep-alive
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/39.0.2171.99 Safari/537.36
Accept-Encoding: gzip, deflate, sdch
Accept-Language: en-US,en;q=0.8
```

任务要求

测试结果：

- 格式错误

samples > request_400

```
GET
/~prs/15-441-F15/ HTTP/1.1
Host: www.cs.cmu.edu
Connection: keep-alive
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Accept-Encoding: gzip, deflate, sdch
Accept-Language: en-US,en;q=0.8
```

```
root@2073dface794:/home/project-1# ./echo_client 127.0.0.1 9999 ./samples/request_400
Sending GET
/~prs/15-441-F15/ HTTP/1.1
Host: www.cs.cmu.edu
Connection: keep-alive
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/39.0.2171.99 Safari/537.36
Accept-Encoding: gzip, deflate, sdch
Accept-Language: en-US,en;q=0.8


Received HTTP/1.1 400 Bad request
```

任务要求

测试结果：

- 未实现

samples > request_501

```
1  HAHA /~prs/15-441-F15/ HTTP/1.1
2  Host: www.cs.cmu.edu
3  Connection: keep-alive
4  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5  User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
6  Accept-Encoding: gzip, deflate, sdch
7  Accept-Language: en-US,en;q=0.8
```

```
root@9e22ef895173:/home/project-1# ./echo_client 127.0.0.1 9999 ./samples/request_501
Sending HAHA /~prs/15-441-F15/ HTTP/1.1
Host: www.cs.cmu.edu
Connection: keep-alive
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/39.0.2171.99 Safari/537.36
Accept-Encoding: gzip, deflate, sdch
Accept-Language: en-US,en;q=0.8


Received HTTP/1.1 501 Not Implemented
```

注意事项

- 每次改动文件后都要重新编译，若添加了文件需要改动Makefile增加相应链接文件
- 修改.y文件后需要删除y.tab.c、y.tab.h，否则使用make命令编译时可能不会重新生成它们
- 浏览器可以看到响应的包，可使用它们进行测试
- 要在Autolab 自动测试平台提交代码进行测试，交的代码格式要按照使用手册的要求，第一周提交不用包含static_site文件夹

天津大学
Tianjin University

谢谢垂听！

演讲人：崔宇鹏

2024.5.15