

# 天津大学

## 《计算机网络》课程设计报告 ——HTTP 的设计与实现



学 号 3019201274、  
3019234242  
姓 名 李嘉伟、曹颂  
学 院 求是学部  
专 业 智能机器平台  
年 级 2019  
任课教师 赵增华

2021 年 9 月 29 日

## 一、 报告摘要

基于 RFC2616 文件，编写 http/1.1 设计文档。并以此文档作为协议，在实现了 C/S 通信的基础上，能够正确封装并解析各种请求与响应，并且能对格式错误的请求和响应进行识别，返回相应的状态码。

## 二、 任务需求分析

使用语义分解工具语言正确解析收到的客户端的消息，基于http协议实现简单的echoWebServer。

表现为server收到client的消息后，能够正确解析出来，并且返回响应消息（responsemessage）。测试时分以下3种情况处理：

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。

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“501MethodUnimplemented”。

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。

## 三、 协议设计

根据任务要求设计协议。包括总体设计、数据结构设计和协议规则设计。

### 总体设计：

在实现了 C/S 通信的基础上，须以 http 协议为基础，正确封装并解析请求与响应。具体设计包括：

客户端中须有将 echo 消息内容封装成 request 结构体的功能，以及将接收到的 response 结构体解析并提取状态码以及收到的响应正文的功能。

服务器端须有将从端口收到的 request 请求解析，识别请求方法与请求正文，并将请求正文打包成 response 结构体并发送回 client 端的功能。

### 数据结构设计：

http 请求由三部分组成，分别是：请求行、消息报头、请求正文

1.请求行以一个方法符号开头，以空格分开，后面跟着请求的 URI 和协议的版本，格式如下：

Method Request-URI HTTP-Version CRLF 其中：

Method 表示请求方法；

Request-URI 是一个统一资源标识符；

HTTP-Version 表示请求的 HTTP 协议版本；

CRLF 表示回车和换行。

2.请求方法有多种，标志着如何从服务器端操作数据。

3.请求报头允许客户端向服务器端传递请求的附加信息以及客户端自身的信息。

4.请求头和请求正文之间是一个空行，这个行非常重要，它表示请求头已经结束，接下来的是请求正文。请求正文中可以包含客户提交的查询字符串信息。

http 响应也是由三个部分组成，分别是：状态行、消息报头、响应正文

1.状态行格式如下：

HTTP-Version Status-Code Reason-Phrase CRLF 其中：

HTTP-Version 表示服务器 HTTP 协议的版本；

Status-Code 表示服务器发回的响应状态代码；

Reason-Phrase 表示状态代码的文本描述。

状态代码有三位数字组成，第一个数字定义了响应的类别，且有五种可能取值：

1xx：指示信息--表示请求已接收，继续处理

2xx：成功--表示请求已被成功接收、理解、接受

3xx：重定向--要完成请求必须进行更进一步的的操作

4xx：客户端错误--请求有语法错误或请求无法实现

5xx：服务器端错误--服务器未能实现合法的请求

2.响应报头允许服务器传递不能放在状态行中的附加响应信息，以及关于服务器的信息和对 Request-URI 所标识的资源进行下一步访问的信息。

**协议规则设计：**详细描述协议完成各个功能的协议规则。

http 请求方法规则：

GET 请求获取 Request-URI 所标识的资源

POST 在 Request-URI 所标识的资源后附加新的数据

HEAD 请求获取由 Request-URI 所标识的资源的消息报头

http 请求报头规则：

Accept 请求报头域用于指定客户端接受哪些类型的信息。

Authorization 请求报头域主要用于证明客户端有权查看资源。

Host 请求报头域主要用于指定被请求资源的 Internet 主机和端口号，它通常从 HTTP URL 中提取出来的。

http 响应状态码规则：

有三位数字组成，第一个数字定义了响应的类别，且有五种可能取值：

1xx：指示信息--表示请求已接收，继续处理

2xx：成功--表示请求已被成功接收、理解、接受

3xx：重定向--要完成请求必须进行更进一步的操作

4xx：客户端错误--请求有语法错误或请求无法实现

5xx：服务器端错误--服务器未能实现合法的请求

常见状态代码、状态描述、说明：

200 OK //客户端请求成功

400 Bad Request //客户端请求有语法错误，不能被服务器所理解

401 Unauthorized // 请求未经授权，这个状态代码必须和 WWW-Authenticate 报头域一起使用

403 Forbidden //服务器收到请求，但是拒绝提供服务

404 Not Found //请求资源不存在，eg：输入了错误的 URL

500 Internal Server Error //服务器发生不可预期的错误

503 Server Unavailable // 服务器当前不能处理客户端的请求，一段时间后，//可能恢复正常

响应报头规则：

Location 响应报头域用于重定向接受者到一个新的位置。Location 响应报头域常用在更换域名的时候。

Server 响应报头域包含了服务器用来处理请求的软件信息。与 User-Agent 请求报头域是相对应的。

## 四、 协议实现

1.对 parser.y 文件的修改：

定义 request\_headers 类,通过递归实现请求中附带多个或零个 header 行。

2.Echo\_server 的实现：

通过 parse 函数，将 client 发送的缓存区中的字符串转换成 Request 结构体，之后通过分析结构体的成员变量 http\_method，实现对 request 的识别和分析。同时需要在 echo\_server 内加入对库文件 parse.h 的引用，makefile 文件中规定的编译规则也需要做相应修改。

3.请求功能的实现

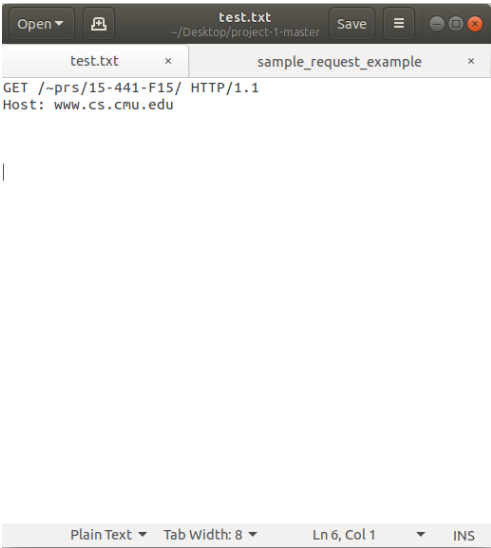
使用 echo\_client,读取文件中的字符串,之后通过 socket 发送给 echo\_server 进行测试，指令如下，其中 test.txt 为发送给服务器的字符串信息。

```
./echo_client 127.0.0.1 9999 test.txt
```

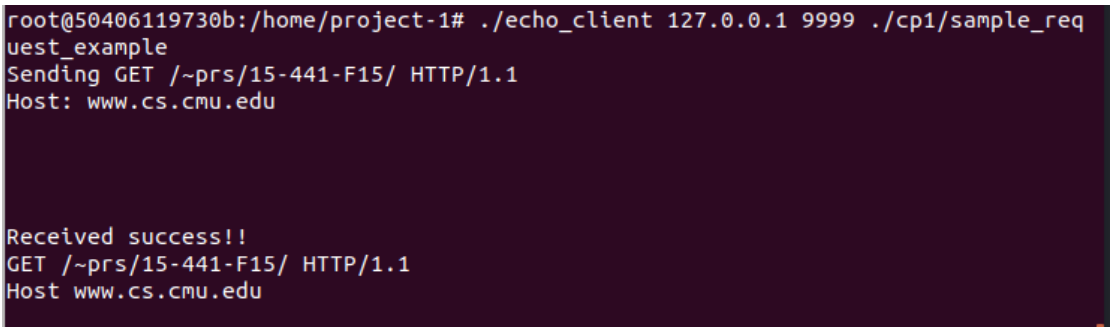
## 五、实验结果及分析

测试所实现协议的功能和性能，并对性能结果进行分析。需要针对考察点逐一展开功能 1 识别 GET、POST、HEAD 方法。

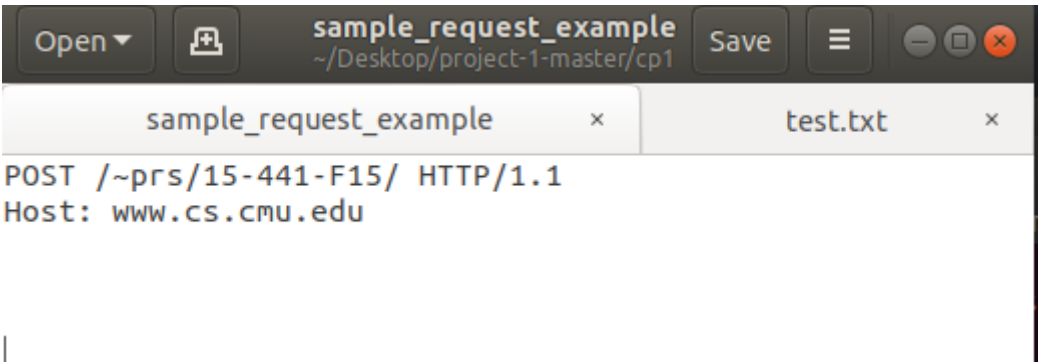
文件 1 内容如图



结果如图。



文件 2 内容如图。



结果如图

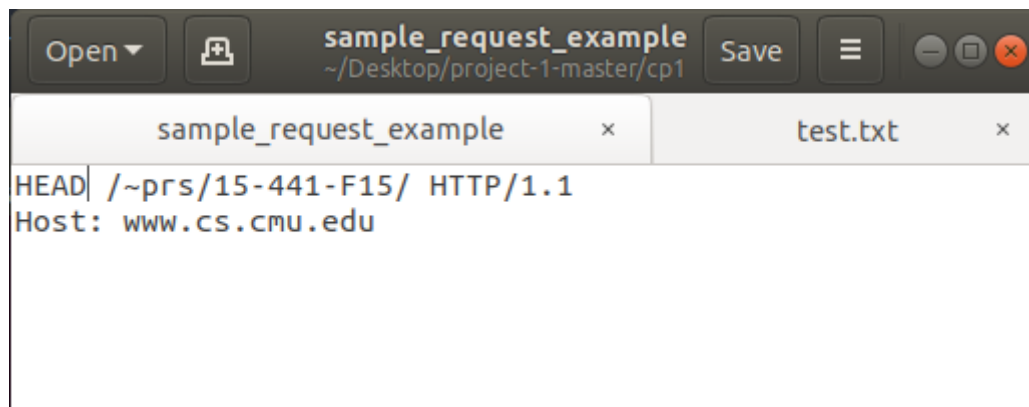
```

root@50406119730b:/home/project-1# ./echo_client 127.0.0.1 9999 ./cp1/sample_request_example
Sending POST /~prs/15-441-F15/ HTTP/1.1
Host: www.cs.cmu.edu

Received success!!
POST /~prs/15-441-F15/ HTTP/1.1
Host: www.cs.cmu.edu

```

文件 3 内容如图。



结果如图。

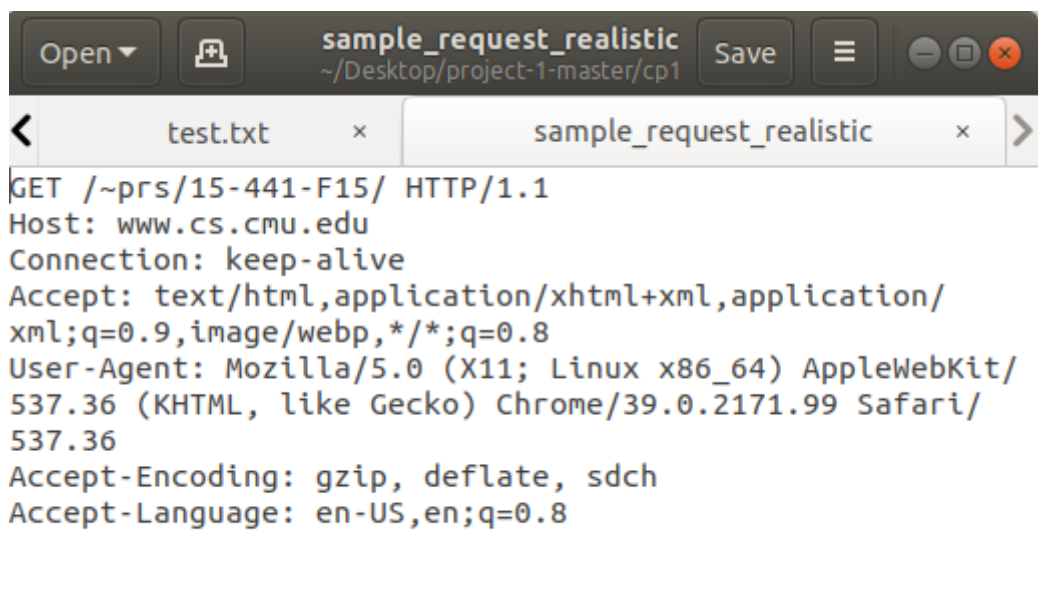
```

root@50406119730b:/home/project-1# ./echo_client 127.0.0.1 9999 ./cp1/sample_request_example
Sending HEAD /~prs/15-441-F15/ HTTP/1.1
Host: www.cs.cmu.edu

Received success!!
HEAD /~prs/15-441-F15/ HTTP/1.1
Host: www.cs.cmu.edu

```

文件 4 内容如图。

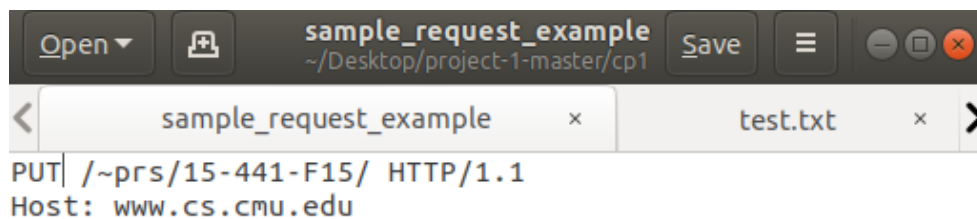


结果如图。

```
root@50406119730b:/home/project-1# ./echo_client 127.0.0.1 9999 ./cp1/sample_request_realistic
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 success!!
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
```

功能2 识别其他方法  
输入文件内容如图。



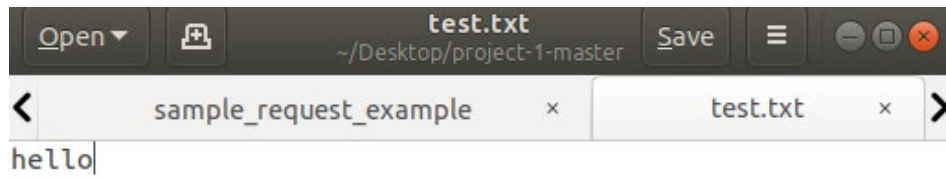
```
sample_request_example
~/Desktop/project-1-master/cp1
Save
sample_request_example × test.txt ×
PUT /~prs/15-441-F15/ HTTP/1.1
Host: www.cs.cmu.edu
```

结果如图。

```
root@50406119730b:/home/project-1# ./echo_client 127.0.0.1 9999 ./cp1/sample_request_example
Sending PUT /~prs/15-441-F15/ HTTP/1.1
Host: www.cs.cmu.edu

Received HTTP/1.1 501 Not Implemented
```

功能3 识别格式错误  
文件1 内容如图。

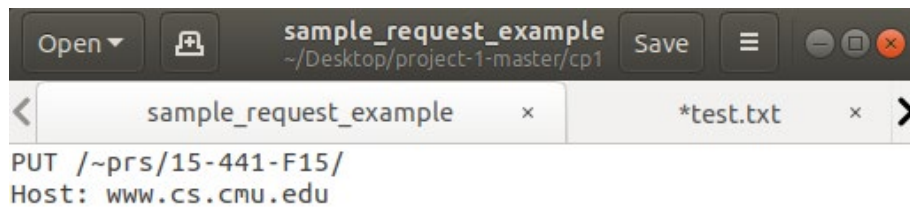


结果如图。

```
root@50406119730b:/home/project-1# ./echo_client 127.0.0.1 9999 ./cp1/sample_request_example
Sending hello

Received HTTP/1.1 400 Bad Request
```

文件 2 内容如图。



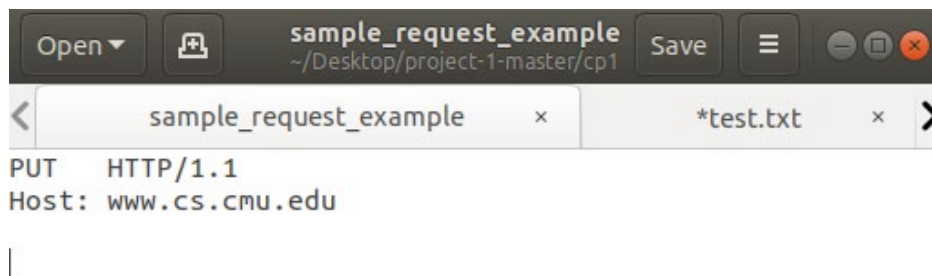
|

结果如图。

```
root@50406119730b:/home/project-1# ./echo_client 127.0.0.1 9999 ./cp1/sample_request_example
Sending PUT /~prs/15-441-F15/
Host: www.cs.cmu.edu

Received HTTP/1.1 400 Bad Request
```

文件 3 内容如图。



|

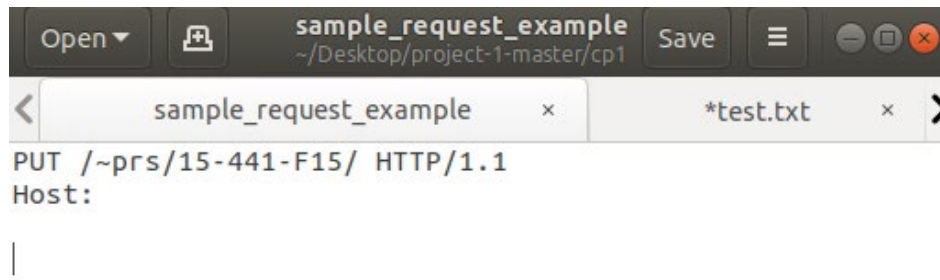


结果如图。

```
root@50406119730b:/home/project-1# ./echo_client 127.0.0.1 9999 ./cp1/sample_req
uest_example
Sending PUT HTTP/1.1
Host: www.cs.cmu.edu

Received HTTP/1.1 400 Bad Request
```

文件 4 内容如图。



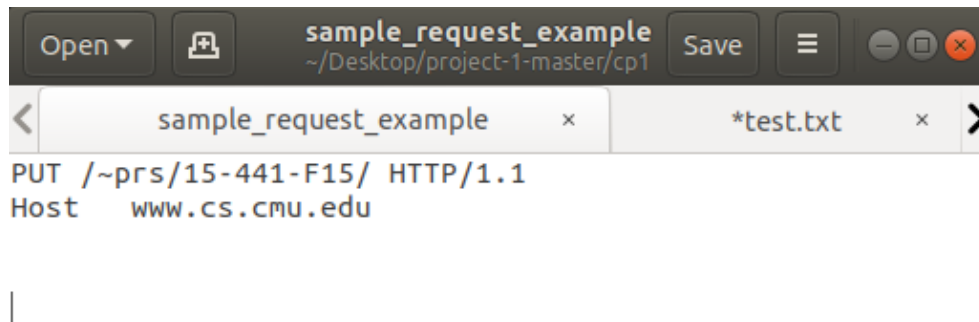
```
sample_request_example
~/Desktop/project-1-master/cp1
Save
sample_request_example x *test.txt x
PUT /~prs/15-441-F15/ HTTP/1.1
Host:
|
```

结果如图。

```
root@50406119730b:/home/project-1# ./echo_client 127.0.0.1 9999 ./cp1/sample_req
uest_example
Sending PUT /~prs/15-441-F15/ HTTP/1.1
Host:

Received HTTP/1.1 400 Bad Request
```

文件 5 内容如图。



```
sample_request_example
~/Desktop/project-1-master/cp1
Save
sample_request_example x *test.txt x
PUT /~prs/15-441-F15/ HTTP/1.1
Host www.cs.cmu.edu
|
```

结果如图。

```
root@50406119730b:/home/project-1# ./echo_client 127.0.0.1 9999 ./cp1/sample_req
uest_example
Sending PUT /~prs/15-441-F15/ HTTP/1.1
Host www.cs.cmu.edu

Received HTTP/1.1 400 Bad Request
```