**《计算机网络》课程设计**

**周进度报告**

**第三周：实现 HTTP 的并发请求**

**学 号 3020244294**

**姓 名 石子跃**

**学 院 智能与计算**

**专 业 网络安全**

**年 级 2020**

**任课教师 仇超**

**2022年 3月 30日**

# 一、任务需求分析

支持HTTP pipelining，支持持久连接（persistent conncetion）的客户端可以以管线的方式发送请求（即无须等待响应而发送多个请求）。服务器必须按接收请求的顺序发送响应。 假定以持久连接方式进行连接，并且假定在连接建立后进行管线方式请求的客户端应该准备去重新尝试连接如果首次管线请求方式尝试失败。如果客户端重新去尝试连接，那么，只有在客户端知道连接是持久连接之后，客户端才能进行管线发送请求。如果服务器在响应所有对应的请求之前关闭了连接，客户端必须准备去重新发送请求。客户端不应该利用非等幂的方法或者非等幂的方法序列进行管线方式的请求。否则一个过早的传输层连接的终止可能会导致不确定的结果。希望发送非等幂方法请求的客户端只有接收了上次它发出请求的响应后才能再次发送请求给服务器。

# 二、协议设计

在从client传入的数据中，按照每个HTTP请求的格式发现最后都以\r\n结尾。于是可以新建一个char指针去寻找其位置，并将中间的数据传入parse中进行解析，当服务器对当前请求进行响应后，将指针继续向后寻找，重复上述动作。在指针发现位置已经到达其尾部则退出当前循环，让server准备好recv接下来的请求。

# 三、协议实现

定义char \*index用来指向\r\n\r\n

使用Strstr函数来定位准确位置

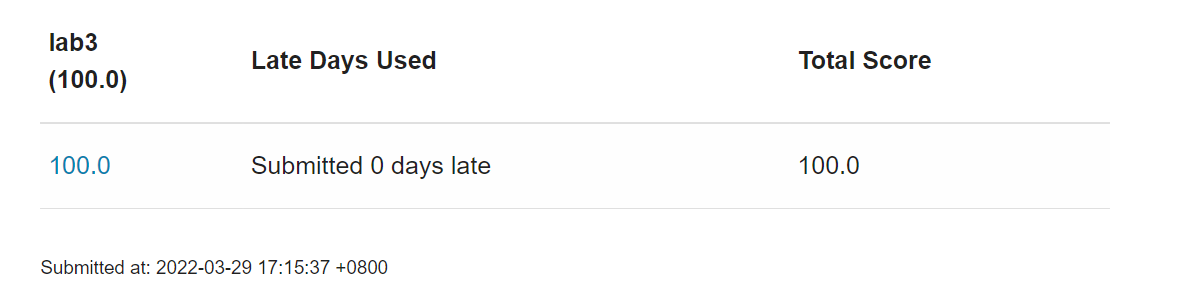
发现psrse函数还需传入一个int size于是需要一个long int 变量length来记录单独的一个请求报文长度。

注意每次提取request把\r\n\r\n这四个字节加上

While（当前读到的位置还未到此次接受的并发请求的长度）就继续协议设计中的步骤。

注意使用send函数时，发送的长度最好使用strlen(buf)如果直接传递BUF\_SIZE可能会因为后续客户端recv不够及时导致缓冲区内数据过多，后续的send函数不能进入。

# 三、实验结果及分析



测试平台100

在本地测试中，前几个请求会单独的通过不同的recv函数，但是当运行一段时间，会有多个响应报文被recv出来，具体原因还有待分析。

# 四、进度总结

**本周任务完成表**（以第一周为例）。

在“完成”“没完成”列对应打“√”

|  |  |  |  |
| --- | --- | --- | --- |
| 本周任务  要求 | 完成 | 没完成 | 备注 |
| 1. 服务器能连续响应客户端使用同一个 TCP 连接同时发送的多个请求  GET/HEAD/POST，即支持 HTTP pipelining  [6]。 | √ |  |  |
| 2. 服务器按照 RFC2616 规定的顺序处理 HTTP 的并发请求。 | √ |  |  |
| 3. 对于 HTTP 的并发请求，如果服务器认为其中一个请求是错误的并拒绝该请 求，那么服务器需要能够正确识别并解析出并发到达的下一条请求。 | √ |  |  |

**上周任务改进表**

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 上周任务 | 改进内容 | 备注 |
| 1 |  |  |  |
| 2 |  |  |  |