**《计算机网络》课程设计**



**周进度报告**

**第二周 实现 HEAD、GET 和 POST 方法**

**学 号 3020001267**

**姓 名 王旭**

**学 院 智能与计算学部**

**专 业 软件工程**

**年 级 2020**

**任课教师 杨亚平**

**2022年 4月 21日**

# 一、协议设计

1.完善服务器的功能，使其能够正确响应 HTTP1.1 请求消息，并按照 RFC 2616中的定义实现 HEAD、GET 和 POST 的持久连接，

1）如果收到客户端发来的 GET, HEAD 和 POST 方法，服务器按照

RFC2616 的规定进行处理并发回响应消息。

2）支持 4 种 HTTP 1.1 出错代码：400，404，501，505。能够准确判别

客户端消息，并发回响应消息。

3）妥善管理接收缓冲区，避免由于客户端请求消息太长导致的缓冲区溢

出问题。

2.服务器能够处理读写磁盘文件时遇到的错误（例如，权限、文件不存在、IO

错误等）。

3. 创建简化的日志记录模块，记录格式化日志。 使用日志记录程序运行的过程，方便调试和追踪。可以参考 Apache 的日志文件。

1) 按照 Apache 日志文件中“Error Log”的格式记录服务器的出错情况；

2) 按照 Apache 日志文件中“Access Log”的“Common Log Format”记

录服务器处理的请求；

3) 可以创建其它方便调试的日志（只是为了方便调试，不做格式要求，

不作为得分点）。

协议

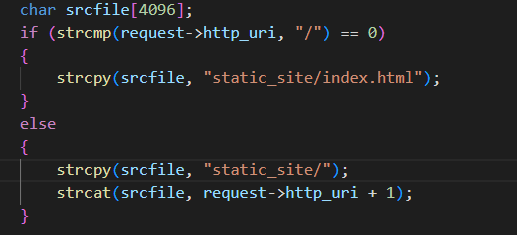
# 二、协议实现

详细描述本周功能实现的细节。主要功能模块使用流程图或者伪代码来辅助说明。禁止贴源码。

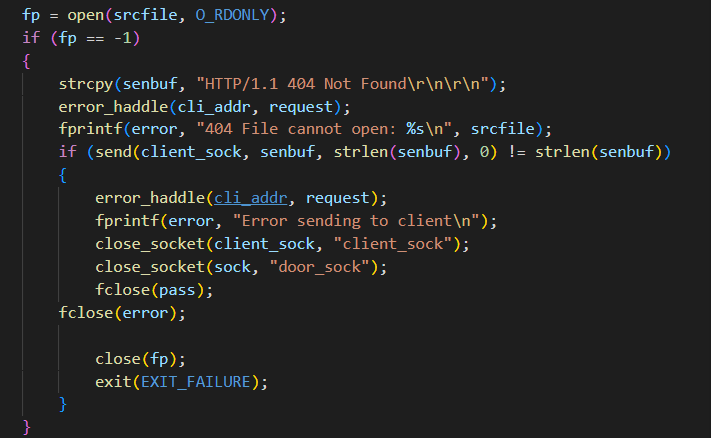
主要额外写了echo\_head,echo\_get处理head，get方法，而不再是简单的echo回去了

1.echo\_get

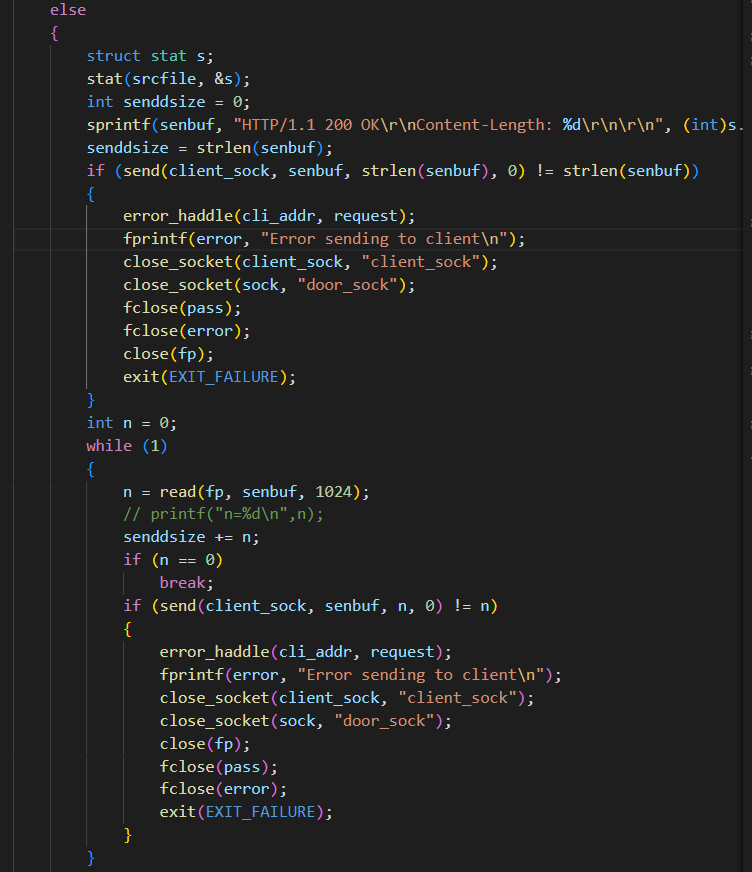
1）主要用request->http\_uri来判断，客户端请求了什么内容，要注意如果是/就说明请求的是默认的索引文件，然后把请求的文件都要加上网页存放的默认路径。



2）srcflie字符串得到了客户想要的文件路径后，用系统IO函数以二进制的形式打开，然后如果文件权限不行或者文件不存在或其他情况都是open函数返回-1，另外可以利用利用stat，得到它的st\_mode状态码获取文件具体打不开的原因，这个再进行后续的异常处理发送回400即可。



3）如果打得开，就把这个文件发送回去，但是要注意文件比较大的时候，一次发不完，应该分多次发送，我在这里的处理就是开了一个1024字节的数组，一次最多read1024字节，然后用一个n取它的返回值，每一次就发送n个字节过去，如果n=0说明取完了，正常返回就行。这里还要注意，要想网页打得开最好在前面发送一个指示文本大小的头部行，文本大小用stat函数判断即可。

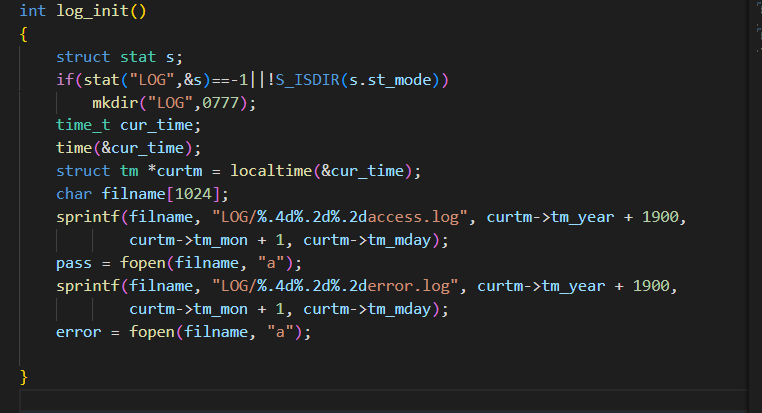


2.echo\_head，这个就是echo\_get不发送文本而已，与前面方法大同小异不在赘述。

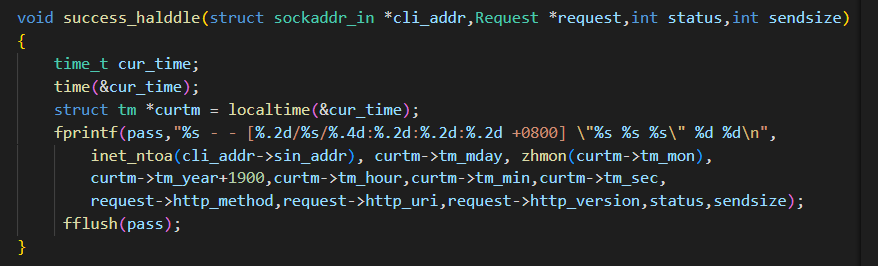
3.对于post的处理正常echo即可，这个第一章已经处理了。

4.比较重要的还有log的处理，分了access和fail的情况

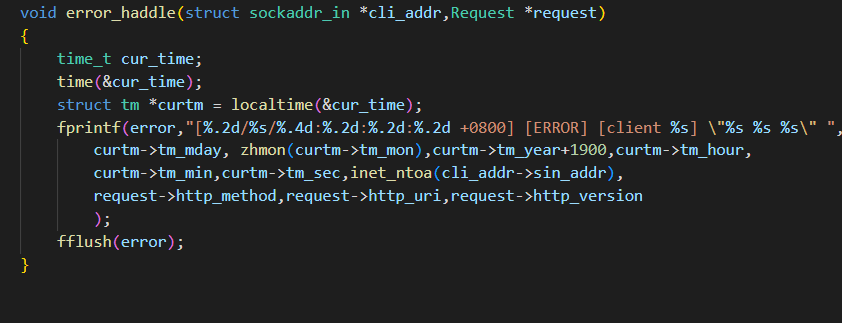
1）首先要初始化，要判断文件路径中有没有LOG这个文件夹，有可能没有这个文件夹或者有这个文件但不是文件夹这两种情况都mkdir一个，log都放到这个目录下。然后分别创建两个文件，一个是access的log，一个是error的log，以当天服务器开启的时间为命名。获取时间的方法就用time头文件中的time\_t，和转换用的time函数来。另外使用标准IO的fopen打开文件，因为这样可以用fprintf来输入，就要方便许多。



2）处理成功的情况，就用fprintf格式化输出一段信息就行了。



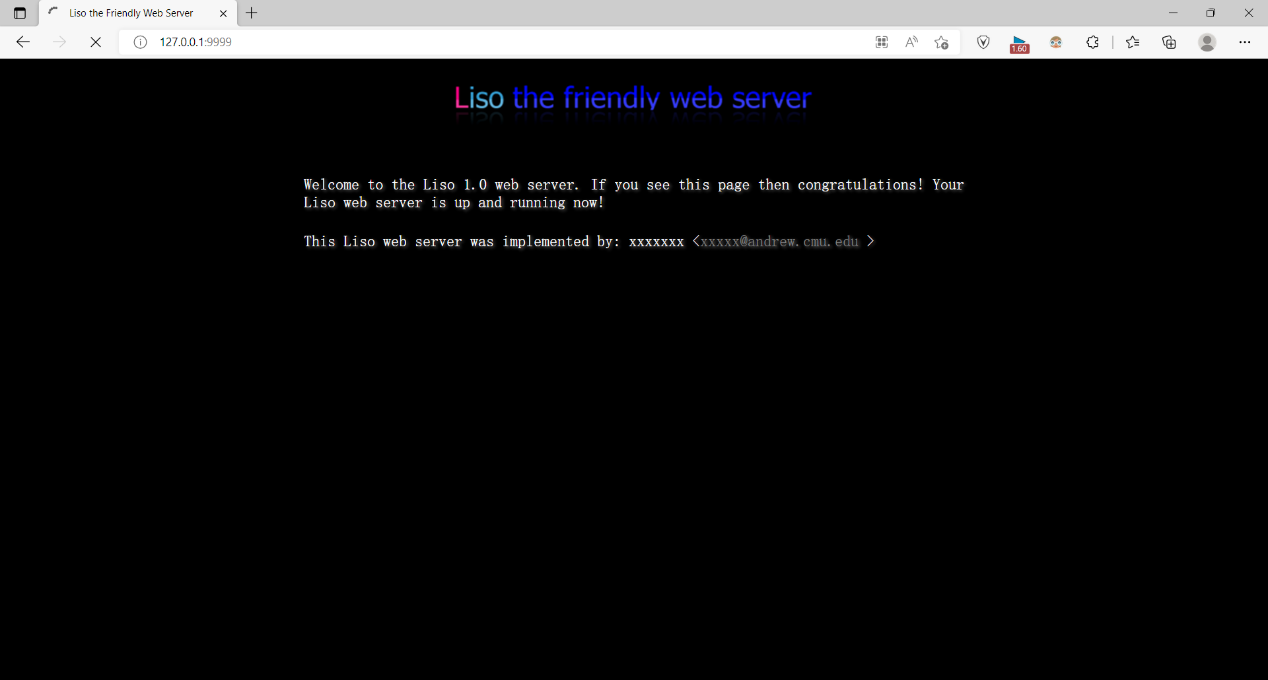
3）处理失败的情况，同理



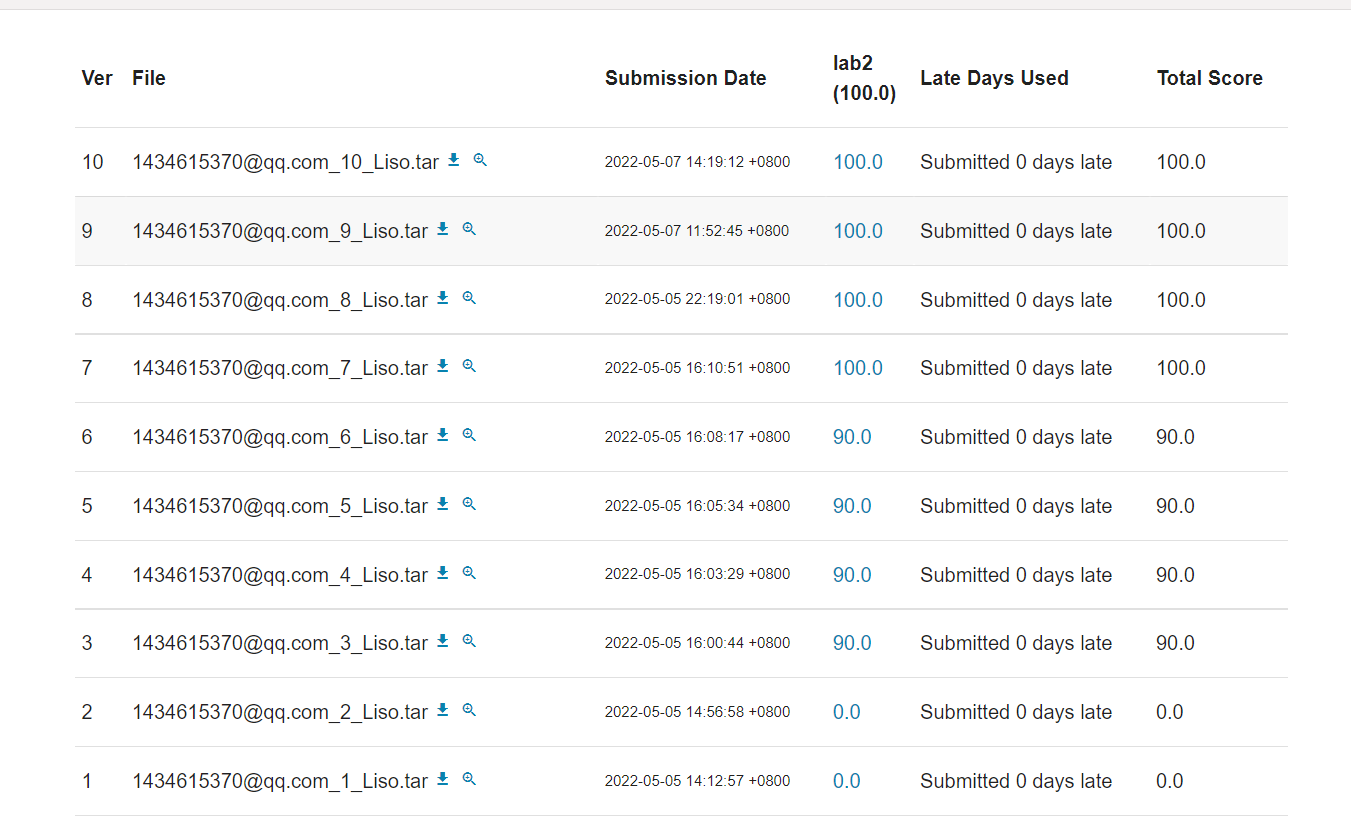
5.分文件编写，把之前涉及过的处理请求的函数全部放到一个文件haddle\_send.里面，在server调用这些函数，同理对于套接字的处理也一样，放到socket\_haddle里面，然后修改makefile，是的能够链接到。这样处理后，server只有92行，比之前少了不少，逻辑更加清晰了。

# 三、实验结果及分析

测试所实现任务的功能和性能，并对结果进行分析。需要针对考察点逐一展开。

经实测能够打开网页，

自动测试也成功。



# 四、进度总结

结合任务要求，填写以下2个表格：本周任务完成表和上周任务改进表。本周任务完成表用于总结说明本周完成任务的情况。上周任务改进表补充说明以前周次任务的改进情况。

注：如果以前的任务有改进，具体内容写进最后的课程设计报告内。不需要修改以前的周进度报告。

**本周任务完成表**。

在“完成”“没完成”列对应打“√”

|  |  |  |  |
| --- | --- | --- | --- |
| 本周任务  要求 | 完成 | 没完成 | 备注 |
| 1.1、如果收到客户端发来的 GET, HEAD 和 POST 方法，服务器按照  RFC2616 的规定进行处理并发回响应消息。 | √ |  |  |
| 1.2、服务器能够处理读写磁盘文件时遇到的错误（例如，权限、文件不存在、IO错误等）。 | √ |  |  |
| **1.3、**妥善管理接收缓冲区，避免由于客户端请求消息太长导致的缓冲区溢  出问题。 | √ |  |  |
| 2.、服务器能够处理读写磁盘文件时遇到的错误（例如，权限、文件不存在、IO  错误等）。 | √ |  |  |
| 3.1按照 Apache 日志文件中“Error Log”的格式记录服务器的出错情况； | √ |  |  |
| 3.2 按照 Apache 日志文件中“Access Log”的“Common Log Format”记  录服务器处理的请求； | √ |  |  |
| 4、功能测试 | √ |  |  |

**上周任务改进表**

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