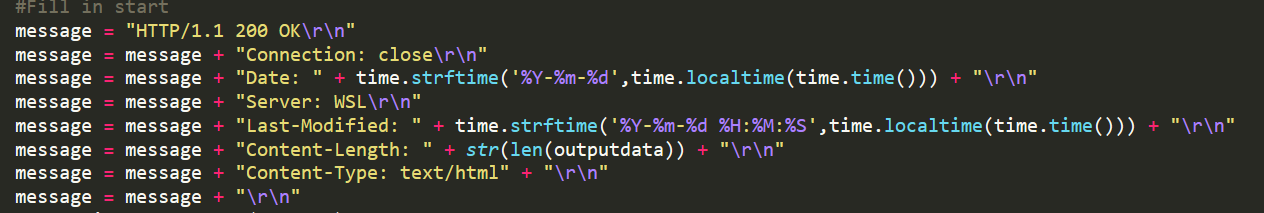
网络编程作业报告

516030910581 刘畅

由于这是我第一次使用socket库，所以费了一些功夫来学习，主要是用了<https://www.cnblogs.com/aylin/p/5572104.html>这篇博客的知识，在完成作业的过程中也受到了冯思远同学的许多帮助。Option的话除了POST都完成了。

1. Web-Server

这道题目本身不是很难，就是在熟悉socket的时候费了点劲，之外就是自己用http协议的样式填充了信息：



虽然有些差距，但是基本上就是正规网站的响应头了。

Option：

1. 多线程，使用了threading库来创建多线程，可以实现多个浏览器同时访问一个网站。
2. 手写的client，根据http协议写的请求头，可以正常访问网站。

2. UDPPing

在client中连续发送10个包，每个包利用settimeout函数规定1s内没有收到回应就视为丢包。

Option：

1. 利用每个包收到的时间和发送时间的间隔计算RRT，并且计算最大值，最小值和丢包率。
2. HeartBeat，client在发送数据包的同时发出自己发包的时间，server收到后通过包里的发出时间计算出RRT。我设计的client每次发10个包，间隔1s，一直持续发送。

3.SMTP

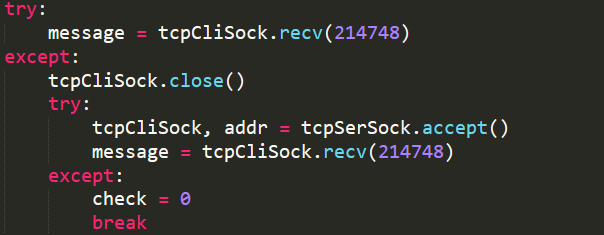
发邮件的过程中，首先是要用nslookup命令查出自己的邮箱对应的邮箱服务器，然后就用smtp协议中的规定运作即可。

Option：

1. 使用tls协议来加密，这里我本来打算用gmail，但是由于GFW的原因总是404，所以用了可以使用tls的qq邮箱，使用授权码作为加密防止信息泄露。
2. 发送图片的过程中使用MIME协议，将图片用base64编码后发出。同时发送文本和图片的话用—sjtu来进行分割。

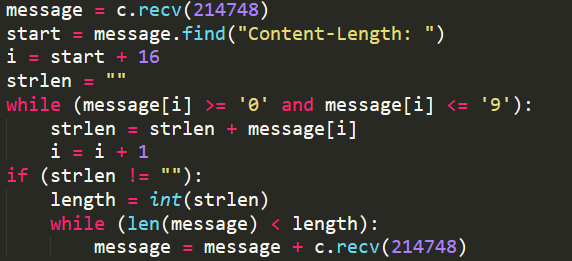
4.Proxy\_Server

这个代理服务器是我写了最久的一道题，花了我很长时间，最难的就是client和server之间的链接问题，中断还是持续链接有点困扰，所以我设计了一下，让两种方法都可以运行。



大概就是，如果client保持链接发信息的话，就接受；如果client中断了连接，新建链接来发信息，就等1s后中断旧的链接，自行等待新的连接。

除此之外，还有就是由于可能要加载图片，所以包可能会比较大，一次接受不完，就需要根据响应头上给的Content-Length来决定要不要继续接受信息。



Option:

1. 404可以正常显示
2. POST 为实现
3. CACHE实现了，访问过一次后会在本地存下响应头方便下次访问。