

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：

```
#fill in start
message = "HTTP/1.1 200 OK\r\n"
message = message + "Connection: close\r\n"
message = message + "Date: " + time.strftime('%Y-%m-%d',time.localtime(time.time())) + "\r\n"
message = message + "Server: WSL\r\n"
message = message + "Last-Modified: " + time.strftime('%Y-%m-%d %H:%M:%S',time.localtime(time.time())) + "\r\n"
message = message + "Content-Length: " + str(len(outputdata)) + "\r\n"
message = message + "Content-Type: text/html" + "\r\n"
message = message + "\r\n"
```

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。

```

try:
    message = tcpCliSock.recv(214748)
except:
    tcpCliSock.close()
    try:
        tcpCliSock, addr = tcpSerSock.accept()
        message = tcpCliSock.recv(214748)
    except:
        check = 0
        break

```

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。

```

message = c.recv(214748)
start = message.find("Content-Length: ")
i = start + 16
strlen = ""
while (message[i] >= '0' and message[i] <= '9'):
    strlen = strlen + message[i]
    i = i + 1
if (strlen != ""):
    length = int(strlen)
    while (len(message) < length):
        message = message + c.recv(214748)

```

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。