

## 【实验题目】应用层实验

【实验目的】掌握应用层的基本工作原理和实现方法

【实验工具】telnet 或 SecureCRT

(1) telnet 的使用说明:

如果在命令行不能运行 telnet, 则需要安装 telnet 客户端程序。

telnet 的安装方法见 <https://jingyan.baidu.com/article/3ea51489ba79e252e61bba97.html>

在控制台可以采用鼠标右键菜单粘贴, 拷贝要点击图标菜单“编辑/标记”

控制台修改缓冲区: 属性/布局/屏幕缓冲区大小/高度: 9999 窗口大小的宽度: 200

(2) SecureCRT 的使用说明

■ SecureCRT 下载: <http://172.18.187.9/netdisk/default.aspx?vm=l7net> (软件下载)

SecureCRT 下载解压后直接可以使用。

- 采用 SecureCRT 中的 telnet: 连接/sessions/点击右键/新建会话/协议: telnet/下一步/主机名: 域名或 IP 地址 端口号: 80 或 21 或 110 或 25/下一步/取名会话/完成。

- SecureCRT 设置中文显示: 选项/会话选项/外观/字体: 新宋体; 字符集: 中文 GB2312 字符编码: utf-8

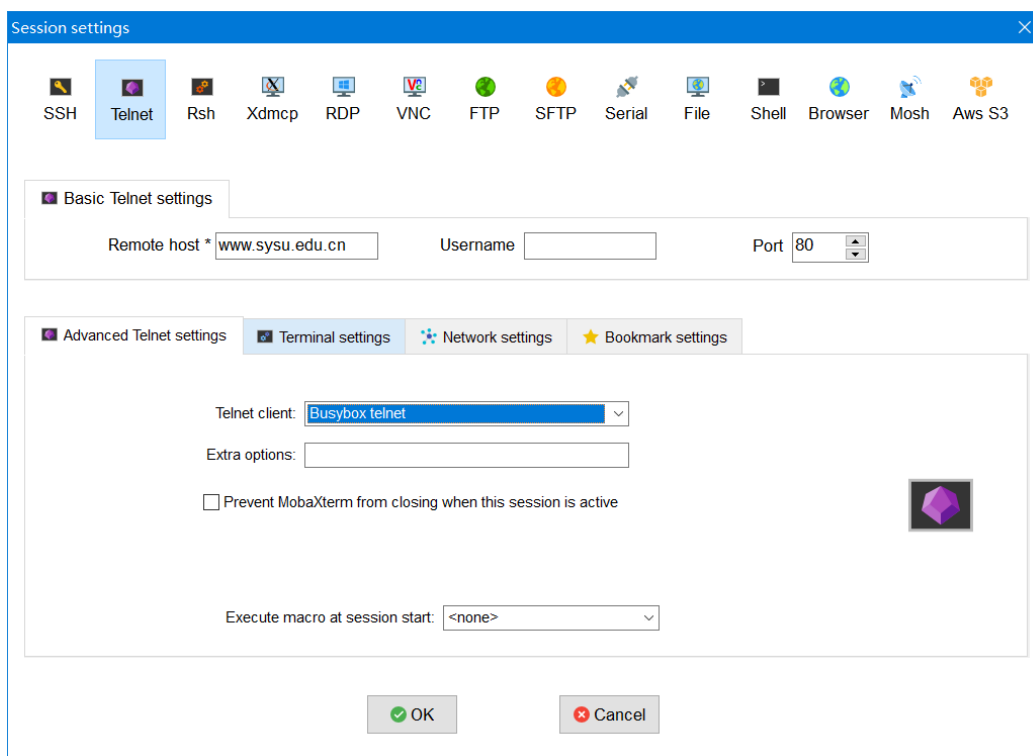
■ 为了可以看到更多的响应信息, 需要增加缓冲区大小。

- secureCRT: 会话选项/终端/仿真/回滚缓冲区: 10000
- 控制台: 属性/布局/屏幕缓冲区大小/高度: 9000

■ SecureCRT 的 SSL 配置--在 SecureCRT 中会话配置/连接/协议: Telnet/SSL

### (3) 我的实验工具

■ MobaXterm Personal Edition v10.9 中的 Busybox Telnet Client



## 【注意事项】

(1) 截屏时注意遮盖掉自己的邮箱密码

(2) sina.com 的邮箱的客户端访问设置:

**指定邮件服务器**

**选择 POP3 帐户**

☒ POP3 帐户 (P) ☐ HTTP 帐户 (T)

POP3 (PostOffice Protocol 3) 服务器是用来接收邮件的服务器，您的帐户上的邮件保存在此服务器上。如 public.guanagzhou.gd.cn

POP3 服务器 (P):  **输入 POP3 服务器地址**

POP3 帐户名 (A):  **输入 POP3 帐户名**

密码 (W):  **输入密码**

SMTP (Simple Mail Transfer Protocol) 服务器用来发送邮件和中转您发出的邮件。SMTP 服务器与 POP3 服务器可以不同。

SMTP 服务器 (S):  **输入 SMTP 服务器地址**

**点击“下一步”继续**

< 上一步 (B) **下一步 (N) >** 取消 帮助

- (3) QQ 邮箱需要在 Web 访问方式下开启 POP3 和 SMTP 服务才允许在客户端访问(设置/账号/开启 POP3/SMTP)，否则会出现错误“454 Authentication failed, please open smtp flag first!”

**POP3/IMAP/SMTP/Exchange/CardDAV/CalDAV 服务**

开启服务: ☒ POP3/SMTP 服务 (如何使用 Foxmail 等软件收发邮件?) ☐ IMAP/SMTP 服务 (什么是 IMAP, 它又是如何设置?) ☐ Exchange 服务 (什么是 Exchange, 它又是如何设置?) ☐ CardDAV/CalDAV 服务 (什么是 CardDAV/CalDAV, 它又是如何设置?)

(POP3/IMAP/SMTP/CardDAV/CalDAV 服务均支持 SSL 连接。如何设置?)

要在客户端访问，QQ 邮箱可能还会要求设置独立密码，即可以设置与 QQ 登录不同的密码。

**QQ 邮箱**

亲爱的用户:

在第三方客户端登录 QQ 邮箱时，需要开启对应的客户端协议服务。系统默认为“关闭”，你可以按照下面的引导手动设置独立密码，然后开启相应服务。

1. 登录 QQ 邮箱，在【设置】-【帐号】中找到并开启对应服务

**POP3/IMAP/SMTP/Exchange/CardDAV/CalDAV 服务**

开启服务:	POP3/SMTP 服务 (如何使用 Foxmail 等软件收发邮件?)	IMAP/SMTP 服务 (什么是 IMAP, 它又是如何设置?)	Exchange 服务 (什么是 Exchange, 它又是如何设置?)	CardDAV/CalDAV 服务 (什么是 CardDAV/CalDAV, 它又是如何设置?)
	已关闭   开启	已关闭   开启	已关闭   开启	已关闭   开启

(POP3/IMAP/SMTP/CardDAV/CalDAV 服务均支持 SSL 连接。如何设置?)

2. 设置邮箱独立密码

**设置独立密码 | 返回**

请设置你的独立密码

请输入你的独立密码

请再次输入独立密码

从 QQ 登录时: ☒ 始终需要输入独立密码 ☐ 通过 QQ 可以直接处理邮件，进入邮箱时才需输入独立密码

QQ 邮箱的安全可以通过独立密码得到充分保证。填写你的邮箱独立密码，设置成功后登录邮箱都需要独立密码验证，请牢记。了解更多

QQ Web 邮箱查看源码的方法:

QQ 邮箱

亲爱的用户:

在第三方客户端登录 QQ 邮箱时，需要开启对应的客户端协议服务。系统默认为“关闭”，你可以按照下面的引导手动设置独立密码，然后开启相应服务。

1. 登录 QQ 邮箱，在【设置】-【帐号】中找到并开启对应服务

**POP3/IMAP/SMTP/Exchange/CardDAV/CalDAV 服务**

开启服务:	POP3/SMTP 服务 (如何使用 Foxmail 等软件收发邮件?)	IMAP/SMTP 服务 (什么是 IMAP, 它又是如何设置?)	Exchange 服务 (什么是 Exchange, 它又是如何设置?)	CardDAV/CalDAV 服务 (什么是 CardDAV/CalDAV, 它又是如何设置?)
	已关闭   开启	已关闭   开启	已关闭   开启	已关闭   开启

(POP3/IMAP/SMTP/CardDAV/CalDAV 服务均支持 SSL 连接。如何设置?)

2. 设置邮箱独立密码

**设置独立密码 | 返回**

请设置你的独立密码

请输入你的独立密码

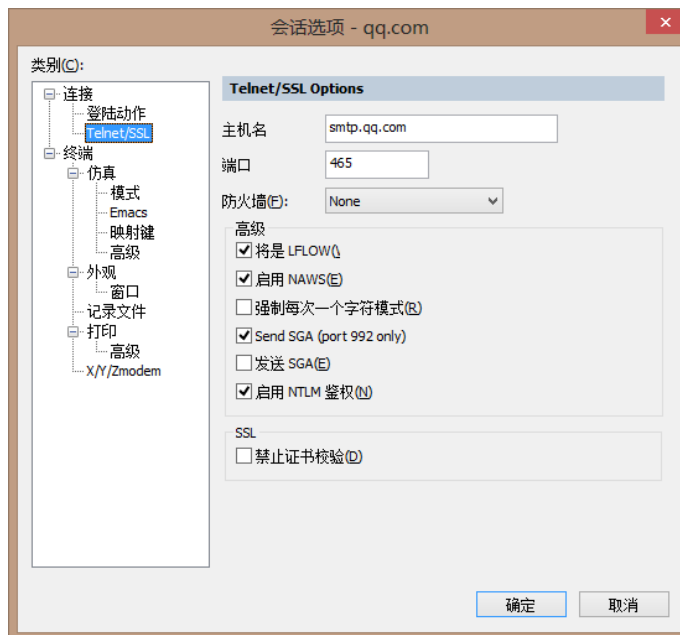
请再次输入独立密码

从 QQ 登录时: ☒ 始终需要输入独立密码 ☐ 通过 QQ 可以直接处理邮件，进入邮箱时才需输入独立密码

QQ 邮箱的安全可以通过独立密码得到充分保证。填写你的邮箱独立密码，设置成功后登录邮箱都需要独立密码验证，请牢记。了解更多

QQ 邮箱编程必须使用 Telnet/SSL 协议(加密模式):

SecureCRT 的 SSL 配置--在 SecureCRT 中会话配置/连接/协议: Telnet/SSL



QQ 邮箱的接收邮件服务器: pop.qq.com, 端口号为 995

QQ 邮箱的发送邮件服务器: smtp.qq.com, 端口号为 465

客户端登录的用户名为 QQ 号的 base64 编码

客户端登录的密码也采用 base64 编码

#### 【实验内容】

先认真阅读课件“Chapter2-application-layer.pdf”,再完成下面内容。注意:协议标准可以查阅 RFC;选取的内容不要与课件相同;响应内容太长时自己选取截取前后部分以及其中重点部分。

参考视频: <http://172.18.187.9/video/>

(选做)使用自己编写的 TcpClient 运行步骤一~步骤四,具体见步骤六。

如果不自己编写 TcpClient,可以尝试使用老师给的 TcpClient.exe 完成步骤一~步骤四,如果不愿意尝试,可以使用 telnet 来完成实验。

## 一、HTTP 协议

看完 HTTP 协议的课件后完成以下实验:

(1)从学院网站(sdc.s.sysu.edu.cn)找一网页下载。

http 请求:

将 Remote host 设置为“sdc.s.sysu.edu.cn”,端口号设置为 80。请求网页为“http://sdc.s.sysu.edu.cn/about/introduction”,使用客户端发送以下 HTTP 请求:

```
GET /about/introduction HTTP/1.1
Host: sdc.s.sysu.edu.cn
Connection: close
```

http 响应:

以下是 HTTP 响应报文的头部行,正文为 HTML 文档内容,已经省略。

```
HTTP/1.1 202 Accepted
Server: *****
Date: Thu, 04 Apr 2019 12:07:30 GMT
Content-Type: text/html; charset=utf-8
Transfer-Encoding: chunked
Connection: close
Pragma: no-cache
Cache-Control: no-cache
```

```
Expires: Thu, 04 Apr 2019 12:07:30 GMT
Set-Cookie:
FSSBB1l1UgzbN7N80T=4TG4usN7CGWnF47rRI8T9a34hsN6j1fAtoZ7pEgvJtzhCr0cG7hILGemEk0
ZmjP9FoAPlyI1s5ft8iq5fgTn4y93Ms43MQxWxe3P48lhugoIJ.PI7lWeBkPLgVh1KHk1nlNm5fAe
IDB.plFdPM623mBHWdXU5Ms3jU9BMchzPNiEPpJmBjjVC7zaiowk1WZBYTV; Path=/;
expires=Sun, 01 Apr 2029 12:07:30 GMT
Set-Cookie:
FSSBB1l1UgzbN7N80S=OhVkk8t8yD1VB_M_zPUVrYuJYS1CL7N00KiJCCAxn60TattckbwYPH00EM0
qWxcY; Path=/; expires=Sun, 01 Apr 2029 12:07:30 GMT; HttpOnly

... (响应报文正文已省略)
```

(2) 从学院网站 (sdcs.sysu.edu.cn) 找一图片下载。

http 请求:

我找了一张天河二号的照片, 图片 URL 为 “http://sdcs.sysu.edu.cn/sites/sdcs.sysu.edu.cn/files/574e48efa871f-300x199.png”, 因此向服务器发送如下 HTTP 请求:

```
GET /sites/sdcs.sysu.edu.cn/files/574e48efa871f-300x199.png HTTP/1.1
Host: sdcs.sysu.edu.cn
Connection: close
```

http 响应:

以下是 HTTP 响应报文的头部行, 正文为 PNG 图片的二进制信息, 显示为乱码, 已经省略。

```
HTTP/1.1 200 OK
Server: *****
Content-Type: image/png
Content-Length: 96721
Connection: close
Date: Thu, 04 Apr 2019 12:17:35 GMT
X-Content-Type-Options: nosniff
Last-Modified: Mon, 20 Feb 2017 13:08:34 GMT
ETag: "179d1-548f5f5f1b880"
Accept-Ranges: bytes
Set-Cookie:
FSSBB1l1UgzbN7N90T=4ahe4BbfPR7ixIOx_Xgm5v6UEwx7RLjjMlz9ffCkkgGiCF.1MPcDMSej
aEPipsoiupiwhnTIIjmHNTw1wvGiMf153P.NlO.CgvPJ6PDLKis._cwt1i3pIdKiUu3mMaxgsJ_
a5GEm.8A0ZEOWxd1wMWWB3qks_DmEFbDt2mi_5qhK0as4D.sJMYWlU.KH4Sez9TxxG; Path=/;
expires=Sun, 01 Apr 2029 12:12:45 GMT
Set-Cookie:
FSSBB1l1UgzbN7N80S=UJBjNf7pfZwz2SZDivBk_mDmd_sDlCr.IRQqjkKq78eUvGJsr5cKyoZa
AACHsXfo; Path=/; expires=Sun, 01 Apr 2029 12:12:45 GMT; HttpOnly

... (响应报文正文已省略)
```

(3) 在 http 请求的头部行中加入 If-Modified-Since: Fri, 16 Jan 2019 13:22:17 GMT 从学院网站下载 (2) 的图片。

http 请求:

```
GET /sites/sdcs.sysu.edu.cn/files/574e48efa871f-300x199.png HTTP/1.1
Host: sdcs.sysu.edu.cn
Connection: close
If-Modified-Since: Fri, 16 Jan 2019 13:22:17 GMT
```

http 响应:

在 HTTP 请求中加入了 If-Modified-Since 字段后, 服务器回应的 HTTP 相应不再返回图片的二进制信息, 而是返回 HTTP 状态码 304 Not Modified。

```
HTTP/1.1 304 Not Modified
Server: *****
Connection: close
Date: Thu, 04 Apr 2019 12:19:09 GMT
ETag: "179d1-548f5f5f1b880"
```

```
Set-Cookie:
FSSBB11UgzbN7N80T=4mILUzRNSwwf_rf0C0w7_XLaDJb2oR0uiVKGOReKej_9ssLRjlq4PUBm
D2YsRm1fq9KURGiMf152gQkc_yeHkt_9RJCGdrI8UCcLaNn5X.VqSXMRYvyZnKBDhbFU1ZoTjtA
NeCKGYyD39Pwgz1jLwtpwhny5ZFxLL1_5yPNKm3SIQApjSI5R1o1.CDI1e3AaaHf3; Path=/;
expires=Sun, 01 Apr 2029 12:14:19 GMT
Set-Cookie:
FSSBB11UgzbN7N80S=Eue2Nj01waBSIkjyH2Tww0dZpWcCy7x3akVEE2YibDfx8tPi3eize0u
zONiteKA; Path=/; expires=Sun, 01 Apr 2029 12:14:19 GMT; HttpOnly
```

(4) 用流水线方式实现前面 (1) (2), 即把它们请求拷贝到一起后发送出去 (可能太长, 第一部分可以只有看到末尾)。

http 请求:

```
GET /about/introduction HTTP/1.1
Host: sdcs.sysu.edu.cn
Connection: keep-alive

GET /sites/sdcs.sysu.edu.cn/files/574e48efa871f-300x199.png HTTP/1.1
Host: sdcs.sysu.edu.cn
Connection: keep-alive
```

http 响应:

```
HTTP/1.1 202 Accepted
Server: *****
Date: Thu, 04 Apr 2019 12:18:08 GMT
Content-Type: text/html; charset=utf-8
Transfer-Encoding: chunked
Connection: keep-alive
Pragma: no-cache
Cache-Control: no-cache
Expires: Thu, 04 Apr 2019 12:18:08 GMT
Set-Cookie:
FSSBB11UgzbN7N80T=4xYDbbzRXFEjryqt_rUWKgjftOT1b9USX2vb2oqwaR.jMnyolu_k50J7
N_7X0EoQqQXihtAiV3issX5YAzFq1R8ERk2.A1JmwCKW.JXyzw8sa89AFIbIRiGJRiWcOIB9qy
h.woZVBsFBob7HbP3MtZDTM3jy4pOeZVBdkkQkP12AJcG.7kPBWmiFyvuY0F07Nq; Path=/;
expires=Sun, 01 Apr 2029 12:18:08 GMT
Set-Cookie:
FSSBB11UgzbN7N80S=.szTkcmKkP.cbmSA2HX_5rrVPbbpH4HnI4SUCHIuyejjgtgAJUC.HoCo
LKESXyv5; Path=/; expires=Sun, 01 Apr 2029 12:18:08 GMT; HttpOnly
```

... (响应报文正文已省略)

```
HTTP/1.1 200 OK
Server: *****
Content-Type: image/png
Content-Length: 96721
Connection: keep-alive
Date: Thu, 04 Apr 2019 12:22:59 GMT
X-Content-Type-Options: nosniff
Last-Modified: Mon, 20 Feb 2017 13:08:34 GMT
ETag: "179d1-548f5f5f1b880"
Accept-Ranges: bytes
Set-Cookie:
FSSBB11UgzbN7N80T=4oNb1JAa9b8a7s9var4MxbQy1FzvK3o8w67jMkZso_o9F.Z_oGJGjH_2
29YoAEr2u2ApPyBSEzPJfBexzphHbKjjn.8_0hFvPASBc08aRfZ80Jif7kCKtZcKHSPopFprCLn
nkqbjGtKguJOpXfX2AwvlGRKESW6iPqfHs137m_7mJnGnVT4Q.B3rh4opbafiUrRa; Path=/;
expires=Sun, 01 Apr 2029 12:18:08 GMT
Set-Cookie:
FSSBB11UgzbN7N80S=Z71_TDRFRy6T7k8jzhkK33dgpwb4dCfb1Lw9tYzZJ9Ik__0pwYfjBuk1
xoCPkP4h; Path=/; expires=Sun, 01 Apr 2029 12:18:08 GMT; HttpOnly
```

... (响应报文正文已省略)

## 二、FTP 协议

看完 FTP 协议的课件后完成以下实验（测试服务器上的目录结构和文件见“参考资料”）：

FTP 服务器：IP 地址为 172.18.187.10，端口号为 21（用户名：abc，密码：123666）

### (1) 上传用学号命名的一个文本文件(学号.txt)

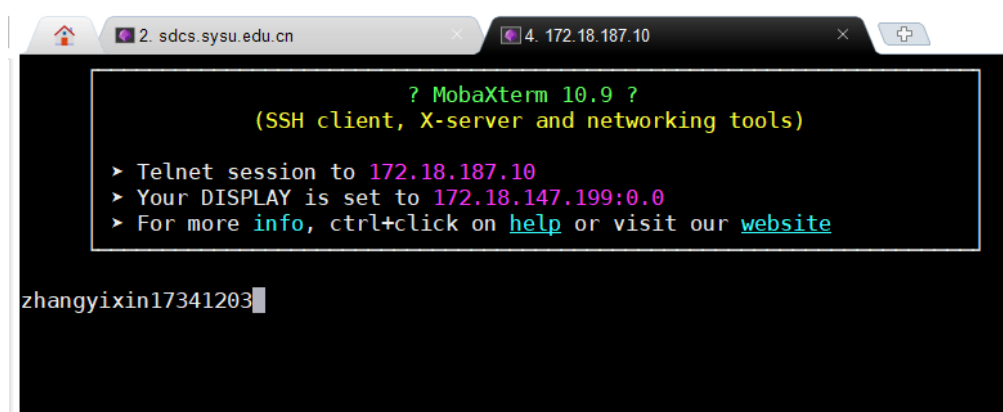
控制连接的请求响应信息：

如下，蓝色字体的是我在客户端输入的命令，黑色字体的是服务器返回来的信息。数据连接的端口号为  $197 \times 256 + 75 = 50507$ ；在控制连接执行 STOR 指令后，在数据连接终端窗口内输入一些字符串，然后回到控制连接窗口执行 ABOR 以终止数据传送。

```
220 Microsoft FTP Service
USER abc
331 Password required
PASS 123666
230 User logged in.
PASV
227 Entering Passive Mode (172,18,187,10,197,75).
STOR 17341203.TXT
125 Data connection already open; Transfer starting.
ABOR
425 Data channel was closed by ABOR command from client.
226 abor command successful.
QUIT
221 Goodbye.
```

数据连接的截屏：

下图中的字符串是我输入的：



### (2) 查看当前目录内容(太多则选择一些)，并标注出(1)中自己上传的文件

控制连接的请求响应信息：

本次数据连接的端口号为  $223 \times 256 + 240 = 57328$ 。

```
220 Microsoft FTP Service
USER abc
331 Password required
PASS 123666
230 User logged in.
PWD
257 "/" is current directory.
PASV
227 Entering Passive Mode (172,18,187,10,223,240).
LIST
125 Data connection already open; Transfer starting.
226 Transfer complete.
QUIT
221 Goodbye.
Connection closed by foreign host
```

数据连接的截屏：

```
? MobaXterm 10.9 ?
(SSH client, X-server and networking tools)

> Telnet session to 172.18.187.10
> Your DISPLAY is set to 172.18.147.199:0.0
> For more info, ctrl+click on help or visit our website

03-31-19 05:03PM 58 16307064.txt
03-31-19 06:06PM 3446 17341198.txt
03-31-19 11:30PM 44 17341203.TXT
03-31-19 09:32PM 51 17372048.txt
03-28-19 12:13PM <DIR> ebook
03-28-19 12:13PM <DIR> text

Session stopped
- Press <return> to exit tab
- Press R to restart session
- Press S to save terminal output to file
```

(3) 下载(1)中自己上传的文本文件

控制连接的请求响应信息:

所请求下载的文件是位于根目录 (/) 下的 17341203.txt。本次数据连接的端口号为  $197 \times 256 + 110 = 50542$ 。

```
220 Microsoft FTP Service
USER abc
331 Password required
PASS 123666
230 User logged in.
PASV
227 Entering Passive Mode (172,18,187,10,197,110).
RETR /17341203.TXT
125 Data connection already open; Transfer starting.
226 Transfer complete.
QUIT
221 Goodbye.
Connection closed by foreign host
```

数据连接的截屏:

```
? MobaXterm 10.9 ?
(SSH client, X-server and networking tools)

> Telnet session to 172.18.187.10
> Your DISPLAY is set to 172.18.147.199:0.0
> For more info, ctrl+click on help or visit our website

zhangyixin17341203

Session stopped
- Press <return> to exit tab
- Press R to restart session
- Press S to save terminal output to file
```

(4) 下载/ebook 下的一个二进制文件 (例如, .pdf 文件)

控制连接的请求响应信息:

由于要下载二进制文件, 因此需要提前设置 TYPE I。本次数据连接的端口号为  $198 \times 256 + 95 = 50783$ 。

```
220 Microsoft FTP Service
USER abc
```

```

331 Password required
PASS 123666
230 User logged in.
TYPE I
200 Type set to I.
PASV
227 Entering Passive Mode (172,18,187,10,198,95).
RETR /ebook/ftp.pdf
125 Data connection already open; Transfer starting.
226 Transfer complete.
QUIT
221 Goodbye.
Connection closed by foreign host

```

数据连接的截屏：

PDF 是二进制文件，因此在终端窗口显示的是乱码，要想下载原文件，需要使用 FTP 客户端，如内容五中我编写的 [FtpDownloader](#) 可以用来下载文件。

```

? MobaXterm 10.9 ?
(SSH client, X-server and networking tools)

> Telnet session to 172.18.187.10
> Your DISPLAY is set to 172.18.147.199:0.0
> For more info, ctrl+click on help or visit our website

%PDF-1.2
%庆?
4 0 obj
<</Length 5 0 R/Filter /FlateDecode>>
stream
x璩踟踟}蟬粲T? ? ?
第y7緋 烂Xj[43bfd枯#踐a*併5UH滴鶻亥 x ouLU綾?
QX[$ G^HN :4? 踟z
? };? ? 选|o窠A?
窠B緋j,tRn ?
一逮綾? 砑
qcm? 极肫謁g
翳 鵲\接g鸞纂 :經剝0銖III 踟耨鵲oz荒W緋7Q<L尽枕
窠H璽JQ+1t.? 浩@窠b
vS賜默? 5潘!賴t樊I踟j簾RI余[VHe/
?
濱鮑錄K? Ehj0鶻? 礪? ? ]LY 砑7i\hヤ? ? H{[ノ"}Si? 蜜n鉞Cq耨? 0頰耨Ly? x居0

```

(5) 采用断点续传下载一个/text 下的一个文本文件的一部分

控制连接的请求响应信息：

本次数据连接的端口号为  $197 \times 256 + 159 = 50591$ 。

```

220 Microsoft FTP Service
USER abc
331 Password required
PASS 123666
230 User logged in.
PASV
227 Entering Passive Mode (172,18,187,10,197,159).
REST 200
350 Restarting at 200.
RETR /text/Web.txt
125 Data connection already open; Transfer starting.
226 Transfer complete.
QUIT
221 Goodbye.
Connection closed by foreign host

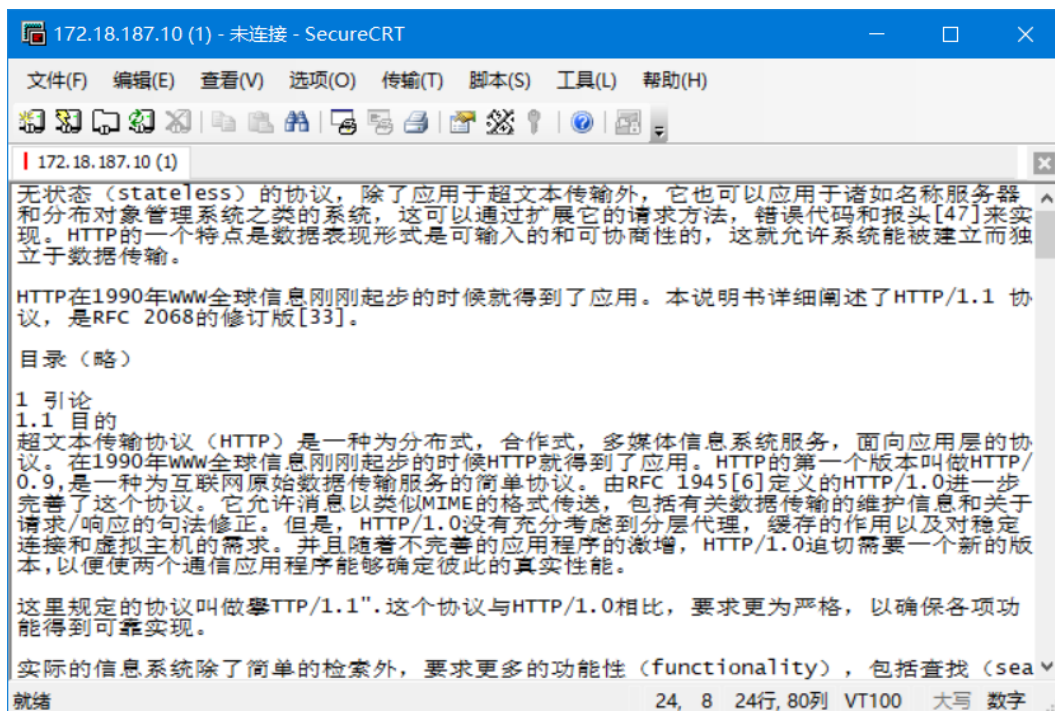
```

数据连接的截屏：



情况说明：由于我之前使用的 MobaXterm 中不支持 GB2312 字符集，因此无法解决中文乱码问题。因此下述截屏来自 SecureCRT 终端软件。

从数据连接的结果来看，返回的文档是并不是从文件头部开始的。如下图：



### 三、SMTP 协议

邮箱 zsusender3@sina.com 的用户名 **zsusender3**，密码：123456Aa

看完 SMTP 协议的课件后做以下实验：

(1) 通过 zsusender3@sina.com 发一封没有附件的邮件到你的邮箱。注意更改时间为当前时间。

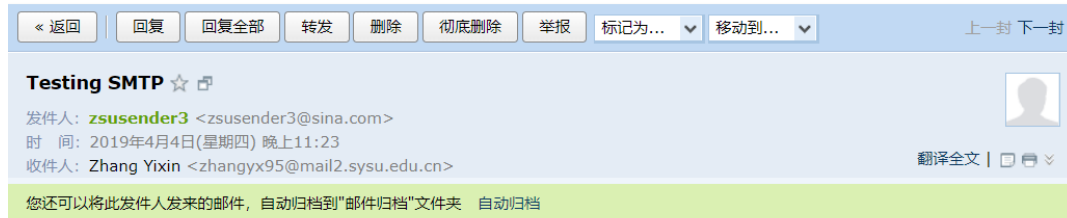
请求和响应信息：

```
220 smtp-5-123.smtpsmail.fmail.xd.sinanode.com ESMTP
HELO zsusender3
250 smtp-5-123.smtpsmail.fmail.xd.sinanode.com
AUTH LOGIN
334 VXN1cm5hbWU6
enN1c2VuZGVyMw==
334 UGFzc3dvcmQ6
MTIzNDU2QWE=
235 OK Authenticated
MAIL FROM:<zsusender3@sina.com>
250 ok
RCPT TO:<zhang*****@mail2.sysu.edu.cn>
250 ok
DATA
354 End data with <CR><LF>.<CR><LF>
Date:Thu, 04 Apr 2019 23:23:01 +0800
From: "zsusender3" <zsusender3@sina.com>
To: "Zhang Yixin"<zhang*****@mail2.sysu.edu.cn>
Subject: Testing SMTP

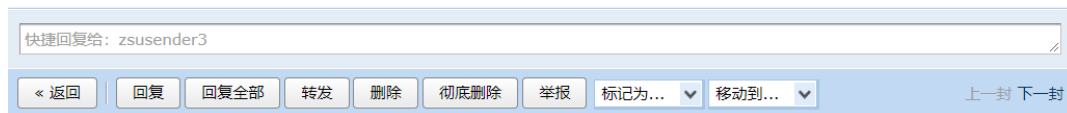
Hi, this is an email sent by telnet using SMTP.
Hope you enjoy it! Bye.
```

```
.
250 ok queue id 635396394003
QUIT
221 smtp-5-123.smtpsmail.fmail.xd.sinanode.com
Connection closed by foreign host
```

成功收到邮件:



Hi, this is an email sent by telnet using SMTP.  
Hope you enjoy it! Bye.



(2) 通过 zsusender3@sina.com 发一封带附件(二进制文件)的邮件(MIME.txt)到你的邮箱。参考课件“MIME.pdf”和观看 MIME 的视频。

请求和响应信息:

我将邮件内容头部的 From 字段修改成了我自己的名字,“张怡昕”的 Base64 编码结果为“1cXi+eq/”。下面蓝色字体表示我输入的,黑色字体是服务器响应的。

```
220 smtp-5-122.smtpsmail.fmail.xd.sinanode.com ESMTP
HELO jed
250 smtp-5-122.smtpsmail.fmail.xd.sinanode.com
AUTH LOGIN
334 VXN1cm5hbWU6
enN1c2VuZGVyMw==
334 UGFzc3dvcmQ6
MTIzNDU2QWE=
235 OK Authenticated
MAIL FROM:<zsusender3@sina.com>
250 ok
RCPT TO:<zhang*****@mail2.sysu.edu.cn>
250 ok
DATA
354 End data with <CR><LF>.<CR><LF>
Date: Fri, 5 Apr 2019 23:39:07 +0800
From: =?GB2312?B?1cXi+eq/?= <zsusender3@sina.com>
To: ZhangYixin<zhang*****@mail2.sysu.edu.cn>
Subject: rfc855
X-Priority: 3
X-Has-Attach: yes
X-Mailer: Foxmail 7, 1, 3, 52[cn]
Mime-Version: 1.0
Message-ID: <201404172030056405561@mail.sysu.edu.cn>
Content-Type: multipart/mixed;
        boundary="-----_001_NextPart061116676577_-----"

This is a multi-part message in MIME format.

-----_001_NextPart061116676577_-----
Content-Type: multipart/alternative;
```

```
boundary="-----_002_NextPart640832068374_-----"

-----=_002_NextPart640832068374_-----
Content-Type: text/plain;
    charset="GB2312"
Content-Transfer-Encoding: base64

dGVzdCENCg0KDQoNCg0K1cXTwMPx

-----=_002_NextPart640832068374_-----
Content-Type: text/html;
    charset="GB2312"
Content-Transfer-Encoding: quoted-printable

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML><HEAD>
<META content=3D"text/html; charset=3Dgb2312" http-equiv=3DContent-Type>
<STYLE>
BLOCKQUOTE {
    MARGIN-BOTTOM: 0px; MARGIN-LEFT: 2em; MARGIN-TOP: 0px
}
OL {
    MARGIN-BOTTOM: 0px; MARGIN-TOP: 0px
}
UL {
    MARGIN-BOTTOM: 0px; MARGIN-TOP: 0px
}
P {
    MARGIN-BOTTOM: 0px; MARGIN-TOP: 0px
}
BODY {
    FONT-SIZE: 10.5pt; FONT-FAMILY: Microsoft YaHei UI; COLOR: #000000; LINE-
HEIGHT: 1.5
}
</STYLE>

<META name=3DGENERATOR content=3D"MSHTML 10.00.9200.16863"></HEAD>
<BODY style=3D"MARGIN: 10px">
<DIV>test!</DIV>
<DIV>&nbsp;</DIV>
<HR style=3D"HEIGHT: 1px; WIDTH: 210px" align=3Dleft color=3D#b5c4df SIZE=
=3D1>

<DIV><SPAN>=D5=C5=D3=C0=C3=F1</SPAN></DIV></BODY></HTML>

-----=_002_NextPart640832068374_-----

-----=_001_NextPart061116676577_-----
Content-Type: application/octet-stream;
    name="rfc855.pdf"
Content-Transfer-Encoding: base64
Content-Disposition: attachment;
    filename="rfc855.pdf"

JVBERi0xLjIKJcfsj6IKNCAwIG9iag08PC9MZW5ndGggNSAwIFivRmlsdGVyIC9GbGF0ZURlY29k
... // 二进制附件的 Base64 编码，省略中间部分
awxlco8PCAvU2l6SAxNCAvUm9vdCAxIDAgUiAvSW5mbyAxMyAwIFIKPj4Kc3RhcnR4cmVmCjQw
MzkKJSVFT0YK

-----=_001_NextPart061116676577_-----
.
250 ok queue id 890795394010
QUIT
221 smtp-5-122.smtpsmail.fmail.xd.sinanode.com
```

Connection closed by foreign host

如下图，成功收到带二进制附件的邮件，并且经过我的验证，该 pdf 附件可以正常打开。



- (3) 通过 `zsusender3@sina.com` 发送另一个带附件的邮件给自己。可以先给你自己发一封带附件的邮件，再通过查看源码截取该响应报文的一部分，参见 `MIME.txt`。
- 用新浪邮箱发送一封邮件，然后下载 `eml` 格式的邮件（Web 端支持该功能），查看下载下来的 `eml` 格式的邮件，如下。注意附件 Base64 编码很长，这里将其省略以节约篇幅。



This is another testing email with a binary attachment. Look "ftp.pdf".



```
X-Mda-Received: from <webmail.sinamail.sina.com.cn>([<10.41.14.100>])
by <mda-14-116.mda.fmail.dbl.sinanode.com> with LMTP id <4704102>
Apr 07 2019 19:39:19 +0800 (CST)
```

```

X-Sina-MID:04A6600E15031F45000998BA40F64FC73A00000000000002
X-Sina-Attnum:1
Received: by webmail.sinamail.sina.com.cn (Postfix, from userid 495)
        id EEC672D0008F; Sun,  7 Apr 2019 19:39:18 +0800 (CST)
Date: Sun, 07 Apr 2019 19:39:18 +0800
Received: from zsusender3@sina.com([120.236.174.157]) by m0.mail.sina.com.cn via
HTTP;
        Sun, 07 Apr 2019 19:39:18 +0800 (CST)
Reply-To: zsusender3@sina.com
From: <zsusender3@sina.com>
To: "zsusender3" <zsusender3@sina.com>
Subject: Testing Mail 2
MIME-Version: 1.0
X-Priority: 3
X-MessageID: 5ca9e16624034ed9_201904
X-Originating-IP: [10.41.14.100]
X-Mailer: Sina WebMail 4.0
Content-Type: multipart/mixed;
        boundary="--sinamail_mix_f6ceb10bdd6c5411e662466178872b2b"
Message-Id: <20190407113918.EEC672D0008F@webmail.sinamail.sina.com.cn>

---sinamail_mix_f6ceb10bdd6c5411e662466178872b2b
Content-Type: multipart/alternative;
        boundary="--sinamail_alt_be7fa9665d224dfabd1d8b2b522cfbb3"

---sinamail_alt_be7fa9665d224dfabd1d8b2b522cfbb3
Content-Type: text/plain;
        charset=GBK
Content-Transfer-Encoding: base64
Content-Disposition: inline

VGhpcyBpcyBhbm90aGVyIHRlc3RpbmcgZW1haWwgd2l0aCBhIGJpbmFyeSBhdHRhY2htZW50LiBM
b29rICJmdHAucGRmIi4=

---sinamail_alt_be7fa9665d224dfabd1d8b2b522cfbb3
Content-Type: text/html; charset=GBK
Content-Transfer-Encoding: base64
Content-Disposition: inline

VGhpcyBpcyBhbm90aGVyIHRlc3RpbmcgZW1haWwgd2l0aCBhIGJpbmFyeSBhdHRhY2htZW50LiBM
b29rICJmdHAucGRmIi4=

---sinamail_alt_be7fa9665d224dfabd1d8b2b522cfbb3--

---sinamail_mix_f6ceb10bdd6c5411e662466178872b2b
Content-Type: application/pdf; name="?GBK?B?ZnRwLnBkZg==?="
Content-Disposition: attachment; filename="?GBK?B?ZnRwLnBkZg==?="
Content-Transfer-Encoding: base64

JVBERi0xLjIKJcfsj6IKNCAwIG9iago8PC9MZWN5dGggNSAwIFIvRmlsdGVyIC9GbGF0ZUR1Y29k
// 已省略
UiAvSW5mbyAyMTEgMBCSCj4+CnN0YXJ0eHJlZgo4MDcwNQo1JUVPRgo=

---sinamail_mix_f6ceb10bdd6c5411e662466178872b2b--

```

(4) (选做) 从你的邮箱发一份邮件到同学的邮箱。QQ 邮箱见【注意事项】。

请求和响应信息：

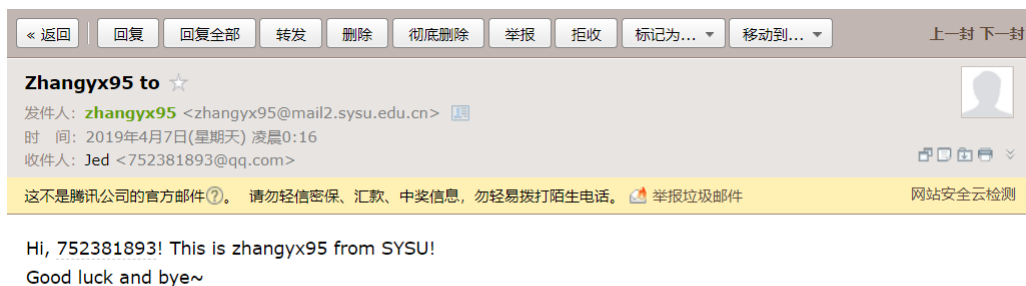
从 zhang\*\*\*\*\*@mail2.sysu.edu.cn 发送到 752381893@qq.com。注意发送方邮箱为中大邮箱，由腾讯企业邮箱管理，因此与 QQ 邮箱类似，连接到 SMTP 服务器时要使用 SSL，而且端口是 465 而不是 25。使

用的用户名为 zhang\*\*\*\*\*@mail2.sysu.edu.cn 的 Base64 编码，即“emhhbmd5eDk1QG1haWwyLnN5c3UuZWRR1mNu”。

```
220 smtp.qq.com Esmtp QQ Mail Server
HELO zhangyixin
250 smtp.qq.com
AUTH LOGIN
334 VXN1cm5hbWU6
emhhbmd5eDk1QG1haWwyLnN5c3UuZWRR1mNu
334 UGFzc3dvcmQ6
***** // 密码的 Base64 编码
235 Authentication successful
MAIL FROM:<zhang*****@mail2.sysu.edu.cn>
250 Ok
RCPT TO:<752381893@qq.com>
250 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
Date:Sun, 7 Apr 2019 00:16:01 +0800
From: "zhang*****" <zhang*****@mail2.sysu.edu.cn>
To: "752381893" <752381893@qq.com>
Subject: Zhang***** to

Hi, 752381893! This is zhang***** from SYSU!
Good luck and bye~
.
250 Ok: queued as
QUIT
221 Bye
```

如下图所示，接收方成功收到邮件：



#### 四、POP3 协议

邮箱 [zsurreceiver3@sina.com](mailto:zsurreceiver3@sina.com) 的用户名 zsurreceiver3，密码：123456Aa

看完 POP3 协议的课件后做以下实验：

(1) 查看 zsurreceiver3@sina.com 中每个邮件大小。

请求和响应信息：

使用 LIST 命令，可以看到邮箱中共有 12 封邮件，它们的大小也显示出来了。

```
+OK sina pop3 server ready
USER zsurreceiver3
+OK welcome to sina mail
PASS 123456Aa
+OK 12 messages (26967 octets)
LIST
+OK 12 messages (26967 octets)
1 939
2 938
```

```
3 8638
4 940
5 939
6 8829
7 959
8 983
9 916
10 958
11 957
12 971
.
QUIT
+OK sina mail see you next time
Connection closed by foreign host
```

(2) 取回 zsureceiver3@sina.com 最后一封邮件的邮件唯一标识符。

请求和响应信息:

从 (1) 可知最后一封邮件的编号是 12, 因此使用 “UIDL 12” 获取该邮件的唯一标识符。

```
+OK sina pop3 server ready
USER zsureceiver3
+OK welcome to sina mail
PASS 123456Aa
+OK 12 messages (26967 octets)
UIDL 12
+OK 12 044FB076B1509BA31B8B2CFBC1CE648AA100000000000001
QUIT
+OK sina mail see you next time
Connection closed by foreign host
```

(3) 取回 zsureceiver3@sina.com 最后一封邮件。

请求和响应信息:

从 (1) 可知最后一封邮件的编号是 12, 因此使用 “RETR 12” 获取该邮件。

```
+OK sina pop3 server ready
USER zsureceiver3
+OK welcome to sina mail
PASS 123456Aa
+OK 12 messages (26967 octets)
RETR 12
+OK 971 octets
X-Mda-Received: from <mx-14-107.mail.sina.com.cn>([<10.41.14.107>])
  by <mda-14-116.mda.fmail.dbl.sinanode.com> with LMTP id <109231>
  Apr 04 2019 18:00:28 +0800 (CST)
X-Sina-MID:044FB076B1509BA31B8B2CFBC1CE648AA100000000000001
X-Sina-Attnum:0
Received: from mail7-215.sinamail.sina.com.cn (HELO
mail7-215.sinamail.sina.com.cn)([202.108.7.215])
  by sina.com with SMTP
  id 5CA5D5BB000073C0; Thu, 4 Apr 2019 18:00:28 +0800 (CST)
X-Sender: zsusender3@sina.com
X-SMAIL-MID: 9623823856122
Received: from unknown (HELO zsusender3)([120.236.174.161])
  by sina.com with ESMTTP
  id 5CA5D57000004158; Thu, 4 Apr 2019 18:00:27 +0800 (CST)
X-Sender: zsusender3@sina.com
X-Auth-ID: zsusender3@sina.com
X-SMAIL-MID: 422433394474
Date:Tue, 25 Mar 2019 14:24:01 +0800
From: "zsusender3" <zsusender3@sina.com>
To: "zsureceiver3"<zsureceiver3@sina.com>
Subject: SDCS 17 isszym
```

```
This is a SDCS message in MIME format from 17341186.  
Hello! SDCS 17 from isszym
```

```
.  
QUIT  
+OK sina mail see you next time  
Connection closed by foreign host
```

(4) 取回三(1)中发到你邮箱的邮件。QQ 邮件见【注意事项】。

请求和响应信息:

我使用的邮箱是 zhang\*\*\*\*\*@mail2.sysu.edu.cn, 可以直接使用邮箱账号和密码连接至 POP3 服务器。

如下, 蓝色字体是我输入的命令, 黑色字体是服务器返回的信息, 红色字体是取回邮件的标准头部和正文内容。

```
+OK QQMail POP3 Server v1.0 Service Ready(QQMail v2.0)  
USER zhang*****@mail2.sysu.edu.cn  
+OK  
PASS ***** // 密码已隐藏  
+OK  
LIST  
+OK  
1 1159  
2 1251  
3 8961  
.  
RETR 2  
+OK  
Received: from mail7-214.sinamail.sina.com.cn (unknown [202.108.7.214])  
    by bizmx12.qq.com (NewMx) with SMTP id  
    for <zhang*****@mail2.sysu.edu.cn>; Thu, 04 Apr 2019 23:28:02 +0800  
X-QQ-FEAT: L3blbkxtNtkTxcxrFtftpY0z5K0ASODasasc8964a6Q8A11asf00VQFRUqjhFQR0RPO  
    fRNWlnqxzdbybjvMJMYaVWraIwsztJJVyVMORLF2IT+Wmnad126QozDV4/fcdC5CPrGJTNS  
    re5nNO96ByiEnSR3ZDaxcCawphDQM9BkmbRlhGlcZ8uKg80N5N+vODLLZ3cu1762KawS9gw  
    95pznU6BUisYSgyjcCVE1w2UHbrN+zjEsFJXsNQJhJ9PEyXygGwZNxJoT9dJjY9AwYovlc  
    a/ipmqv4vnQTeTgPMGqo9DhDjcdRorgU7aKA==  
X-QQ-MAILINFO: MJpDEouUnwTTpiyE/aFdfqrPUTjufMiDeZDFPS5ZLzr16ZrjBmzMI17hw  
    I87Drq7yrKQuTu615FC6AS57X6fx76asd7fcD2BBF6Rhlea28V13HD4Mdyas+dsBF4BQwC  
    zA5dU514x9AsyEzQV+cJfsM9vSSgICCPKQ==  
X-QQ-mid: bizmx12t1554391683tcyeds0wv  
X-QQ-ORGSender: zsusender3@sina.com  
Received: from unknown (HELO zsusender3)([120.236.174.157])  
    by sina.com with ESMTP  
    id 5CA6224000005912; Thu, 4 Apr 2019 23:28:01 +0800 (CST)  
X-Sender: zsusender3@sina.com  
X-Auth-ID: zsusender3@sina.com  
X-SMAIL-MID: 635396394003  
Date: Thu, 04 Apr 2019 23:23:01 +0800  
From: "zsusender3" <zsusender3@sina.com>  
To: "Zhang Yixin" <zhang*****@mail2.sysu.edu.cn>  
Subject: Testing SMTP
```

```
Hi, this is an email sent by telnet using SMTP.  
Hope you enjoy it! Bye.
```

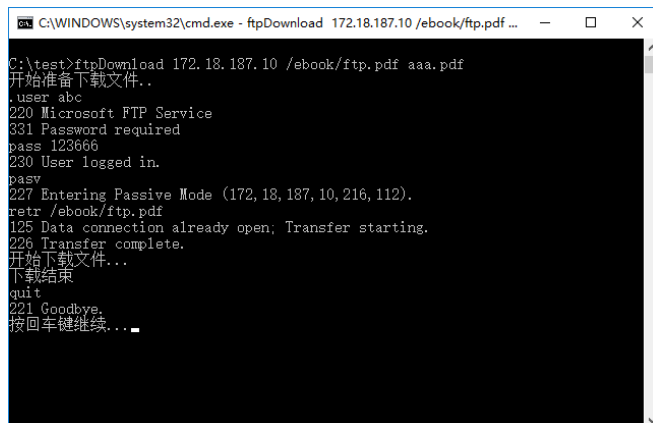
```
.  
QUIT
```



```
+OK Bye
Connection closed by foreign host
```

## 五、编写一个程序（聊天程序的客户端），用 FTP 协议下载指定文件（选做）

参考运行截屏：



参考字符串函数：

```
strchr();
strrchr();
sprintf(buf, "retr %s\r\n", filename);
sscanf(st, "%d,%d,%d,%d,%d,%d", &ip1, &ip2, &ip3, &ip4, &port1, &port2);
```

参考自编函数：

```
SOCKET dataConn() { ... } 建立数据连接
unsigned __stdcall myrecv(void *p) {...} 接收服务器的消息（线程函数）
void sendmsg(SOCKET sock, char *msg) { ... } 发送控制消息
int saveFile(SOCKET sock, char * fileName) { ... } 通过数据连接接收数据并保存为文件
```

参考程序：ftpDownload.exe

源代码：

本程序采用单线程（串行）方式编写，整个程序的流程是“发送命令——接受回应”不断重复。下面给出程序的源代码：

头文件 FtpDownloader.h

```
#include <stdio.h>
#include <WinSock2.h>
#include <stdlib.h>
#include <string.h>
#include <string>
#pragma comment(lib, "ws2_32.lib") // 使用winsock 2.0 Llibrary

const int BUFLen = 2000; // 缓冲区大小
const int FILEBUFLen = 10000; // 接受文件的缓冲区大小

int sendmsg(SOCKET sock, char* msg);
int recvmsg(SOCKET sock, char* buf);
SOCKET dataConn(char* received_msg);
int saveFile(SOCKET sock, char* fileName);
```

源文件 FtpDownloader.cpp

```
#include "FtpDownloader.h"

int main(int argc, char *argv[])
{
    printf("[FTP Downloader]\n");

    if (argc != 6) {
        printf("[-] 无效参数\n"
               "Command: FtpDownloader.exe ServerIPAddress User Password
```

```

FileName DestFile\n"
        "Example: FtpDownloader.exe 172.18.187.10 abc 123666
/ebook/ftp.pdf ftp1.pdf\n");
    exit(-1);
}

const char *host = argv[1]; // server IP to connect
u_short port = 21;          // server port to connect
struct sockaddr_in sin;      // an Internet endpoint address
char buf[BUFLen + 1];        // buffer for one line of text
SOCKET sock;                 // socket descriptor
int cc;                       // recv character count

WSADATA wsadata;
WSAStartup(MAKEWORD(2, 0), &wsadata); //加载winsock library

sock = socket(PF_INET, SOCK_STREAM, IPPROTO_TCP); //创建套接字, 参数: 因
特网协议簇(family), 流套接字, TCP协议
memset(&sin, 0, sizeof(sin)); // 从&sin开始的长度为
sizeof(sin)的内存清0
sin.sin_family = AF_INET; // 因特网地址簇
sin.sin_addr.s_addr = inet_addr(host); // 服务器IP地址(32位)
sin.sin_port = htons(port); // 服务器端口号
if (connect(sock, (struct sockaddr *)&sin, sizeof(sin)) == 0) { // 连接到
服务器
    printf("连接到 ftp://%s 成功! \n", argv[1]);
    printf("-----\n");
}
else {
    printf("[-] 失败: %d\n", GetLastError());
    printf("[-] 请检查输入的地址是否正确。按回车退出。 \n");
    exit(1);
}

/* 接收服务器欢迎信息 */
recvmsg(sock, buf);

/* 登录账户 */
sprintf(buf, "USER %s\r\n", argv[2]);
sendmsg(sock, buf);
recvmsg(sock, buf);
sprintf(buf, "PASS %s\r\n", argv[3]);
sendmsg(sock, buf);
recvmsg(sock, buf);

/* 被动连接 */
sprintf(buf, "PASV\r\n");
sendmsg(sock, buf);
recvmsg(sock, buf);
SOCKET data_sock = dataConn(buf);

/* 下载文件 */
sprintf(buf, "RETR %s\r\n", argv[4]);
sendmsg(sock, buf);
recvmsg(sock, buf);
int filesize = saveFile(data_sock, argv[5]);

/* 关闭连接 */
sprintf(buf, "QUIT\r\n");
sendmsg(sock, buf);
recvmsg(sock, buf);

closesocket(sock); // 关闭监听套接字
WSACleanup();      // 卸载winsock library

printf("-----\n");
printf("下载结束: %s, 共传输 %d 字节。 \n", argv[5], filesize);

return 0;
}

```

```

int sendmsg(SOCKET sock, char* msg) {
    printf(">>> ");
    printf(msg); //显示消息
    return send(sock, msg, strlen(msg) + 1, 0); // 发送消息
}

int recvmsg(SOCKET sock, char* buf) {
    int result = recv(sock, buf, BUFLen, 0);
    buf[result] = '\0';
    printf("[+] ");
    printf(buf);
    return result;
}

SOCKET dataConn(char* received_msg) {
    SOCKET result = socket(PF_INET, SOCK_STREAM, IPPROTO_TCP);
    int ip1, ip2, ip3, ip4, port1, port2;
    char data_ipaddr[17];
    int data_port;
    char* pos = strchr(received_msg, '(');
    sscanf(pos + 1, "%d,%d,%d,%d,%d,%d", &ip1, &ip2, &ip3, &ip4, &port1, &port2);
    sprintf(data_ipaddr, "%d.%d.%d.%d", ip1, ip2, ip3, ip4);
    data_port = port1 * 256 + port2;

    struct sockaddr_in data_sin; // an Internet endpoint address
    memset(&data_sin, 0, sizeof(data_sin));
    data_sin.sin_family = AF_INET; // 因特网地址簇
    data_sin.sin_addr.s_addr = inet_addr(data_ipaddr); // 服务
器IP地址(32位)
    data_sin.sin_port = htons(data_port); // 服务
器端口号

    connect(result, (struct sockaddr *)&data_sin, sizeof(data_sin));
    return result;
}

int saveFile(SOCKET sock, char* fileName) {
    int size = 0; // 下载的文件大小
    char filebuf[FILEBUFLen+1];
    FILE* destfile = fopen(fileName, "wb");
    int recvlen = 0; // 每次接收到的数据量
    while ((recvlen = recv(sock, filebuf, FILEBUFLen, 0)) > 0) { // 循环接收
数据, 直到文件传输完毕
        size += recvlen;
        fwrite(filebuf, 1, FILEBUFLen, destfile);
    }
    fclose(destfile);
    return size; // 返回下载的文件大小
}

```

运行截屏:

注意, 该程序必须在控制台窗口运行, 而不能直接通过双击的方式运行, 否则程序会一闪而过, 用户无法看到提示信息。正确的参数格式为:

FtpDownloader.exe ServerIPAddress User Password FileName DestFile

以下是各种不同情况的运行截图:

①输入的参数个数不正确:

```
Cmder

E:\JedData\OneDrive\SYSU_Lessons\计算机网络\计网实验\4.应用层实验\FtpDownloader\x64\Debug
# FtpDownloader.exe 172.18.187.10
[FTP Downloader]
[-] 无效参数
Command: FtpDownloader.exe ServerIPAddress User Password FileName DestFile
Example: FtpDownloader.exe 172.18.187.10 abc 123666 /ebook/ftp.pdf ftp1.pdf

E:\JedData\OneDrive\SYSU_Lessons\计算机网络\计网实验\4.应用层实验\FtpDownloader\x64\Debug
# |
```

②给定的 FTP 服务器无法连接（如：错误的 IP 地址）：

```
Cmder

E:\JedData\OneDrive\SYSU_Lessons\计算机网络\计网实验\4.应用层实验\FtpDownloader\x64\Debug
# FtpDownloader.exe 127.0.0.1 abc 123666 /ebook/ftp.pdf ftp1.pdf
[FTP Downloader]
[-] 失败: 10061
[-] 请检查输入的地址是否正确。按回车退出。

E:\JedData\OneDrive\SYSU_Lessons\计算机网络\计网实验\4.应用层实验\FtpDownloader\x64\Debug
#
```

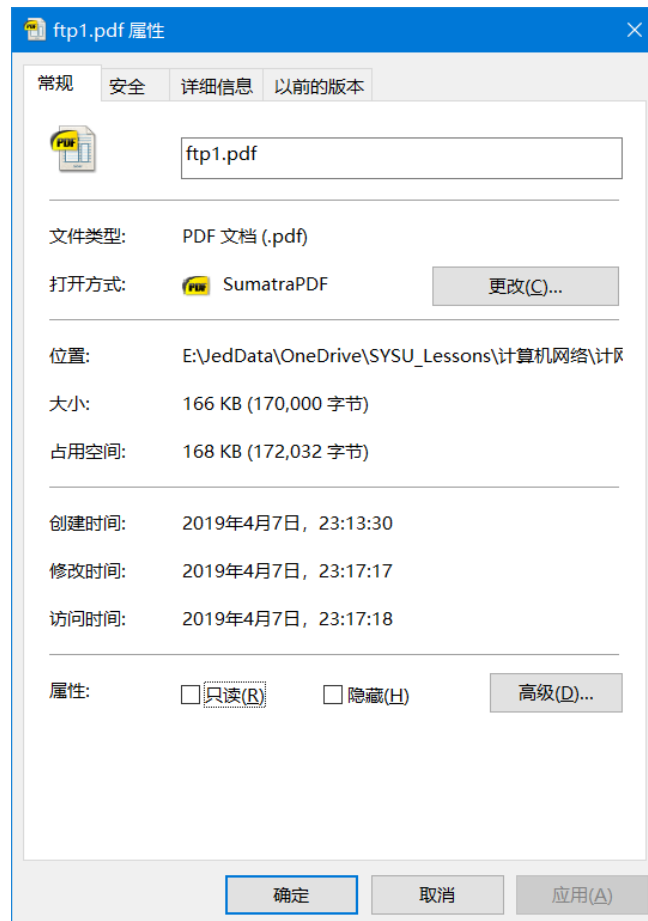
③正确地运行并成功下载文件：

```
Cmder

E:\JedData\OneDrive\SYSU_Lessons\计算机网络\计网实验\4.应用层实验\FtpDownloader\x64\Debug
# FtpDownloader.exe 172.18.187.10 abc 123666 /ebook/ftp.pdf ftp1.pdf
[FTP Downloader]
连接到 ftp://172.18.187.10 成功!
-----
[+] 220 Microsoft FTP Service
>>> USER abc
[+] 331 Password required
>>> PASS 123666
[+] 230 User logged in.
>>> PASV
[+] 227 Entering Passive Mode (172,18,187,10,195,53).
>>> RETR /ebook/ftp.pdf
[+] 125 Data connection already open; Transfer starting.
>>> QUIT
[+] 226 Transfer complete.
-----
下载完成: ftp1.pdf, 共传输 85028 字节。

E:\JedData\OneDrive\SYSU_Lessons\计算机网络\计网实验\4.应用层实验\FtpDownloader\x64\Debug
#
```

经验证，上面下载的 ftp1.pdf 文件下载成功，并且可以正常地打开阅读。下面是成功下载文件的属性截图：



六、采用聊天程序的客户端。该客户端采用两个线程实现：一个输入和发送线程，一个接收线程。（选做）  
源代码：

头文件 TcpClient.h

```
#include <stdio.h>
#include <stdlib.h>
#include <WinSock2.h>
#include <string.h>
#include <process.h>
#include <string>
#pragma comment(lib, "ws2_32.lib") // 使用winsock 2.0 Library

const int BUFLen = 1000; // 发送缓冲区大小
const int RBUFLen = 10; // 接收缓冲区大小

unsigned __stdcall recvThread(void *p);
```

源文件 TcpClient.cpp

```
#include "TcpClient.h"

int main(int argc, char *argv[])
{
    if (argc != 3) {
        printf("[+] Invalid arguments\n");
        printf("Usage: TcpClient.exe IPAddress PortNumber\n");
        exit(-1);
    }

    const char* host = argv[1]; // server IP to connect
    u_short port = atoi(argv[2]); // server port to connect
    struct sockaddr_in sin; // an Internet endpoint address
```

```

char buf[BUFLen + 1];           // buffer for one line of text
SOCKET sock;                   // socket descriptor

WSADATA wsadata;
WSAStartup(MAKEWORD(2, 0), &wsadata); //加载winsock library

sock = socket(PF_INET, SOCK_STREAM, IPPROTO_TCP); //创建套接字, 参数: 因特网协
议簇(family), 流套接字, TCP协议
memset(&sin, 0, sizeof(sin)); // 从&sin开始的长度为
sizeof(sin)的内存清0
sin.sin_family = AF_INET; // 因特网地址簇
sin.sin_addr.s_addr = inet_addr(host); // 服务器IP地址(32位)
sin.sin_port = htons(port); // 服务器端口号
if (connect(sock, (struct sockaddr *)&sin, sizeof(sin)) != 0) { // 连接到服务器
    printf("[+] 失败: %ld\n", GetLastError());
    printf("[+] 请检查输入的地址是否正确。按回车退出。 \n");
    exit(1);
}

HANDLE recv_thread = (HANDLE)_beginthreadex(NULL, 0, &recvThread, (void *)&sock, 0,
NULL);
while (1)
{
    fgets(buf, BUFLen, stdin);
    size_t cmdlen = strlen(buf) - 1;
    buf[cmdlen] = '\r';
    buf[cmdlen+1] = '\n'; // 命令以"\r\n"结尾
    int sendlen = send(sock, buf, cmdlen + 2, 0); // 发送命令
}

CloseHandle(recv_thread);
closesocket(sock); // 关闭监听套接字
WSACleanup(); // 卸载winsock library

return 0;
}

unsigned __stdcall recvThread(void *p)
{
    char rbuf[RBUFLen + 1]; // 接收缓冲区

    SOCKET sock = *(SOCKET *)p;
    while (1)
    {
        int recvlen;
        while ((recvlen = recv(sock, rbuf, RBUFLen, 0)) > 0) {
            rbuf[recvlen] = '\0';
            printf("%s", rbuf);
        }

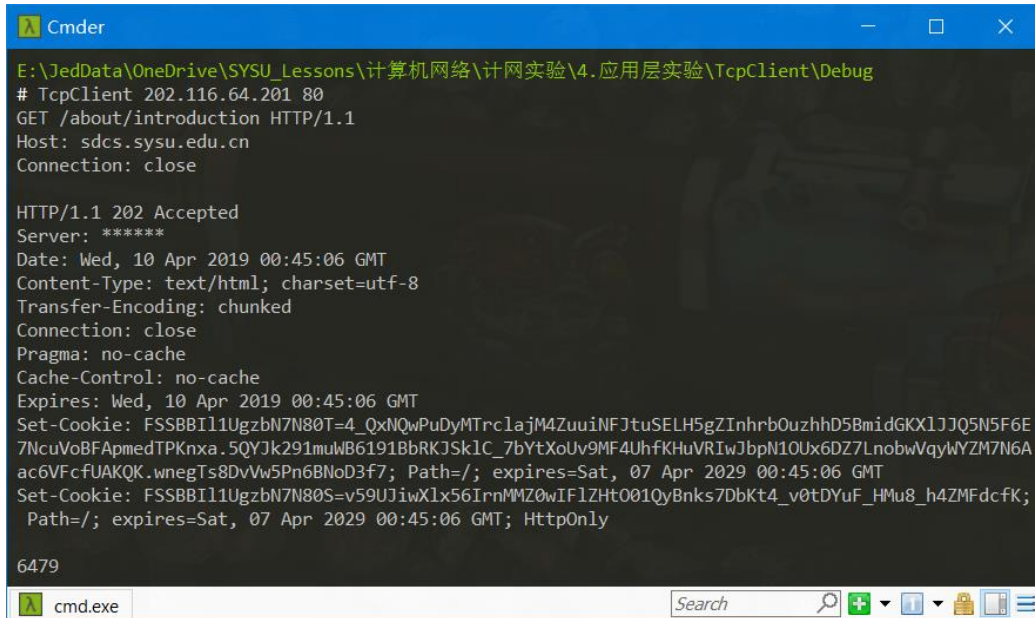
        if (recvlen == SOCKET_ERROR)
        {
            printf("[+] Error: %ld.\n", GetLastError());
            exit(-1);
        }
        else if (recvlen == 0)
        {
            printf("[+] Connection closed!\n");
            closesocket(sock); // 关闭监听套接字
            WSACleanup(); // 卸载winsock library
            exit(-1); // 退出程序
            break;
        }
    }

    return 0;
}

```

使用该 TcpClient 重复尝试步骤一到步骤四, 均可成功, 且结果与以上相同, 不再赘述。唯一不同的是, 使用我编写的 TcpClient 前需要提供 IP 地址, 即需要手动利用 DNS 服务将域名解析成 IP 地址后

再建立 TCP 连接。示例截屏如下：



```
E:\JedData\OneDrive\YSYSU_Lessons\计算机网络\计网实验\4.应用层实验\TcpClient\Debug
# TcpClient 202.116.64.201 80
GET /about/introduction HTTP/1.1
Host: sdc.sysu.edu.cn
Connection: close

HTTP/1.1 202 Accepted
Server: *****
Date: Wed, 10 Apr 2019 00:45:06 GMT
Content-Type: text/html; charset=utf-8
Transfer-Encoding: chunked
Connection: close
Pragma: no-cache
Cache-Control: no-cache
Expires: Wed, 10 Apr 2019 00:45:06 GMT
Set-Cookie: FSSBB1l1UgzbN7N80T=4_QxNQwPuDyMTrclajM4ZuuiNFJtuSELH5gZInhrbOuzhhD5BmidGKXlJJQ5N5F6E
7NcuVoBFAPmedTPKnxa.5QYJk291mulWB6191BbRKJsklC_7bYtXoUv9MF4UhfKHuVRIwJbpN10Ux6DZ7LnobwVqyWYZM7N6A
ac6VFcfUAKQK.wnegTs8DvVw5Pn6BNoD3f7; Path=/; expires=Sat, 07 Apr 2029 00:45:06 GMT
Set-Cookie: FSSBB1l1UgzbN7N80S=v59UJiwXlX56IrnMMZ0wIFlZht001QyBnks7DbKt4_v0tDYuF_HMu8_h4ZMFdcfK;
Path=/; expires=Sat, 07 Apr 2029 00:45:06 GMT; HttpOnly

6479
```

#### 【完成情况】

是否采用了老师的 TcpClient.exe 执行命令(√ ×): [×]

是否完成以下步骤?(√完成 ×未做)

(1) [√] (2) [√] (3) [√] (4) [√] (5) [√] (6) [√]

#### 【实验体会】

1. **HTTP 请求头之后有一个空行。**HTTP 请求报文分为请求行、头部行和消息正文，其中头部行与消息正文之间有一个空行(\\r\\n)。在本次实验发送的 HTTP 请求中，消息正文部分均为空，因此在使用 Telnet 发送 HTTP 请求时，需要在头部行后面紧跟两个空行，否则请求不完整，则不会得到相应。如果空行方式不当，则可能导致服务器返回 400 错误。
2. **HTTP 的 Connection 值一般是 keep-alive 的。**使用 Telnet 发送带有“Connection: close”的请求，得到服务器响应后该连接立即被关闭；若发送带有“Connection: keep-alive”的请求，则服务器响应后连接没有关闭，可以紧接着直接发送下一条请求。如果不发送该字段，我发现效果与“Connection: keep-alive”是相同的。查阅资料得知，这是因为服务器默认的 Connection 设置为 keep-alive 的。
3. **FTP 协议的控制连接和数据连接。**FTP 协议的显著特征是具有控制连接和数据连接——前者负责发送命令，后者负责上传或下载数据；前者一般使用 21 号端口（也可以修改），而后的端口号是不确定的，使用数据连接之前要先通过控制连接协商一个地址和端口号。使用 Telnet 的过程中发现，数据连接只能进行“一次服务”，如上传一个文件、下载一个文件或者显示一次文件目录，一旦完成了这些动作，数据连接立即被关闭。若想进行另一个操作，需要建立新的数据连接。新的数据连接和之前的数据连接的地址与端口号不一定相同，需要重新协商。
4. **SMTP 协议用到的字符全部是可显示 ASCII 码，也就是 7 位 ASCII 码。**如果要传输特殊字符或者二进制文件，则需要使用特殊的编码方式将它们编码为 7 位 ASCII 码。用到的编码方式主要是 Base64 和 quoted-printable 这两种。经查阅资料得知，早期的电子邮件功能简单，仅仅用来传输可见的字符，因此只支持 7 位 ASCII。不过随着互联网和计算机的发展，媒体种类越来越多，才采用 MIME 来支持非 ASCII 数据的传输。
5. 本次实验内容的五和六都通过上次实验(Chat 实验)的程序改写而来。对于 FtpDownloader 程序，我使用了单线程(串行)的方法——采用这种方法是由 FTP 协议的特点决定的——仅有发送命令、接受回复这两个功能。对于 TcpClient 程序，采用了与聊天程序相似的多线程写法——主线程负责发送，子线程负责接收。在编写这两个程序的过程中需要特别注意的是，**客户端向服务器发送**

的命令都必须以“\r\n”结尾。

6. **关于我使用的终端软件。**平日里我使用 SSH 时最常用的终端软件是 MobaXterm，它具有许多功能，其中就包含 Telnet 的支持，因此我选用该软件作为完成本次实验的主要软件。除此之外，我还使用 cmd 中的 Telnet、SecureCRT、老师提供的 TCPClient.exe、我自己编写的 TcpClient 一共 5 种不同的客户端来测试实验中的内容一到四，结果基本相同。