实验二 Soskct 编程实现网络通信
自 45 柳荫 2014011858

实验目的 :掌握 Socket 编程思想,并实现简单的 Socket 应用的连接通信过程。

实验内容：

1, Simple SMTP

在不使用身份验证的情况下，发件者的邮箱不起实质作用，目的地址应该是真实的地址，本实验中我将邮件通过 gmail 邮箱“代发“到自己的清华邮箱中(liuyin14@mails.tsinghua.edu.cn)。

```
/** 与邮件服务器建立TCP连接. */
// TODO: 1.在""中填入我们的smtp服务器和正确端口，助教老师的服务器地址166.111.74.90，端口是25
// e.g. Socket socket = new Socket("mails.163.com",25);
Socket socket = new Socket("mails.tsinghua.edu.cn",25);
```

```
// TODO: 2.把code改为合适的代码
int code = 220; //把-1改为合适的代码
if (!response.startsWith(Integer.toString(code))) {
    throw new Exception(code + " reply not received from server.");
}
```

```
// TODO: 3.填入命令
command = "HELO smtp\r\n";
System.out.print(command);
os.write(command.getBytes("US-ASCII"));
response = br.readLine();
System.out.println(response);
// TODO: 4.把code改为合适的代码
code = 250; //把-1改为合适的代码
if (!response.startsWith(Integer.toString(code))) {
    throw new Exception(code + " reply not received from server.");
}

/** 发送 MAIL FROM 命令. */
// TODO: 5.填入命令
command = "MAIL FROM: <yinliuchr@gmail.com>\r\n";
System.out.print(command);
os.write(command.getBytes("US-ASCII"));
response = br.readLine();
System.out.println(response);
// TODO: 6.把code改为合适的代码
code = 250; //把-1改为合适的代码
if (!response.startsWith(Integer.toString(code))) {
    throw new Exception(code + " reply not received from server.");
}
```

```java
/** 发送 RCPT TO 命令. */
// TODO: 7.填入命令
command = "RCPT TO: <liuyin14@mails.tsinghua.edu.cn>\r\n";
System.out.print(command);
os.write(command.getBytes("US-ASCII"));
response = br.readLine();
System.out.println(response);
// TODO: 8.把code改为合适的代码
code = 250; //把-1改为合适的代码
if (!response.startsWith(Integer.toString(code))) {
    throw new Exception(code + " reply not received from server.");
}

/** 发送 DATA 命令. */
// TODO: 9.填入命令
command = "DATA\r\n";
System.out.print(command);
os.write(command.getBytes("US-ASCII"));
response = br.readLine();
System.out.println(response);
// TODO: 10.把code改为合适的代码
code = 354; //把-1改为合适的代码
if (!response.startsWith(Integer.toString(code))) {
    throw new Exception(code + " reply not received from server.");
}
```

```java
/** 自动写入当前的日期 */
String date = "DATE " + dFormat.format(dDate) + "\r\n";
System.out.print(date);
os.write(date.getBytes("US-ASCII"));
String str = "";
// TODO: 11.把"x@x.x"改为邮件中显示的发件人地址
str = "From:" + "academic@DA.tsinghua.edu.cn" + "\r\n";
System.out.print(str);
os.write(str.getBytes("US-ASCII"));
// TODO: 12.把"x@x.x"改为邮件中显示的收件人地址
str = "To:" + "liuyin14@mails.tsinghua.edu.cn" + "\r\n";
System.out.print(str);
os.write(str.getBytes("US-ASCII"));


/** 发送邮件内容. */
// TODO: 13.在"x"中填入Subject内容.
str = "SUBJECT:" + "Scholarship" + "\r\n\r\n";
System.out.print(str);
os.write(str.getBytes("UTF-8"));

// TODO: 14.在"x"中填入邮件正文内容.
str = "柳荫同学: " + "\r\n" +"    恭喜你获得了清华大学自动化系国家奖学金！" + "\r\n" + "    请本周四下午2.00去主楼409签字！"
        + "\r\n" + "\r\n" + "自动化系教务处" + "\r\n";
System.out.print(str);
os.write(str.getBytes("UTF-8"));
```

```java
/** 以.作为邮件内容的结束符 */
str = ".\r\n";
System.out.print(str);
os.write(str.getBytes("US-ASCII"));
response = br.readLine();
System.out.println(response);
// TODO: 15.把code改为合适的代码
code = 250; //把-1改为合适的代码
if (!response.startsWith(Integer.toString(code))) {
    throw new Exception(code + " reply not received from server.");
}

/** 发送 QUIT 命令. */
//TODO: 16.填入命令
command = "QUIT\r\n";
System.out.print(command);
os.write(command.getBytes("US-ASCII"));
response = br.readLine();
System.out.println(response);
```

实验结果：

```
220 tsinghua.edu.cn Anti-spam GT for Coremail System (tsinghua[20160516])
HELO smtp
250 OK
MAIL FROM: <yinliuchr@gmail.com>
250 Mail OK
RCPT TO: <liuyin14@mails.tsinghua.edu.cn>
250 Mail OK
DATA
354 End data with <CR><LF>.<CR><LF>
DATE: Wednesday, October 19, 2016 11:36:32 PM CST
From:academic@DA.tsinghua.edu.cn
To:liuyin14@mails.tsinghua.edu.cn
SUBJECT:Scholarship

柳荫同学：
    恭喜你获得了清华大学自动化系国家奖学金！
    请本周四下午2.00去主楼409签字！

自动化系教务处
.
250 Mail OK queued as D8xvpgAX+an8kgdYppwkAA--.10576S2
QUIT
221 Bye

Process finished with exit code 0
```

Scholarship

发件人：academic@DA.tsinghua.edu.cn
(由 yinliuchr@gmail.com 代发)
时　间：2016年10月19日 23:43:11 (星期三)
收件人：liuyin14@mails.tsinghua.edu.cn

柳荫同学：
恭喜你获得了清华大学自动化系国家奖学金！
请本周四下午2.00去主楼409签字！

自动化系教务处

2 SMTP with authentification

仅需在原有代码上添加身份验证的 3 段密码即可：

```java
/** 发送 AUTH LOGIN 命令. */
command = "AUTH LOGIN\r\n";
System.out.print(command);
os.write(command.getBytes("US-ASCII"));
response = br.readLine();
System.out.println(response);
code = 334;
if (!response.startsWith(Integer.toString(code))) {
    throw new Exception(code + " reply not received from server.");
}
```

```java
/** 登陆账号 */
command = "liuyin14@mails.tsinghua.edu.cn";
System.out.print(command + "\r\n");
command = encoder.encode(command.getBytes()) + "\r\n";      //编码再发送

System.out.println(command);

os.write(command.getBytes("US-ASCII"));
response = br.readLine();
System.out.println(response);
code = 334;
if (!response.startsWith(Integer.toString(code))) {
    throw new Exception(code + " reply not received from server.");
}
```

```java
/** 登陆密码 */
command = "[redacted]";
System.out.print(command + "\r\n");
command = encoder.encode(command.getBytes()) + "\r\n";      //编码再发送

System.out.println(command);

os.write(command.getBytes("US-ASCII"));
response = br.readLine();
System.out.println(response);
code = 235;
if (!response.startsWith(Integer.toString(code))) {
    throw new Exception(code + " reply not received from server.");
}
```

此外就是思考题里的将发送信件的内容改成如下：

```java
// TODO: 13.在"x"中填入Subject内容.
str = "SUBJECT:" + "Simple SMTP" + "\r\n\r\n";
System.out.print(str);
os.write(str.getBytes("US-ASCII"));
// TODO: 14.在"x"中填入邮件正文内容.
str = "Hi TA" + "\r\n";
System.out.print(str);
os.write(str.getBytes("US-ASCII"));
str = "I'm very glad to inform you that I successfully complete the simple SMTP with authentification." + "\r\n";
System.out.print(str);
os.write(str.getBytes("US-ASCII"));
str = "I am LiuYin. My studentID is 2014011858." + "\r\n";
System.out.print(str);
os.write(str.getBytes("US-ASCII"));
```

实验结果：

```
"C:\Program Files\Java\jdk1.8.0_102\bin\java" ...
Files\Java\jdk1.8.0_102\bin\java" -Didea.launcher.port=7547 "-Didea.launcher.bin.path=C:\0_soft\Intelij\Intel
HELO yinliuchr
250 OK
AUTH LOGIN
334 dXNlcm5hbWU6
liuyin14@mails.tsinghua.edu.cn
bGl1eW1uMTRAbWFpbHMudHNpbmdodWEuZWR1LmNu

334 UGFzc3dvcmQ6
Gofrom=ato8
R29mcm9tPWFObzg=

235 Authentication successful
MAIL FROM:<liuyin14@mails.tsinghua.edu.cn>
250 Mail OK
RCPT TO:<yinliuchr@163.com>
250 Mail OK
DATA
354 End data with <CR><LF>.<CR><LF>
DATE Wednesday, October 19, 2016 11:50:43 PM CST
From:academic@stanford.edu.cn
To:yinliuchr@163.com
SUBJECT:Simple SMTP

Hi TA
I'm very glad to inform you that I successfully complete the simple SMTP with authentification.
I am LiuYin. My studentID is 2014011858.
.
250 Mail OK queued as DMxvpgCXne1Q1gdYOGs1AA--.9094S2
QUIT
221 Bye

Process finished with exit code 0
```

Simple SMTP

发件人：academic<academic@stanford.edu.cn> (由 liuyin14@mails.tsinghua.edu.cn 代发，帮助)

收件人：我<yinliuchr@163.com>

时　间：2016年10月19日 23:50 (星期三)

翻译成中文

工作需知：职场人的6大自我修养！ 查看详情>>

Hi TA

I'm very glad to inform you that I successfully complete the simple SMTP with authentification.

I am LiuYin. My studentID is 2014011858.

3 UDP Ping 实验

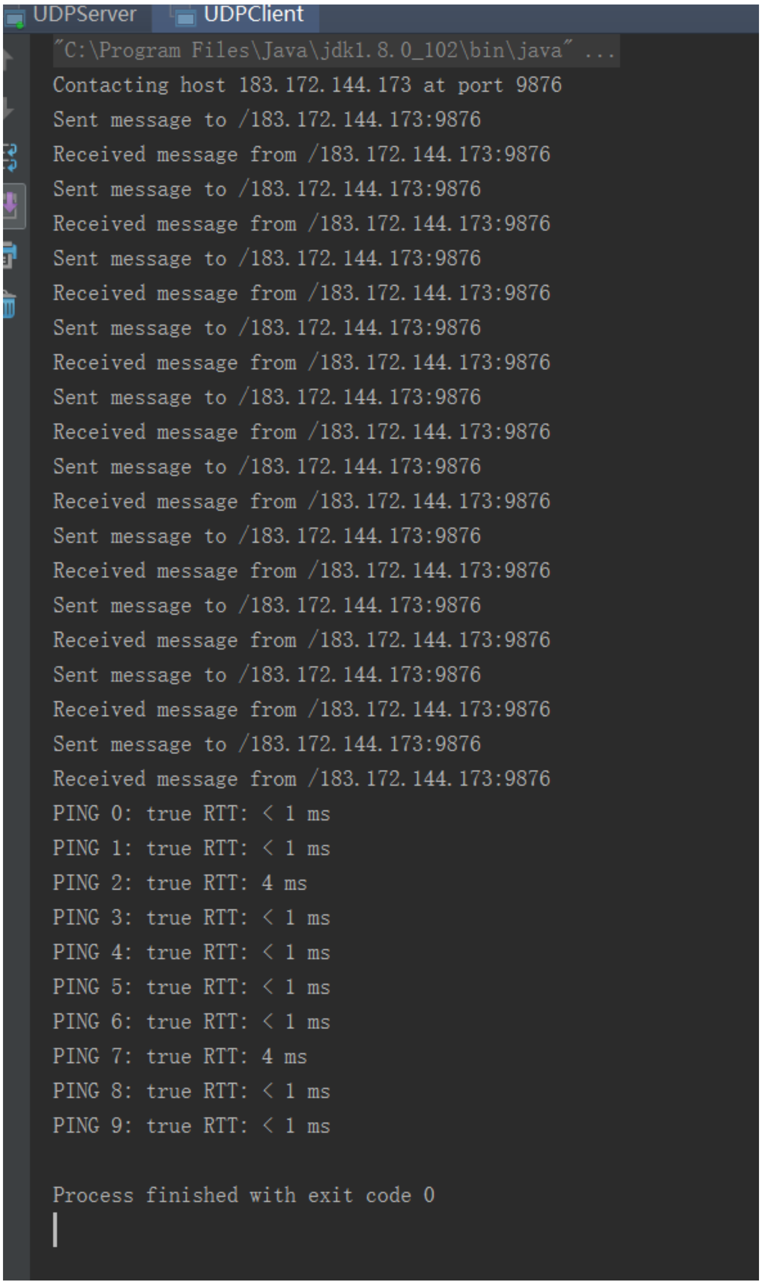
```
/* Send ping to recipient */
try {
    //TODO: 1.PingMessage对象包含了什么信息?
    //TODO: 1.PingMessage对象包含了server的地址、端口号及'message'信息（上面81行有显示）
    ping = new PingMessage(InetAddress.getByName(remoteHost),
            remotePort, message);
```

```
try {
    //TODO: 2.这里取得的PingMessage reply对象与上面发送的ping对象内容是否一样?
    //TODO: 2.这里取得的PingMessage reply对象与上面发送的ping对象内容一样，因为从UDPServer类中可以看到它又把receivepacket发了回来
    PingMessage reply = receivePing();
    //TODO: 3.handleReply的作用?是以致它所改变的变量值是什么?
    //TODO: 3.handleReply的作用是更新UDPClient的变量内容 是以致它所改变的变量值是是2个全局变量replies和rtt（从本程序结尾的该函数可看出）
    handleReply(reply.getContents());
```

```
try {
    socket.setSoTimeout(REPLY_TIMEOUT);
} catch (SocketException e) {
    System.out.println("Error setting timeout REPLY_TIMEOUT: " + e);
}
//TODO: 4.这个while循环的作用是什么,与第96行的receivePing调用有什么关系,既然上面都已调用了receivePing()，为何这里要重新调用??
//TODO: 4.这个while循环的作用是处理超时的 reply,第96行的receivePing调用是处理在1000ms内到达的,
//          而这里的是为了处理在最后5s内到达的，从上面的118行的try语句可以看出
while (numReplies < NUM_PINGS) {
    try {
```

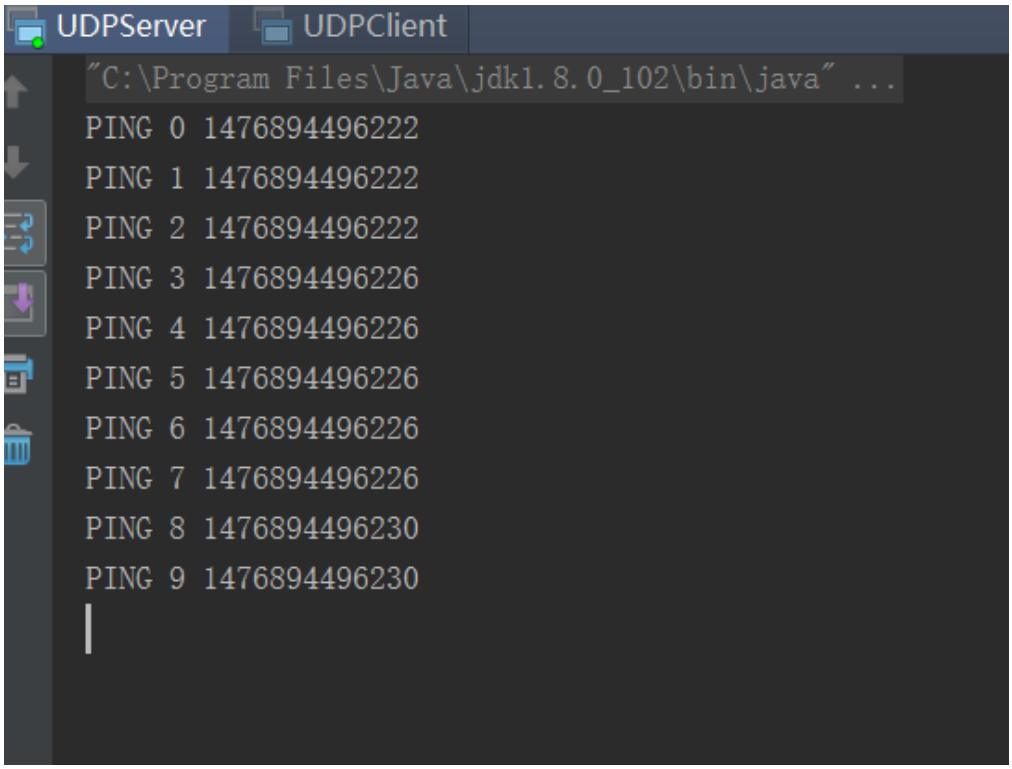
```
/* Calculate RTT and store it in the rtt-array. */
Date now = new Date();
//TODO: 5. 请简要说明这里的rtt的计算过程.
//TODO: 5. 请简要说明这里的rtt的计算过程.then表示的是reply字符串中的时间，即发送时间，now读出当前时间，即接受时间，两者相减即为RTT
rtt[pingNumber] = now.getTime() - then;
numReplies++;
```

实验结果：



```
"C:\Program Files\Java\jdk1.8.0_102\bin\java" ...
Contacting host 183.172.144.173 at port 9876
Sent message to /183.172.144.173:9876
Received message from /183.172.144.173:9876
Sent message to /183.172.144.173:9876
Received message from /183.172.144.173:9876
Sent message to /183.172.144.173:9876
Received message from /183.172.144.173:9876
Sent message to /183.172.144.173:9876
Received message from /183.172.144.173:9876
Sent message to /183.172.144.173:9876
Received message from /183.172.144.173:9876
Sent message to /183.172.144.173:9876
Received message from /183.172.144.173:9876
Sent message to /183.172.144.173:9876
Received message from /183.172.144.173:9876
Sent message to /183.172.144.173:9876
Received message from /183.172.144.173:9876
Sent message to /183.172.144.173:9876
Received message from /183.172.144.173:9876
Sent message to /183.172.144.173:9876
Received message from /183.172.144.173:9876
PING 0: true RTT: < 1 ms
PING 1: true RTT: < 1 ms
PING 2: true RTT: 4 ms
PING 3: true RTT: < 1 ms
PING 4: true RTT: < 1 ms
PING 5: true RTT: < 1 ms
PING 6: true RTT: < 1 ms
PING 7: true RTT: 4 ms
PING 8: true RTT: < 1 ms
PING 9: true RTT: < 1 ms

Process finished with exit code 0
```



实验思考与分析：

1, *simple SMTP 程序和常用的客户端在功能和结构上的比较：*

前者功能简单，不能发送附件，也不能一次发给多个人，更不能密送、抄送等，而且不能正确显示其发件人，但是可以在"由…代发"中可以看出。

结构上，simple SMTP 结构简单，容错性能差，各种发送过程中遇到的问题与错误都要靠发件人自己修改配置和程序来解决；而常用客户端则能处理许多复杂情况。相同的是其使用的协议相同----SMTP。

2, *UDP Pinger 和 windows 自带的 ping.exe 程序在功能、协议和输出结果上的比较:*

下图是 ping.exe 的运行结果：

```
C:\Users\liuyin14>ping 183.172.144.173

正在 Ping 183.172.144.173 具有 32 字节的数据:
来自 183.172.144.173 的回复: 字节=32 时间<1ms TTL=128
来自 183.172.144.173 的回复: 字节=32 时间<1ms TTL=128
来自 183.172.144.173 的回复: 字节=32 时间<1ms TTL=128
来自 183.172.144.173 的回复: 字节=32 时间<1ms TTL=128

183.172.144.173 的 Ping 统计信息:
    数据包: 已发送 = 4，已接收 = 4，丢失 = 0 (0% 丢失)，
往返行程的估计时间(以毫秒为单位):
    最短 = 0ms，最长 = 0ms，平均 = 0ms

C:\Users\liuyin14>
```

功能上：

ping.exe 能进行域名解析，获取并统计 ttl，还能显示有关数据包的发送接收的统计信息，还能对往返行程时间进行估计，这些都不是 UDP Pinger 具备的功能。

协议上:

Ping.exe 是基于网络层间的 ICMP 协议，而 UDP Pinger 是基于传输层的 UDP 协议。ICMP 报文是 IP 报文的有效负

载，故不需要指定其端口号。而 UDP 要指定端口号。

输出结果上：

UDP Pinger 只输出 rtt，ping.exe 输出 rtt，ttl，字节数，以及其他统计信息。

3 程序中的问题及解决办法:

我的日期不能显示，虽然在控制台上显示正确，但收件方不显示；

解决办法：多次尝试后，把程序中"DATE：" 中的"："删去了，就可以正确显示，原因可能是和下面的 date.getBytes("US-ASCII")有关，冒号可能不符合这种编码，导致冒号后的指令显示不了。

```
String date = "DATE " + dFormat.format(dDate) + "\r\n";

System.out.print(date);

os.write(date.getBytes("US-ASCII"));

String str = "";
```