

实验内容(只需要做题目 1-3)

- 计划课时：1.5
- 基本网络工具的使用
- InetAddress
- Socket、客户端程序
- ServerSocket、服务器、多线程服务器
- URL、URLConnection 与 Web 服务器
- UDP
- 非阻塞通信
- JavaMail
- 使用 Swing 显示 HTML 文档(创建简单浏览器)
- 基本考核：3.1，3.2
- **特别考核**：3.3，3.4，4，5，6，7

题目 1：基本网络工具的使用(基础知识)

1.使用 ipconfig 查看自己的网络适配器的属性--如 IP 地址。(使用“网络连接”也可查看，并且可以设置属性)

2.使用 ping 工具查看与对方电脑是否连通。

ping cec.jmu.edu.cn 与 ping 222.76.205.136 尝试回答：上面这两个命令中有何不同？有何关系？

比较 ping www.baidu.com 与 ping cec.jmu.edu.cn，分析返回结果有何不同？为什么会有这样的不同？

3. 使用 telnet 工具登录远程主机(win7 以上系统，需要先在“控制面板-程序-程序功能-启用或关闭 Windows 功能- Telnet 客户端”中安装 telnet)

telnet bbs.newsmth.net 尝试回答，上面这条命令连接的是远程主机的什么端口号？（百度之）

参考下面操作步骤， telnet cec.jmu.edu.cn 80，尝试回答：你从屏幕看到的是一些什么内容？返回的内容是不是 cec.jmu.edu.cn 的主页内容？尝试使用 GET/HTTP/1.0 命令，会返回什么？

说明：实际上此实验就是演示客户端(telnet)与远程服务器(cec.jmu.edu.cn)使用 HTTP 协议进行交互，其中 **GET/HTTP/1.0** 就是符合 HTTP 协议的命令，该命令含义详见：参考资料目录中的“HTTP 协议简介.doc”，思考：如果想请求学院其他网页，那么命令该怎么写？

操作:

- ❏ 1.telnet cec.jmu.edu.cn 80
- ❏ 2.ctrl+] //打开回显
- ❏ 3.按回车
- ❏ 4.输入 GET / HTTP/1.0 //协议, sen GET / HTTP/1.0
- ❏ 5.输入 Host: cec.jmu.edu.cn
- ❏ 6.按两下回车

4.使用 ftp 命令登录 ftp 站点(略):

[ftp 210.34.148.132](ftp://210.34.148.132)

根据屏幕提示输入用户名密码。登录以后使用 ? 查看当前可用命令。下面为可使用的命令列表。

常用命令: ls -- 查看当前目录的文件, dir --查看当前目录下的所有文件及目录,
cd--进入某个目录, get -- 下载某个文件(如 get JavaFTP 新增内容.txt) , ! 退出 ftp

尝试回答: 该命令连接的是远程 ftp 服务器的什么端口号?

5.使用 ConnectionWatcher 查看当前计算机使用的端口号(xp 下使用 Fport)

双击打开 ConnectionWatcher 启动若干联网的程序, 如 ie,qq,telnet,ftp, 然后查看本电脑已经使用的端口号及程序。

思考: 尝试分析 ConnectionWatcher 中运行结果。本机上有何程序联网, 使用了什么端口?

XP 下下载 Fport 工具, 命令行下执行里面的可执行文件。启动若干联网的程序, 如 ie, qq, telnet, ftp, 然后执行 Fport 工具查看本电脑已经使用的端口号及程序

题目 2: InetAddress 与 Socket、客户端-TCP 编程入门(基础知识、TCP 协议)

1.编写几个方法 (略)

- a.获得本地主机的 IP、主机名, 获得远程主机的 IP (查 JDK 文档)。
- b.指定主机名的 IP

参考代码: InetAddressTest.java

2.客户端程序编写: 参照 SocketTest.java 编写一个简单的连接远程服务器小程序 Client.java。观察返回结果?

- 将程序中的网址改成 bbs.newsmth.net, 端口号改为 23, 再次观察返回结果? 尝试使用 telnet bbs.newsmth.net 23。对比返回结果。
- 将程序中的网址改成 cec.jmu.edu.cn, 端口号改为 80 进一步观察返回结果?
回答: 没有任何返回结果, 这是为什么? //提示: 要使用 HTTP 协议获得数据, 如何改进?

指导:

- a.我们连接 cec.jmu.edu.cn 的 80 端口, 实际上使用了什么协议?

- b.该协议获取数据的语法是什么? (参见实验 题目 1.3),
- c.如何改写程序获取返回结果?

参考资料(进阶):

百度百科: 超文本传输协议 (主要看里面的 GET)

参考资料目录中的“HTTP 协议简介.doc”

3. 启动 Tomcat, 创建自己的 JSP 应用。然后尝试使用 Client.java, 遵循 HTTP 协议, 连接 Tomcat 服务器的相关 web 应用。观察返回结果。

4.测试套接字超时 (略)

将上题中的端口号改为 9999(即服务器没有提供服务的一个端口), 然后使用 Socket 的 setSoTimeout 方法, 设置 3 秒后超时。然后观察抛出了什么异常? 捕获这个异常, 提示出错信息。待完善: 见孙卫琴版 Socket 章节

5.进阶: 使用图形界面实现 SocketTest.java, 使用可中断套接字让客户端在连接过程中, 如果等待太久的情况下, 可手动中断连接。(略)

参考代码:

SocketTest.java

InterruptibleSocketTest.java (可中断套接字)

题目 3: ServerSocket 与服务器(TCP 基础)

1. 创建服务器 MenuServer.java, 在 8190 端口监听。客户端连接后, 进入**初始状态**, 在该状态下, 当用户输入 **quit** 的, 则服务器终止。用户在客户端输入 **menu**, 则显示菜单 (服务器将菜单传输给客户端)。然后用户可以根据传递过来的菜单, 输入相应的指令。服务器则根据客户端传过来的指令, 执行一些相关操作。

常见错误:

现象: Address is already in use...

原因: 有的时候服务器已经启动或者因为错误操作 (如频繁启停服务器) 造成相关服务器没有正常关闭, 占用了相关端口。

解决方案: 打开任务管理器, 找到 java.exe 或者 javaw.exe 并结束。

2. 使用 telnet localhost 8190 连接到该服务器, 测试 MenuServer。

常见错误:

现象: telnet 客户端连接到服务器, 但是并没有接收到服务器发送过来的信息。

原因: 服务器的 out.println 只是将字符发送到缓冲区, 并没有真正的发送到客户端。

解决方案: flush....

3. **考核点:** MenuServer 编写指导: 在服务器端写一个函数 public static void menu(.....){ 显示菜单..., 接受输入...}, 该函数主要负责显示菜单和接受客户端的输入, 并根据输入来调用相应的服务, menu 的输入参数应该是什么请思考?。菜单有 4 个选项:

1.显示服务器当前时间

2.您的 IP //显示客户端的 IP 与客户端的 IP 与主机名

3.文件搜索 //搜索服务器中符合条件的文件名并返回其绝对路径名

4.退出菜单

//仅仅退出菜单显示，退回初始状态而不断开连接

注意：菜单要能反复显示，只有当选择 4 的时候才能退出菜单。

4. **考核**：将 MenuServer 改造成多线程服务器。参见 ThreadedEchoServer.java。

5. **特别考核**：改写题目 2 的 Client.java，连接 MenuServer。观察返回结果？思考：如何让 Client.java 既可向 MenuServer 发送命令，同时还能接收 MenuServer 的信息。提示：使用多线程。

后期完善：可以使用图形界面，包括：IP 文本框(JTextField)、端口文本框(JTextField)、连接按钮(JButton)、命令文本框(JTextField,负责存放发送到服务器的字符串)、显示 JTextArea(负责显示服务器返回的字符串)，当点击连接按钮时，可以连接服务器，并显示可能返回的信息。

注意：试验中如果出现 Address already in use: JVM_Bind 异常，则修改成另外一个端口监听。或者直接使用任务管理器结束占用该端口的程序。

参考代码：

EchoServer.java

ThreadedEchoServer.java (多线程服务器)

题目 4：简易 UDP 服务器

任务 1：

使用 UDP 改写 MenuServer

任务 2：

编写类似 QQ 的聊天程序，允许两个用户在知道对方 IP 与端口的情况下，进行聊天。

进阶：

a. 实现文件传输。

b. 编写服务器端程序和客户端程序，使得两个用户即使不知道对方 IP，也能发送消息(在线时相互之间直接发送消息，不在线的时候可以发送离线消息)

高级进阶：开发简易的 HTTP 服务器(Socket 与 ServerSocket)

使用 Socket 与 ServerSocket 遵循 HTTP 协议模仿 Tomcat 开发一个简易的 HTTP 服务器。

要求：

1. 能返回静态页面，如 HTML

2. 进阶：能处理对 Servlet 请求。

参考资料:

[HTTP 协议简介](#)

[漫画: HTTP 协议极简教程, 傻瓜都能看懂!](#)

题目 5: 使用 Apache HTTPComponents 或 JDK 11 中的 Java HTTP Client 与 Web 服务器通讯(HTTP 协议)

开发一个查询成绩的程序: 用户输入用户名、密码、验证码, 即可自动查询教务处服务器, 获取成绩。注: Apache HTTPComponents 与 Java HTTP Client 任选一个即可

参考资料:

[HTTP Client 教程](#)

[Apache HTTPComponents 官网](#)

[Apache HTTPComponents 4.5 教程](#)

[JDK11 中的 HTTP Client-Introduction to the Java HTTP Client](#)

[JDK11 中的 HTTP Client-Examples and Recipes](#)

[HTTP 协议简介](#)

[漫画: HTTP 协议极简教程, 傻瓜都能看懂!](#)

题目 6: 网络爬虫(HTTP 协议与正则表达式)

使用简单 Socket 或 Apache HTTPComponents 或 JDK 11 中的 Java HTTP Client, 结合使用正则表达式, 将返回的信息中有用的搜索结果提取出来。有用的信息包括标题与该标题相对应的超链接信息

如: 搜索 2012, 则有用的信息如下所示:

链接:

```
<a href="information/2012-05/4732.shtml" target="_blank" ><font color='red'>2012</font>届毕业设计答辩时间通知(改在周日 5 月 27 日)</a>
```

有用的信息

链接: <a href="information/2012-05/4732.shtml"

标题: 2012 届毕业设计答辩时间通知(改在周日 5 月 27 日)

进阶(网络爬虫):

设计一个程序,可以根据输入的关键字将所有的通过该搜索引擎查找到的网页全部获取(需考虑如何翻页),并将这些网页以文件的形式下载到本地指定目录。
再进阶:将这些网页中的资源文件,如 doc 文件、rar 文件或者 img 文件下载下来。

参考资料:

正则表达式

参考代码:

GetTest.java(已废弃)

题目 7: 非阻塞通信

一种非阻塞的高效通信方式。

题目 8: JavaMail(SMTP 协议)

使用 SMTP 收发电子邮件。

参考代码: Core Java 2 高级技术 P168 MailTest.java

题目 9: 简易浏览器(HTTP 协议)