**深 圳 大 学 实 验 报 告**

**课 程 名 称： 计算机网络**

**实验项目名称： Socket编程**

**学 院： 计算机与软件学院**

**专 业： 计算机科学与技术**

**指 导 教 师： 邹永攀**

**报告人： 刘睿辰 学号： 2018152051 班级： 数计班**

**实 验 时 间： 2021年3月30日~2021年4月13日**

**实验报告提交时间： 2021年4月27日**

**教务处制**

|  |  |
| --- | --- |
| **一、实验目的：**   1. 学习网络编程基本概念、InetAddress的应用、URL的应用、URLConnection的应用； 2. 掌握Socket的TCP通信、 Socket的UDP通信。 | |
| **二、实验环境：**   1. 操作系统：Windows 10 操作系统； 2. 编程语言：Anaconda Python 3.6.3； 3. 电脑具有Internet连接；具有WampServer数据库以及IDEA java环境。 | |
| **三、实验内容：**   1. 获取本地机的名称和IP地址，并获取以下网站的IP地址，若存在多个IP地址，则要求全部返回。   网站1：[www.baidu.com](http://www.baidu.com)  网站2：[www.csdn.net](http://www.csdn.net)  网站3：www.google.com   1. 下载深圳大学首页<https://www.szu.edu.cn/>，并统计下载得到网页文件的大小。要求将深大主页上面的全部对象（图片、文本等）全部下载并保存到本地。 2. TCP通信部分：    1. 指令交互：客户端向服务器端发送Time命令，服务器端接受到该字符串后将服务器端当前时间返回给客户端；客户端向服务器端发送Exit命令，服务器端向客户端返回“Bye”后关闭连接退出程序；    2. 文件传输：客户端向服务器端发送文件请求，服务器端接收到请求后，先向其传输文件名，当客户端接收后，再向客户端传输文件。客户端接收文件，并对其改名（文件名可自行定义）和存在本地。 3. UDP通信部分：    1. 编写完整程序包含一个服务器端程序和多个客户端程序（不同客户端代码几乎一样），使多个客户端之间可以彼此通信聊天（类同微信、QQ）；    2. 客户端之间能够借助用户设计的UI界面显示聊天交互过程，并包含上述所有功能。 |
| **四、实验步骤：**   1. 获取本地机的名称和IP地址，并获取网站的IP地址，若存在多个IP地址，则要求全部返回。   为了得到本地机名称，我们可以用python的socket库下的gethostname函数得到本地机名称。此外为了得到IP地址，我们可以使用该函数：gethostbyname(name)函数，参数为本地机名称。事实上，运行这个函数我们可以得到如图1所示的结果，可以看到只有一个返回结果。  为了得到全部的IP地址，我们可以使用gethostbyname\_ex函数，这样可以得到全部的IP地址，如图2所示。通过和ipconfig指令得到的结果来做对比，发现结果是一致的。其中1/2行返回的是VMware Network Adapter VMnet1/8的IP地址，第三行是本机以太网IP地址。    图1. gethostbyname函数运行结果    图2. gethostbyname\_ex函数运行结果  代码如图3所示。由于IP地址位于元组的第三位，所以我们要访问的是hostip[2].  图3. 得到本机所有IP地址的python程序  我们还可以用gethostbyname\_ex方法来获取网站的IP地址。我们得到了网站的域名之后，通过该函数可以解析到网站的IP地址，如图4-图6所示，我们分别得到了百度、csdn以及谷歌的所有IP地址。    图4. 百度的所有IP地址    图5. csdn的所有IP地址    图6. 谷歌的所有IP地址   1. 下载深圳大学首页<https://www.szu.edu.cn/>，并统计下载得到网页文件的大小。要求将深大主页上面的全部对象（图片、文本等）全部下载并保存到本地。   方法：正则表达式匹配文件名称。   1. 首先，我们要明确网址<https://www.szu.edu.cn/>的网页文件都有什么，或者它们的类型都是什么样的。所以我们需要检查网页源代码，使用Fn+F12之后显示网站源代码。在Network选项可以看到网站的文件格式。例如，在JS部分，我们可以看到JavaScript文件，如图7所示。     图7. 网站的JavaScript文件  而在Img部分我们可以看到图片的文件格式，有png和jpg格式的图片。点击CSS可以看到所有的css文件，点击Media可以看到MP4文件（视频文件），如图8所示。  图8. 网站的MP4视频文件  通过对网站的仔细检查，发现资源文件格式基为.jpg（图片）、.png（图片）、.css（层叠样式表）、.js（JavaScript文件）以及.mp4（视频文件）。现在我们需要将所有的资源文件的URL路径保存下来方便我们爬取。如图8所示，以视频文件为例我们已经从Request URL中得知了MP4文件的文件路径，但是后面还有e=.mp4，所以我们可以将文件格式进行规范化。  以如下网址  https://www.szu.edu.cn/\_\_local/9/CA/0C/1333E74F83F686DB1888690982B\_4356DEE1\_15CA6A5.mp4?e=.mp4  为例，我们想要规范化该网址，所以我们可以先用python的split方法分离出后面的相对路径  1333E74F83F686DB1888690982B\_4356DEE1\_15CA6A5.mp4?e=.mp4  然后用python的find方法找到”MP4”首次出现的位置，加上文件后缀名的长度进行截取，就可以得到  1333E74F83F686DB1888690982B\_4356DEE1\_15CA6A5.mp4  同理，别的文件名我们统一进行标准化，得到标准化代码如图9所示。  图9. 文件名标准化   1. 现在我们知道了文件名标准化的方法，那么如何得到所有文件的URL地址呢？这里我们必须要用到python爬取网站库requests中的函数     这里返回一个名为reply 的响应对象。我们可以从这个对象中获取所有我们想要的信息。url自然就是我们访问主页的网址，header是一个请求头，里面对user-agent（中文名用户代理）进行伪造。从User-agent中服务器可以知道客户端的 操作系统类型和版本，电脑CPU类型，浏览器种类版本，浏览器渲染引擎等等。这是爬虫当中最最重要的一个请求头参数，所以一定要伪造，甚至多个。如果不进行伪造，而直接使用各种爬虫框架中自定义的user-agent，很容易被封禁。  常见的用户代理如图10所示。  图10. 常见的用户代理  得到了reply之后意味着我们想要的资源都在这里面。首先我们应该知道网站的编码，然后指定reply的编码方式为网站的编码方式。网站的编码方式在源代码<meta charset>中可以查看，然后获取网页文本信息写入文件即可。  写入文件我们可以用with open实现，参数设置为’w’代表打开一个文件只用于写入。还有很多其他的参数，例如’r’代表只读，’w+’代表读写等等。然后制定编码，用write函数实现写入什么内容，具体代码如图11所示。这里面我们将文本信息写入了SZUpage.html文件中，注意该文件保存的位置是与python文件一个文件夹的路径地址。  找到文件位置然后打开SZUpage.html，可以看到结果如图12所示。可以看到所有的文字都被保存了下来，但是原网页的所有图片都消失了，因为我们仅仅爬取了网站的文本信息。    图12. 网站文本信息  图11. 爬取网站文本信息  网站文本信息得到了之后，我们再来看网站别的资源的url地址。我们之前看到网站文件比较多，尽管可以用列举的方法一一列出，但是对于大型网站资源比较多的情况下，列举就不是一个很好的办法，为此我们引入正则表达式(regular expression)的方法进行字符串匹配。  我们首先观察网页，网站文件一般储存在src(source)和href(Hypertext Reference，超文本引用)中。在网页源代码中，src 是 source 的缩写，是指向外部资源的位置，指向的内部会迁入到文档中当前标签所在的位置；在请求 src 资源时会将其指向的资源下载并应用到当前文档中，例如js脚本，img图片和frame等元素。href用来建立当前元素和文档之间的链接。常用的有：link和a。  首先我们看一下正则表达式速查表，如图13所示。  图13. 正则表达式功能  我们先来看src里面的文件。检查网站代码，一般的src格式都是：    所以根据正则表达式表我们得到该正则表达式为    根据href中的内容，检查网页源代码发现，a或link和href之间有时候有其他参数，对比一下如图14所示。所以我们需要将这部分考虑在内。    图14. href需要判断多种情况  以上这些并不能说是全部的文件。例如，有一些图片被放进了div的style参数种，所以我们要将其提取出来。这个格式还是有不同，如图15所示，空格所在位置的不同导致我们要分门别类进行匹配。    图15. background-image不同样式  综上所述，正则表达式匹配文件的代码如图16所示。  图16. 正则表达式匹配文件  得到了文件路径之后，我们可以用之前规范化的方法进行规范化，再加上主机名就是文件的所在位置。   1. 前两步完成了文件的提取，接下来我们需要按照它们原来的相对路径，在我们的工作目录下重新建立这些文件夹，以方便我们提取文件。我们的方法是，将得到的文件路径，按照号进行分割，然后用os.mkdir逐层创建文件夹。代码如图17所示。   文件夹创建好之后就可以用写入(write)的方法来写入文件。在这里我们要注意一点，我们下载流文件的时候，我们需要将requests库中的stream设置为True，然后分块进行传输。因为流文件一般比较大，一次传输可能会有数据丢失。我们设置chunk(块)进行传输，大小设置为1024即可，代码如图18所示。  文件夹创建好之后就可以用写入(write)的方法来写入文件。在这里我们要注意一点，我们下载流文件的时候，我们需要将requests库中的stream设置为True，然后分块进行传输。因为流文件一般比较大，一次传输可能会有数据丢失。我们设置chunk(块)进行传输，大小设置为1024即可，代码如图18所示。  然后我们统计各个格式文件的大小总和。    图18. 传输流文件  图17. 创建文件夹  我们知道，python的os.path.getsize方法可以得到文件的大小。但是，观察新建的文件夹结构，我们可以发现有以下文件夹里面嵌套了一些文件夹，所以直接用这个方法是不够的。所以我们需要进入文件夹内部，遍历所有文件来计算文件大小总和。对于文件夹内部的子文件夹也有可能有文件夹，我们可以用递归的方式来进行计算。这里介绍两个os库的函数。  **os.listdir()**用于返回一个由文件名和目录名组成的列表；  os.path.isdir()用于判断对象是否为一个目录；  os.path.join().连接两个或更多的路径名组件。  结合以上三个函数以及我们叙述的递归方法，我们可以计算所有文件夹的大小。计算文件夹大小的代码如图19所示。  图19. 计算文件夹大小  然后将我们提取出来的文件夹分别计算大小，即可得到所有文件夹的大小。   1. 拓展部分   在这部分我们完成对站长工具<https://sc.chinaz.com/ppt/free.html里面ppt>网页的爬取。在这里面我们使用etree以及xpath的方法进行爬取。  函数解释：  etree.HTML()可以用来解析字符串格式的HTML文档对象，将传进去的字符串转变成\_Element对象。  xpath: 获取html源码中的内容并进行解析，返回的结果是一个列表，所以通常在取xpath()方法结果的时候，只取列表中的第一个元素。  首先通过get方法获取网页的文本信息，然后观察网页。每一个ppt选项链接都是有如图20所示的结构的。  图20. ppt在文本中的路径  所以，我们要找到xpath路径为"//div[@class='bot-div']/a/@href"，这样就得到了ppt所在网页的路径。对于主页面的所有ppt链接，我们都这样去做，既可以得到所有的链接地址。  然后，针对一个链接我们来进行爬取。首先获取ppt名称。打开网页源代码，点击网页中的ppt名称，可以看到源代码中有了定位，如图21所示。这样我们就得到了ppt名称的路径"//h1[@class='title']/text()"    图21. ppt名称路径  然后我们开始爬取ppt资源。在网页中找到ppt下载的位置，对应到网页源代码中如图22所示。  图22. ppt下载路径  这里我们选其中一个链接即可，以第一个为例，那么ppt的下载路径就是："//div[@class='download-url']/a[1]/@href"  得到了下载路径之后就用requests方法来爬取即可。代码如图23所示。    图23. 用etree和xpath来爬取网页ppt代码   1. TCP通信部分：    1. 指令交互：客户端向服务器端发送Time命令，服务器端接受到该字符串后将服务器端当前时间返回给客户端；客户端向服务器端发送Exit命令，服务器端向客户端返回“Bye”后关闭连接退出程序；   在任何类型的通信开始之前，网络应用程序都必须创建套接字。有两种类型的套接字，即基于文件的套接字和面向网络或者地址的套接字。在所有的地址家族之中，目前AF\_INET是使用得最广泛的。为了创建 TCP套接字，必须使用 SOCK\_STREAM 作为套接字类型。  此外，我们还需绑定地址关键字，AF\_INET下以元组的形式表示地址。常用bind((host,port))，host表示主机IP地址，port表示端口号。这一部分的代码如图24所示。  图24. 服务器设置TCP通信并绑定地址关键字  接下来，我们需要被动接受TCP客户端的连接。这里python使用的是accept方法。这个方法返回一个新的套接字以及链接地址。具体方法为    服务器与TCP客户端连接之后，就可以接受客户端发过来的信息了。python中我们用**recv**函数来接收数据，recv有一个参数，指定数据缓冲区的大小，默认设置为1024，单位为byte。发送数据可以采用**send**，参数就是想要发送的字符串。send前面是套接字，这个用来确定发送到哪个客户端。在服务器中，不管是接收数据还是发送数据，都要用编码的方式。发送的时候，将要传输的数据编码发送，一般以为编码方式。发送的时候要进行encode(加密)，接受的时候要decode(解密)。客户端的方法亦然。  例如，当客户端发送Time的时候，服务器首先要接收数据，然后进行解码得到Time指令。然后我们通过datetime库来获取当下时间。时间获取之后转成字符串，通过send方法进行发送。代码如图25所示。  图25. 服务器接收客户端指令  时间信息由服务器发送之后，客户端接收信息并解码就可以输出当前的时间信息了。  这里面的reply是确定好了的，如果需要我们手动输入reply的信息，可以调用input函数，然后让reply接受输入即可。   * 1. 文件传输：客户端向服务器端发送文件请求，服务器端接收到请求后，先向其传输文件名，当客户端接收后，再向客户端传输文件。客户端接收文件，并对其改名（文件名可自行定义）和存在本地。   这部分和指令交互的区别是，我们需要读取文件。在文件存在的条件下，这里面我们要调用之前提及的with open，只不过这里面调用参数来读取文件进行发送。值得注意的是，有一些文件可能比较大，如果一次性发送文件，那么客户端的recv缓冲区可能无法全部接收。所以我们可以每次发送1024 bytes的大小，直到全部发送完毕。  如果文件不存在，则返回No such file提示信息。  代码如图26所示。  图26. 发送文件数据的函数  接下来从客户端角度出发来接收文件。客户端需要接输入想要下载的文件，然后服务器端搜索是否存在该文件。如果存在，那么则进行接收。接受的过程依然与发送过程类似，发送的时候是以1024 bytes为一块进行发送，接受的时候也要以1024 bytes为一块进行接收。由于服务端发送了文件的大小，所以客户端需要计算有多少块（不足向上取整），然后用with open进行写入。代码如图27所示。  下载完成之后，我们还可以调用os.path.getsize计算下载量。如果没有该文件，那么就输出No such file，如果客户输入Exit可以直接退出。  用户细节：我们可以将服务器中所有的文件打印出来方便用户选择。可以调用之前提到过的os.listdir以及os.path.join来输出文件名。    图27. 将数据写入文件   * 1. 拓展部分：在服务器端存储一个包含学生学号、姓名、性别信息的数据库，客户端每次发送一个学生学号给服务器，服务器则返回对应该学号的学生姓名和性别（提示：需要解决TCP粘包问题）；以及任意格式文件的传输   首先启动WampServer建立一个虚拟的学生数据库，字段数为3，分别为学号、姓名以及性别，如图28所示。  图28. 建立学生信息数据库（模拟）  我们以指令交互的代码为基础来添加这部分功能。我们规定查找某个学生要以命令为起始以作区分。  为了实现python连接数据库，我们需要下载pymysql库，然后使用connect方法连接数据库，如图29所示。  图29. python连接数据库  其中，host和user是数据库参数，database指明了是哪一个数据库，cursorclass=pymysql.cursors.DictCursor设置数据库返回结果为字典类型。然后cursor是一个游标，方便我们访问数据库。  当我们的用户输入了一个学号之后，我们要进行查找。mysql数据库中查找语句为select，我们需要截取后面的学号信息，然后进行查找即可。查找语句储存在sql中，然后需要调用execute方法来执行。这里介绍两个cursor的属性:  cursor.rowcount：返回符合查找条件的元组个数；  cursor.fetchall()：返回结果集，由于我们设置了以字典形式返回，所以它也是一个字典型变量。  这里需要解决TCP粘包的问题。我们知道我们需要返回的是学生姓名以及学生性别，这两个量是粘连在一起的。所以我们在发送给客户端的时候就用号来进行分离，然后客户端调用split函数来进行分离。  (split函数：分隔字符串。参数为一个字符，该功能可以按照该字符来切割字符串，以列表的形式返回。当然，split函数也可以存在第二个参数，表示分割的个数。)  服务器端代码如图30所示。  图30. 查找学生--服务器端代码  客户端代码如图31所示。这里我们调用split函数来解决TCP粘包问题，第二个参数设置为2，代表用号将服务器发来的结果分割成3块。结合图30中的代码，可以看到学生姓名在第二个，学生性别在第三个，按照这个顺序输出信息即可。  图31. 查找学生—客户端代码  此外，关于任意格式文件的收取，由于我们原来是按照二进制方法来读取文件，而任何格式的文件实际上都是由二进制组成的，所以我们的程序可以做到任意格式文件的接收。结果见实验结果。   1. UDP通信部分：    1. 编写完整程序包含一个服务器端程序和多个客户端程序（不同客户端代码几乎一样），使多个客户端之间可以彼此通信聊天（类同微信、QQ）；   首先我们实现UDP客户端与服务器之间的沟通。首先，服务器需要确定socket类型。由于这里面是基于UDP协议的通信，所以我们选择SOCK\_DGRAM。这是是无保障的面向消息的socket，主要用于在网络上发广播信息。它基于UDP协议，专门用于局域网，基于广播。然后我们需要绑定主机以及端口号。与TCP协议不同的是，我们不需要accept方法以及设置监听个数，直接就可以接收与服务器连接的客户端的信息。  一般情况下，客户端和服务器都使用sendto方法来发送信息，用recvfrom方法来接收信息。通信流程如图32所示。    图32. UDP通信流程  接下来我们讨论客户端之间的通信问题。我们已经确定了客户端与服务器的通信，那么我们设想如果以服务器作为中转站，将服务器作为一个传话筒，这样就可以实现客户端之间的联系。传输过程如图33所示。    图33. 客户端通过服务器彼此沟通  我们针对图33进行解释。假如是信息的发送方，   1. 将想要发送的信息，也就是，以及想要发送的目标发送到服务器。我们可以用将两部分信息分割开来； 2. 服务器接收到的请求信息之后，需要起到中转站的作用。我们需要将发起通话的发信人信息发送给，同时发送的端口号，告诉收信人是谁在跟他通话。收到来自服务器的信息之后，就知道是谁在跟它通话、通话的内容是什么，这两个信息用^来分割； 3. 收到了的信息之后，可以输入想要回复的信息，然后告诉服务器这是回复信息，所以要告知服务器的端口号； 4. 服务器接收到回复信息之后，将信息返回给，并将的端口号告知。   以上就是两客户端通信的基本过程。  代码如图34所示。    图34. UDP客户端通信服务器端代码  为了确定是谁先来发信息，我们要在客户端设置两个条件，一个是接收信息，一个是准备发送信息。这个判断由服务端确定，这里由服务端来指定发送信息的客户端是哪一个。  发送信息：如图35所示。  图35. UDP客户端通信客户端代码--发送信息  这里我们输入一个客户端，这里所有的客户端由服务器发送给客户端，然后客户端选择其中一个客户端进行通话。这里我们要设置一个判断条件，就是选择的是哪一个，然后是否存在这个客户端。然后将发送的信息和目标一起发送到服务器。  接收信息：如图36所示。  图36. UDP客户端通信客户端代码—接收信息  接收信息的话，我们需要首先用^号先进行分割，得到了发信人以及通话的内容。  现在我们得到了发信人以及会话内容。然后输入我们想要回复的内容，发送到服务器即可。   * 1. 客户端之间能够借助用户设计的UI界面显示聊天交互过程，并包含上述所有功能。   这里面我们用Java来实现通话功能。  Java通过DatagramPacket类和DatagramSocket类来使用UDP套接字，客户端和服务器端都通过DatagramSocket的send方法和receive方法来发送和接收数据，用DatagramPacket来包装需要发送或者接收到的数据。   * + 1. 发送信息时，Java创建一个包含待发送信息的DatagramPacket实例，并将其作为参数传递给DatagramSocket实例的send（）方法；     2. 接收信息时，Java程序首先创建一个DatagramPacket实例，该实例预先分配了一些空间，并将接收到的信息存放在该空间中，然后把该实例作为参数传递给DatagramSocket实例的receive方法。   有关UDP通信部分，Java的实现方法和python的实现方法是类似的，用法区别在于以上两点。现在叙述界面的构造。  我们使用的是JPanel，它是 Java图形用户界面(GUI)工具包swing中的面板容器类。我们用到的类型有：JTextArea(文本域组件)、JButton(按钮组件)等等。同时我们注意到，当一个容器内放置了许多组件，而容器的显示区域不足以同时显示所有组件时，如果让容器带滚动条，通过移动滚动条的滑块，容器中位置上的组件就能看到。滚动面板JScrollPane能实现这样的要求，JScrollPane是带有滚动条的面板。JScrollPane是Container类的子类，也是一种容器，但是只能添加一个组件。  我们首先设想我们的界面的大致形状，如图37所示。    图37. UDP通信界面  然后定义组件，并设置布局。代码如图38所示。    图38. 定义控件以及进行布局  然后我们需要将设置send按钮，并为其设置监听事件，实现点击按钮可以进行发送功能。代码如图39所示。exit按钮则监听system.exit(0)事件。  图39. 定义按钮以及监听事件  然后添加内容即可。然后还要设置setVisible(true)，这是允许JVM可以根据数据模型执行paint方法开始画图并显示到屏幕上了，并不是显示图形，而是可以运行开始画图了。然后设置一个默认的关闭操作，结束进程。代码如图40所示。    图40. 添加组件、可视化以及默认关闭操作  这样，一个聊天界面就制作完毕了。两个客户端的代码是类似的。相关结果位于实验结果部分。  拓展部分：这部分我们主要是UI界面的美化，给聊天界面添加背景图。  对于JPanel来说，添加背景图片有很多方法，例如重写panel方法等等。这里面我们采取相对容易一点的方法，直接在JTextArea部分添加背景图。首先声明Image型变量，接收资源图片。然后写paint函数，打印图片并设置图片随聊天框大小改变而改变。要用到的工具如下：  paintComponent是绘制容器中的组件；应该绘制组件的内容时调用此方法；例如首次显示组件或者组件已损坏并需要修复时。Graphics 参数中的矩形框设置为需要绘制的区域；  getParent方法：该函数获得一个指定子窗口的父[窗口句柄](http://baike.baidu.com/view/1452762.htm" \t "_blank)。例如，通过getSize方法得到窗口大小，这里为了实现图片大小随着窗口大小改变而改变，应该通过这个函数获得窗口当前大小，然后相应地进行绘制；  drawImage方法：绘制指定图像中已缩放到适合指定矩形内部的图像。  函数原型：drawImage (image, x, y, width, height, observer).参数作用都比较显然，其中observer代表当转换了更多图像时要通知的对象，这里不需要通知任何对象，设置为null即可。  代码如图41所示。    图41. 界面美化—添加背景图片 |
| **五、实验结果：**   1. 实验4.1部分关于爬取深大主页的结果。   在python文件的工作目录中，我们看到了如图42所示的文件夹。  图42. 爬取深大主页之后得到的文件夹  为了验证爬取下来的文件夹是否与网站的文件相对路径吻合，我们以src为例观察一下网站的src，如图43所示。可以看到，文件夹与相对路径是一样的，点击进文件夹来看也是一样的结果。  图43. 网站文件夹  以其中一个文件夹js为例，可以看到如图44所示，和网站的js文件夹是一样的。  图44. js文件夹文件内容  打开其中一个js文件，如图45所示可以看到其内容并没有出现乱码的情况，说明文件编码是正确的。  图45. JavaScript文件正常  接下来我们展示一下统计数据，如图46所示，所有文件夹的大小都被统计了出来。    图46. 各文件夹大小统计结果  实验4.1拓展部分实验结果  如图47所示，所有的ppt下载相对地址都被打印了出来。    图47. 所有ppt文件链接地址  再来看文件名称的打印，如图48所示，我们截取了一部分文件名称以及ppt的名称，可以看到文件名称以及ppt的名称都是正确的。    图48. 文件名称以及ppt名称的打印  再来到python工作目录，可以看到所有的下载好的rar文件，如图49所示。  图49. 所有下载好的文件  文件打开之后，可以看到文件都是正常的，如图50所示。  图50. rar文件都是正常的   1. TCP通信部分结果    1. TCP指令交互   1）输入Time指令之后返回了系统时间，如图51所示。    图51. Time指令返回结果  2）输入一般的语句，可以进行通信，如图52所示。    图52. TCP聊天结果   * 1. 输入Exit，可以退出程序。如图53所示。     图53. Exit退出程序   * 1. TCP文件传输   我们尝试ls操作，可以得到如图54所示的结果。    图54. 列举服务器所有文件  然后调用download操作，可以进行下载，其中包括对违法情况的判断。然后调用Exit指令，退出程序，如图55所示。    图55. download操作下载文件  进入桌面，看到下载的文件，如图56所示。下载文件大小与统计下载结果对比来说是一样的，可以说明下载是正常的。打开了文件之后，发现文件都是正常的。  图56. download操作下载文件显示在文件夹中   * 1. TCP与数据库的结合与任意格式文件的下载   如图57所示，我们看到了查询数据库的结果，这与数据库中存储的数据是吻合的。此外还包括为违法情况的判断。  图57. 连接数据库并查询学生信息结果  任意格式文件的下载：图片我们之前下载过，这里我们以视频、音频为例。首先可以看到，MPEG-4音频文件大小为3.37MB，视频文件大小为2.89MB，如图58所示。    图58. 下载其他格式文件的大小  然后我们用我们的程序进行下载，如图59所示。可以看到文件大小是接近的。为此我们检查一下文件夹中的文件，发现播放都是正常的，所以文件格式没有问题。    图59. 下载其他格式文件正常  此外，我们通过优化代码，提供重新命名服务。以其中一个python文件为例，我们将其改名后可以看到client所在的文件夹已经有了新命名的文件，如图60和图61所示。      图61. 文件改名成功  图60. 文件改名   1. UDP通信部分结果    1. UDP通信   这里面我们提供1个服务器和4个客户端，服务器为server，客户端分别命名为client 1、client 2、client 3以及client 4。  启动服务器以及所有客户端，将会出现如图62所示的结果。所有客户端的信息都被打印了出来，这就相当于是一个好友列表。然后选择一个好友进行通讯。    图62. 客户端收到好友列表并选择通讯对象  这里我们以client 3联系client 1，client 4联系client 2为例，先来看client 3联系client 1的结果，如图63所示。图63(a)是client 3端的通话信息，图63(b)是client 1端的通话信息。通话的最后，client 3发送Exit指令，然后双方退出通话。这里包括了对违法情况的判断。    图63(a). client 3向client 1答复信息    图63(b). client 1向client 3答复信息  我们看到服务器端显示了聊天的过程，如图64所示。  图64. client 1和client 3沟通过程  同时，client 4向client 2发送信息，然后client 2向client 4答复信息，如图65所示。图65(a)是client 4端的通话信息，图65(b)是client 2端的通话信息。    图65(a). client 4向client2发送信息    图65(b). client 2向client4答复信息  我们看到服务器端显示了聊天的过程，如图66所示。  图66. 服务器端体现交互信息   * 1. UDP通信Java界面，如图67所示。   图67. Java实现界面体现交互信息  点击send可以发送信息，点击exit可以关闭窗口。 |
| **六、实验小结：**   1. 通过python编程的方法，掌握了如果获取本机IP以及获取网站IP的方法。同时安装了requests库来进行爬虫功能的实现。爬虫方法有很多，类似正则表达式、xpath以及Beautifulsoup等等，每一种方法都有其独到之处。但是，相比之下，正则表达式是一种更加万能的方法； 2. 本次实验利用python变成，实现了TCP指令交互、文件传输以及结合数据库，用TCP来实现查询的功能。此外，我们用python实现了客户端之间的UDP通信，并用 Java图形用户界面(GUI)工具包swing中的面板容器类JPanel来进行了可视化功能； 3. 本次实验内容虽然很多，但是加强了python编程能力、加深了对TCP/UDP协议的理解，同时学会了用多种方法进行爬虫来爬取网页资源。 |
| **七、实验附件：**   1. Socket是整个python代码文件夹，其中包括socket1.1、socket1.2、socket2.1、socket2.2；    1. socket1.1包含getPCIP.py，用于获取本机IP；getIP.py用于获取网站IP；SZU\_fetcher.py用于抓取深大主页文件。同时附加的文件夹都是抓好的深大主页文件，具体的在实验内容已经详细阐述；    2. socket1.1包含fetch\_ppt.py，这个是实验4.1的附加部分，用于抓取ppt文件；附加的文件夹是抓好的ppt文件压缩包；    3. socket2.1包含client.py和server.py，这个是关于TCP指令交互的；client\_file.py和server\_file.py，这个是关于TCP文件交互的；    4. socket2.2包含server\_UDP.py、client\_UDP.py、client\_UDP\_1.py、client\_UDP\_2.py以及client\_UDP\_3.py这是一个UDP服务器与4个UDP客户端。分别运行就可以进行UDP通信。 2. UDPconnect是一个Java工程文件，里面src文件夹中两个java文件模拟的是UDP两个客户端之间的沟通。 |
| **指导教师批阅意见：**  **成绩评定：**  **指导教师签字：邹永攀**  年 月 日 |
| 备注： |