

四川大學

本科生课程设计



题 目 Query-flooding-based Resource Sharer

课 程 名 计算机网络和分布式系统

任课教师 宋万忠

学 院 软件学院

专 业 软件工程

学生姓名 秦旭坤

学 号 2019141460443 年级 2019

目录

Query-flooding-based Resource Sharer	3
一、 开发环境与工具.....	3
1. 开发环境	3
2. 主要使用的模块:	3
3. 开发工具	3
二、 系统设计	3
1. main.py	3
2. client.py	3
3. load.py	3
4. query.py	4
5. listen.py	4
6. search.py	4
7. frozen_dir.py.....	4
三、 算法设计.....	4
四、 测试与使用说明.....	5
1. 使用的先期准备	5
2. 正常使用流程:	6
3. 程序的其它路径	9

Query-flooding-based Resource Sharer

一、开发环境与工具

1. 开发环境

Python 3.6

2. 主要使用的模块:

socket, sys, os, threading

3. 开发工具

Pycharm

二、系统设计

1. main.py

main.py 是程序的入口文件，文件中定义了 main 函数，主要用于进行对线程的管理和控制。该应用程序运行过程中主要运行三个线程。客户端线程主要负责与用户的交互以及发出 query 请求，监听 ACK 请求和发出 get 请求。UDP 监听线程主要用于监听 UDP 端口，接收其它对等方发出的 query 请求。TCP 监听线程则负责监听 TCP 端口，接收其它对等方发出的 get 请求，并向请求文件的对等方发送文件。

2. client.py

client.py 定义了程序的 client()函数，该函数运行于第一个线程上，主要用于实现与客户端与用户的交互功能，并根据用户指令对另外的两个线程进行操纵。

3. load.py

load.py 中定义了 getIPList()函数，主要用于获取用户在本地设置的已知 IP 地址列表，用于进行查询请求的转发。

4. query.py

query.py 主要用于向其它设备或本设备的其它端口发送请求，其中定义了 query(), get()和 send_quit()三个函数。其中，query()函数用于向局域网中的所有设备发送 UDP 广播，用以向所有设备请求所需的文件。而 get()函数则用于向第一个发回 ACK 信息的设备请求建立 TCP 连接，来下载文件。第三个函数，send_quit()的作用则是当用户输入退出指令之后，用来向两个监听线程发送信息，让监听线程停止工作。

5. listen.py

listen.py 主要用于监听本设备的端口，其中定义了 listen_query(), listen_ack(), listen_get()三个函数。第一个函数，listen_query()的作用是用于监听 UDP 端口（本程序设置为 16380），对其他设备发出的 query 请求做出响应，同时，向 IP 列表中的其它设备转发这个 query 信息。而 listen_ack()函数的作用则是在发送 query 请求之后监听 UDP 端口（本程序设定为 17065），并向客户端返回 ACK 信息中的数据。最后一个 listen_get()函数的作用是用来监听 TCP 端口（本程序设定为 16381），对请求方传来的 get 请求做出响应，向请求文件的对等方发送文件。

6. search.py

该文件中定义了 search()函数，主要用于在收到 query 请求后，在本机的 share 文件夹中搜索相应的文件，若存在该文件，则返回文件路径，否则返回 None，表示本机没有相应的文件。

7. frozen_dir.py

该文件中定义了 app_path()函数，其作用是判断当前文件是以可执行文件的方式运行还是以 python 文件的方式运行，并根据文件类型返回不同的文件路径，用以解决 python 文件打包后 __file__ 指令和相对路径失效的问题。

三、算法设计

该系统的主要流程如下：

1. 用户打开程序，三条线程开始运作。
2. 用户进行 get 操作，客户端线程向局域网中广播 query 请求。query 请求的格

式为“query+文件名+请求方 IP 地址+TTL”

3. 收到 query 请求的对等方解析 query 请求，并在本地的 share 文件夹中遍历所有文件和子目录，寻找相对应的文件。
4. 若找到相应的文件，则向请求方发送 ACK 信息，ACK 信息格式为“ACK+文件路径+文件大小”。而如果在本地未找到对应文件，则向对等方存储的 IP 列表中的主机转发这条 query 请求，但要注意的是，为了防止两个对等方之间循环发送 query 请求，需要将 query 请求的 TTL-1。（进行转发的操作需要双方都具有公网 IP，否则无法直接连接）
5. 请求方若超过 10s 未接收到 ACK 信息，则认为无法搜索到对应文件。当请求方接收到 ACK 信息后，需要对 ACK 信息进行解析，同时，向发回 ACK 信息的对等方发送 get 请求，请求通过 TCP 进行文件传输。
6. 对等方接到 get 请求之后同意进行 TCP 连接，并向请求方进行文件传输。当文件传输结束后，向请求方发送 b'end'表示文件传输结束。
7. 请求方收到文件数据，并将其写入系统硬盘当中。当收到 b'end'或是空数据包之后，意为文件传输结束，则返回用户指令。
8. 用户输入 quit 指令，客户端线程向 UDP 端口（17063）和 TCP 端口（17064）分别发送 quit 信息，两端口收到该信息之后，对应的监听线程停止运行，然后，客户端线程也停止运行，正常退出程序。

四、测试与使用说明

1. 使用的先期准备

由于 Windows 10 等系统为了保证安全性，对于网络的设置做出了限制，因此，要正常的使用该程序，需要提前做出一定的修改。

以 Windows 10 为例，我们需要打开控制面板，点击系统与安全-Windows Defender 防火墙-高级设置

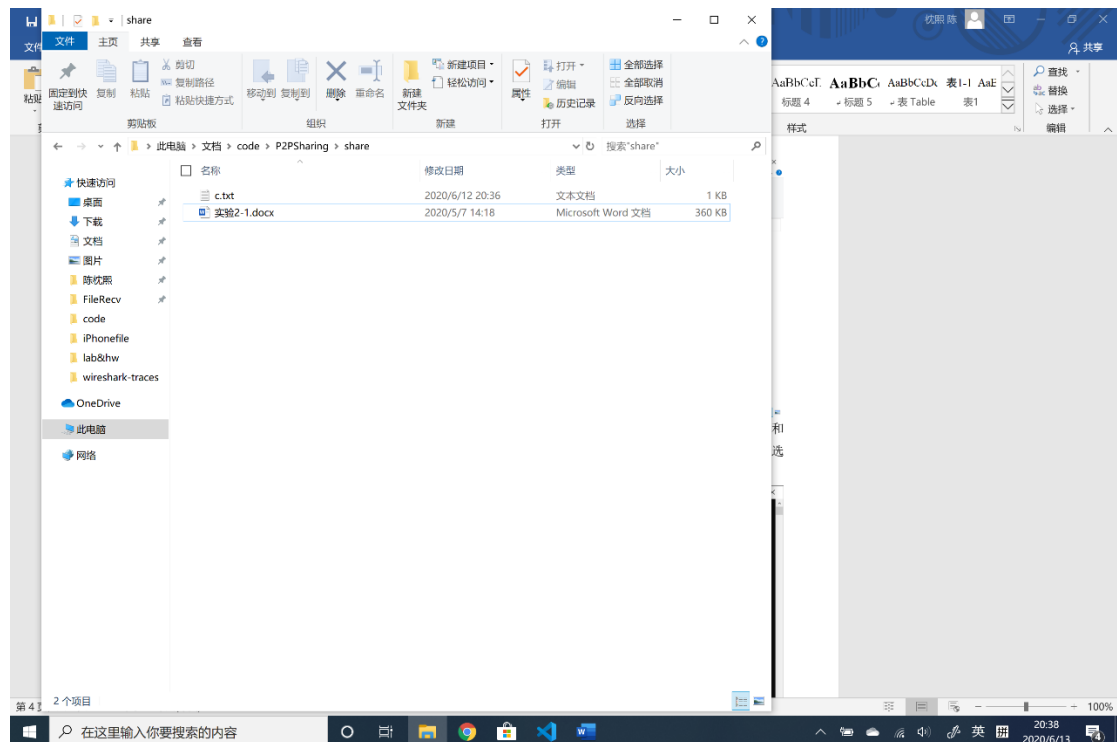


之后，依次点击入站规则-新建规则，选择端口，开放 17063 端口为 UDP 端口。以同样的操作开放 UDP 端口 17065 以及 TCP 端口 17064。同时，在运行应用程序时，最好将允许网络的专用网络和公用网络都勾上。

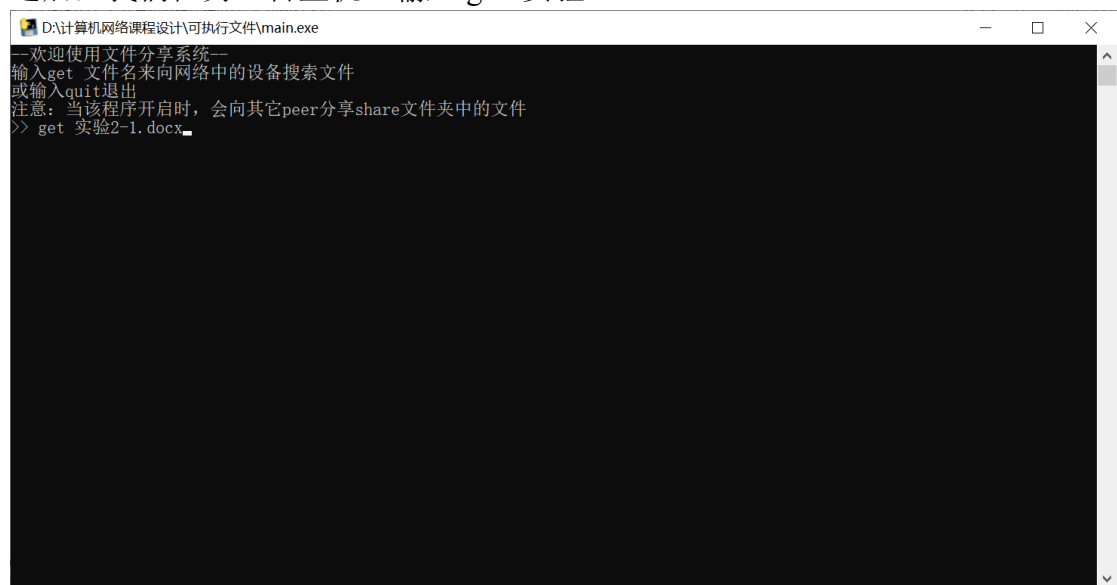
2. 正常使用流程：

同时在两台主机上打开该文件（可以运行可执行文件，也可以直接通过执行 main.py 来运行，实际上，后者的速度比前者更快）。

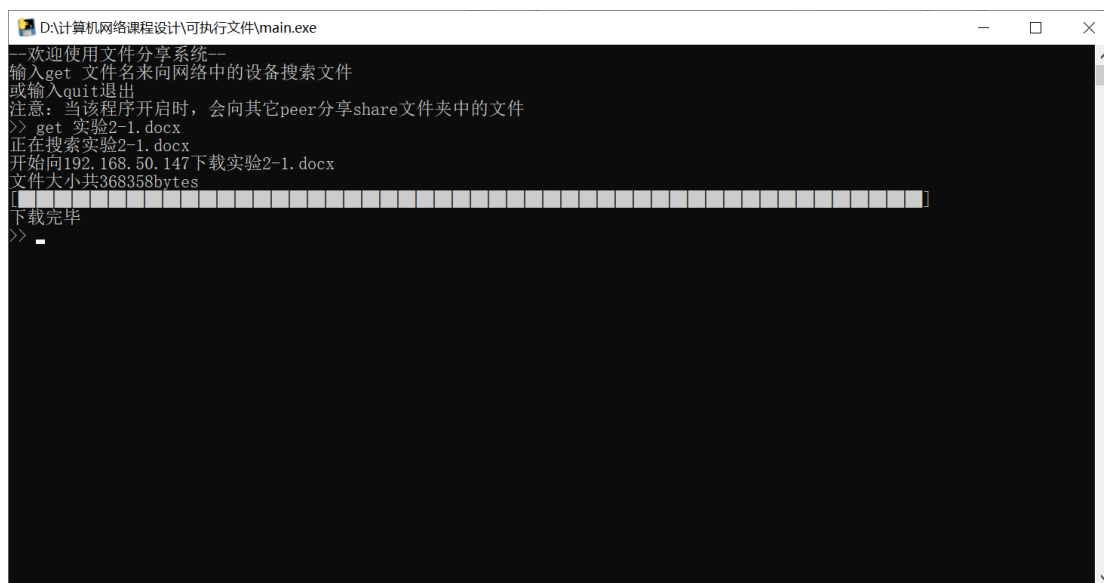
如图，我们首先在其中一台主机中程序的 share 文件夹中放置我们要传输的文件实验 2-1.docx



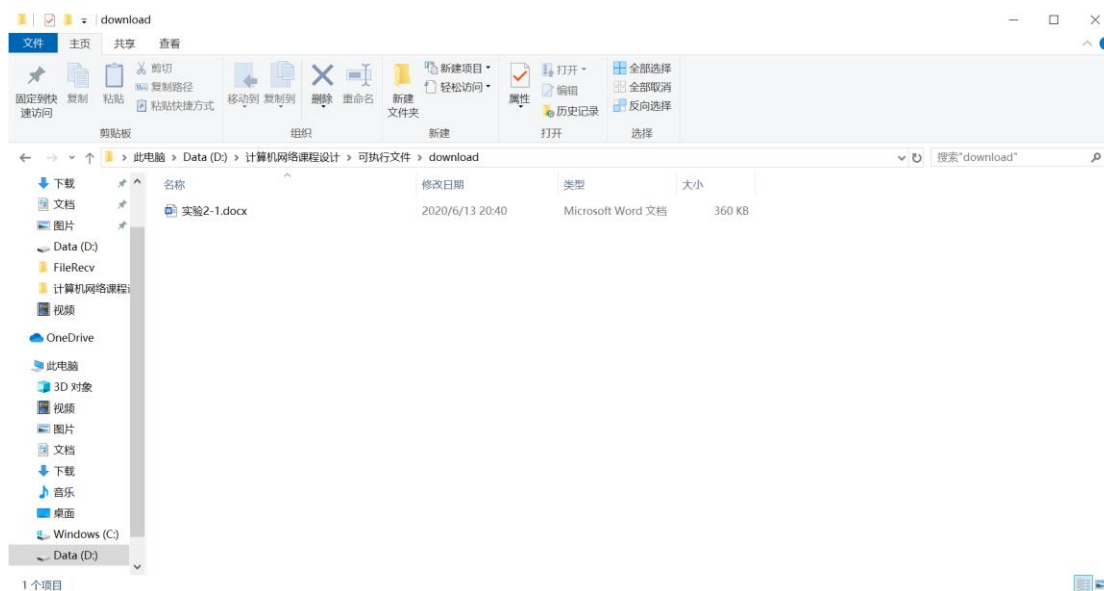
之后，我们在另一台主机上输入 `get 实验 2-1.docx`



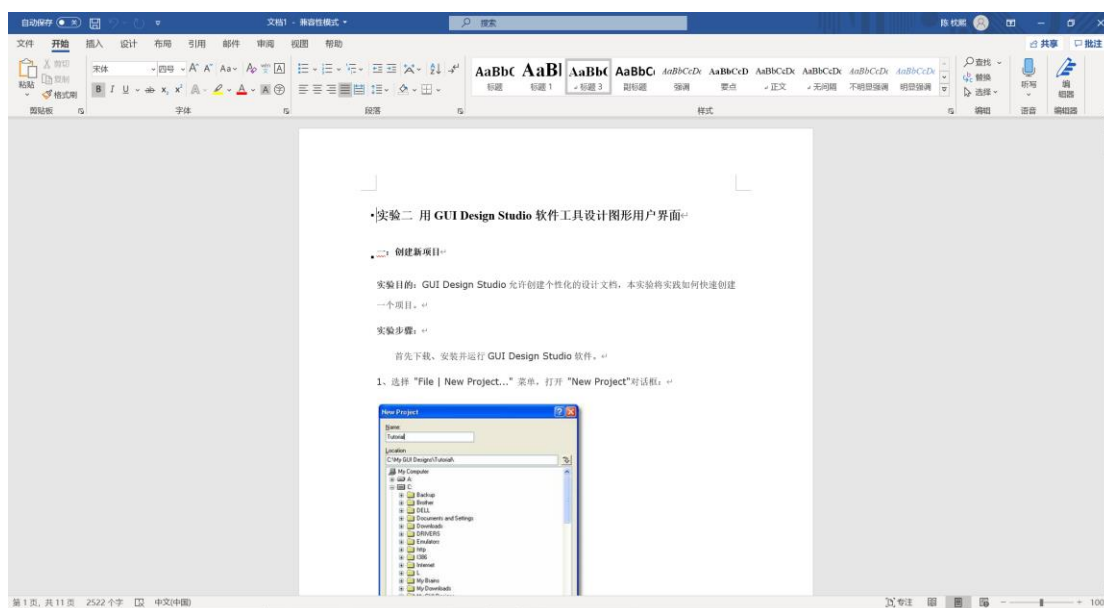
之后，便会开始文件的搜索以及下载。



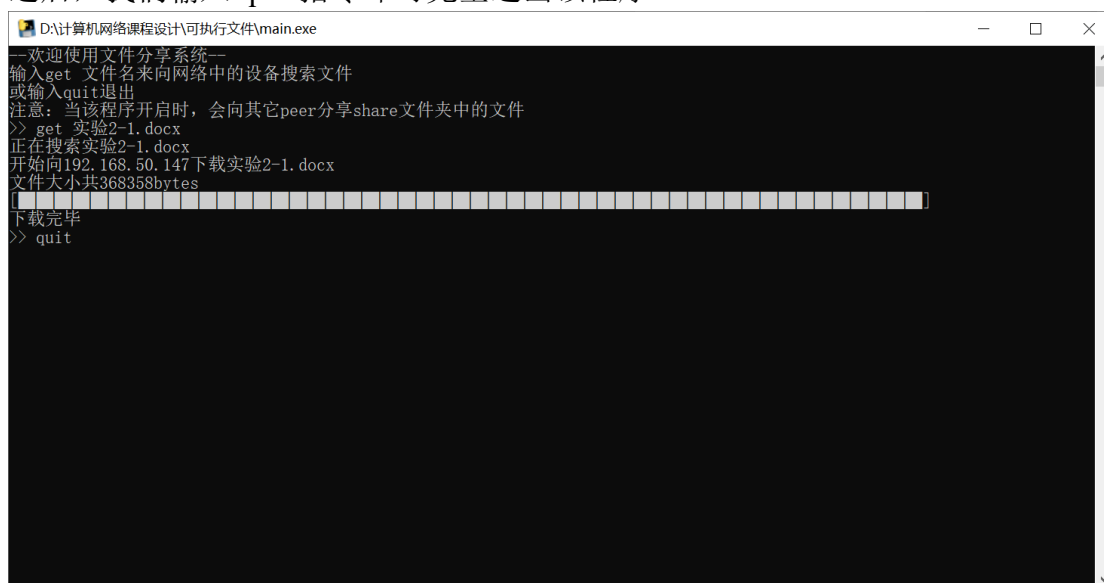
文件下载完毕后，我们可以在本机的 download 文件夹下找到该文件。



同时，打开后，我们可以看到文件被完整地传输了过来。



之后，我们输入 `quit` 指令即可完全退出该程序。



3. 程序的其它路径

该程序对一些常见的非正常路径进行了处理。比如指令错误、网络中断、找不到该文件等。

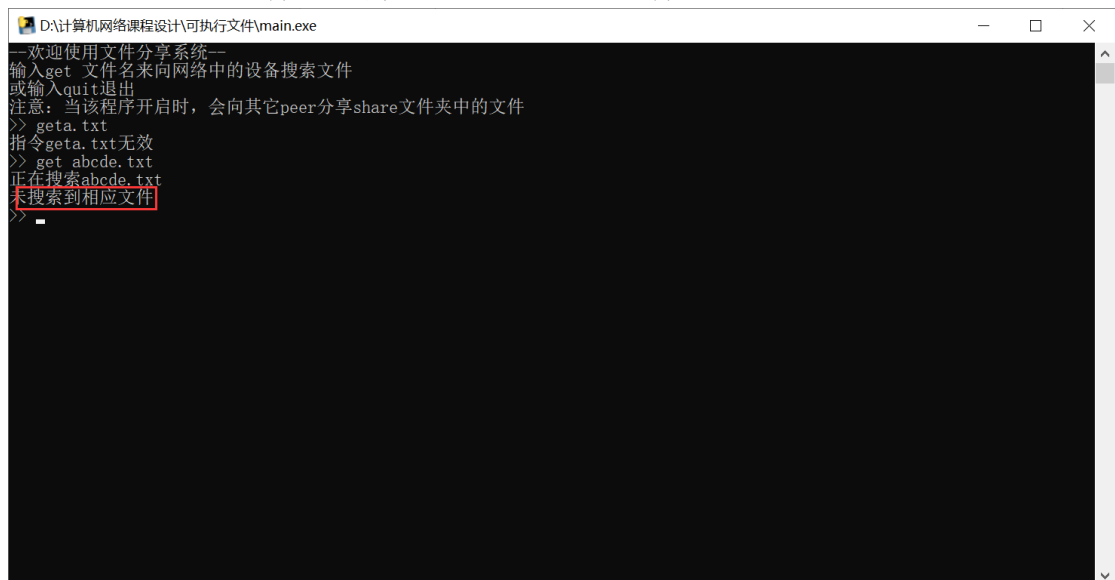
当用户输入错误指令时，系统会给出相应的提示。



```

D:\计算机网络课程设计\可执行文件\main.exe
—欢迎使用文件分享系统—
输入get 文件名来向网络中的设备搜索文件
或输入quit退出
注意：当该程序开启时，会向其它peer分享share文件夹中的文件
>> geta.txt
指令geta.txt无效
>>
  
```

当搜索不到所需文件时，会提示为搜索到该文件。



```

D:\计算机网络课程设计\可执行文件\main.exe
—欢迎使用文件分享系统—
输入get 文件名来向网络中的设备搜索文件
或输入quit退出
注意：当该程序开启时，会向其它peer分享share文件夹中的文件
>> geta.txt
指令geta.txt无效
>> get abcde.txt
正在搜索abcde.txt
未搜索到相应文件
>>
  
```

同理，当发生其它错误时，该系统也会将错误信息输出在控制台上，用以便于错误的解决。