开发的基本思路是，实现一个图形界面的python应用，包括分享文件，搜索并下载文件，管理分享的文件三部分功能。

启动程序，有客户端、监听udp、监听tcp三个线程同时启动

**客户端**

其中，分享文件要求用户输入本地文件的绝对路径，程序判断文件存在，则以绝对路径的形式加入文本文件url.txt中。

下载文件首先要求用户输入文件关键字，程序通过udp向局域网内广播查询消息，格式为

*message = 'query' + ' ' + keyword + ' ' + ip + ' ' + str(getudp\_ack\_port())+' '+ttl*

其中‘query‘是查询标志，keyword是搜索的关键词，ip是本机ip，udp\_ack的接收端口，以及ttl转发次数。

同时，对udp的ack监听listen\_udp启动。listen\_udp的搜索时间通过settimeout()确定，暂定为2s，如果两秒内未收到任何ack信息，则显示提示消息，若收到，则存储所有消息，绘制一个图形界面显示搜索到的信息。

用户可以选择需要的文件点击下载进行下载。

**监听udp**

监听开头为“query”的查询udp消息，如果收到则提取keyword与本地url.txt文件中分享的文件进行对比，如果找到类似的文件，则将一个或多个类似的文件的绝对路径打包，向（ip,udp\_ack\_port）发送一个确认消息，格式为

ack + filepath1\_file1\_size + filepath2\_file2 + ……

当用户选择需要下载的文件时，此线程调用函数download(source,address)开始从指定的地址下载指定的文件。文件的下载使用TCP。

Download（）函数建立一个socket与传入的那个地址链接，根据source这个文件路径，和address这个地址，编辑一条请求消息，消息开头是“down”

格式为：

Message = 'down ' + filepath

目的主机的返回消息被受到后开始下载，打开文件夹统一目录下的文件夹download开始向里下载对应的文件，同时有进度条显示（为了看到进度条，程序中设置了一个sleep（）实际使用时可以删除）

下载完成后由弹窗提示，文件可以在download文件夹下找到。

**监听tcp**

监听tcp线程运行listen\_download()函数，相当于服务器，等待网络中发起的down链接。当连接到请求主机并受到down请求后，解析请求，获取资源路径打开资源并传输。（注意，由于资源的打开使用了open（）打开文件夹和大文件可能出现异常，只获取异常，并没有处理此的问题）每次读取tcp最大的传输数量进行传书。在文件传输结束后发送‘end’信号，继续等待连接。

遇到的最大问题基本上都是关于图形化编程的，tkinter图形界面的逻辑没有学过，相较于有很多便利框架的js前端开发，显得有些繁杂。从0开始看csdn看文档看谷歌编程，导致出现了很多意想不到的bug和很多难以实现和修正的操作。

主要问题有：

1.button组件的command命令只能绑定无参的函数，如果要传入参数要使用使用StringVar()和Entry textvariable对参数进行绑定或者使用lambda，只能照着文章写但并不明白原理。若用到for in循环，传入的参数并不能绑定到每一个button上，因此还copy了部分事件处理的方法，使用bind绑定每一个按钮的事件，使用了更多内存，且按钮点击效果消失。

解决：减少了部分图形界面的逻辑，代码变得冗长复杂。允许图形界面出现部分bug，只要不特别影响核心功能都视为成功。

2.