



上海交通大学

SHANGHAI JIAO TONG UNIVERSITY

计算机通信网络原理与技术课程实践

FTP文件传输系统的实现

姓名： 张敏

学号： 5140219246

2016 年 12 月 29 日

目录

1 概述	2
1.1 运行环境	2
1.2 编译工具	2
1.3 程序文件列表	2
2 主要算法	3
2.1 server部分	3
2.1.1 与client连接通信部分	3
2.1.2 修改server配置部分	3
2.2 client部分	4
2.2.1 功能函数部分	4
2.2.2 图形界面部分	7
3 主要数据结构	8
4 程序测试截图及说明	9
5 遇到的问题及解决办法	20
6 体会与建议	21

1 概述

1.1 运行环境

本实验采用的是64位的Linux操作系统，版本为Ubuntu 15.04。程序实现采用Python语言，使用PyQt4进行用户界面的设计。

1.2 编译工具

实验中程序在Sublime中编写，直接使用Linux的Terminal进行python编译。

1.3 程序文件列表

- server
 - server.py #服务器端主文件
 - account.py #管理用户信息的文件
 - account.pyc
 - account.json
 - modify.py #用于修改账户路径名
 - port_host.py #管理host和port的文件
 - port_host.pyc
 - port_host.json
 - anonymous
 - .. #用户文件
 - Michelle
 - .. #用户文件
 - root
 - .. #用户文件
- client
 - clientGui.py #用户界面版客户端主文件
 - client.py #无用户界面版客户端主文件
 - dialog.py #弹窗设置文件
 - dialog.pyc
 - get_fileProperty.py #获取本地文件调用的文件
 - get_fileProperty.pyc
 - icons
 - .. #用户界面使用的图标
 - client_recv
 - .. #使用无用户界面时接收文件的文件夹

2 主要算法

FTP服务器采取的是c\s结构。算法介绍中就server与client两部分进行介绍。

在前期学习过程中，我知道了python中实现FTP服务器有两种方法，一是使用ftplib模块，一是基于socket编程。由于实验要求，设计采用了第二种方式。

2.1 server部分

server部分分为与客户端连接通信部分与服务器端修改设置部分。在运行server.py时询问进入通信模式还是进入修改配置模式。确认后执行进一步的操作。

2.1.1 与client连接通信部分

与客户端连接通信部分构造了一个MyTCPHandler的类，其中对handle()函数进行了重写，用于多客户端访问服务器端时的处理。当client创建连接后，若server确定连接后，client将要执行的信息传输给server。server对此信息进行解析，确定功能函数。对应的功能函数分别有身份认证函数ftp_authentication()，文件传输函数file_transfer()，删除服务器端文件（夹）函数delete_file()，返回服务器端文件夹列表函数list_file()，在服务器端创建文件夹的函数make_dir()，修改服务器端文件（夹）名函数rename()，进入相应文件夹的函数switch_dir()。对于client传递过来的信息，若server处理成功，则返回成功信息，若处理失败，返回失败信息。

account.py与account.json用于处理FTP服务器中的用户身份信息。server.py通过调用account.py，确认用户的权限。若用户有执行相应操作的权限，则有进一步处理。若无权限，则给client返回错误信息。

2.1.2 修改server配置部分

修改server的配置包括：创建新的用户acc_add()，删除用户acc_del()，修改用户属性attr_change()，显示用户属性show_attr()，修改用户密码pwd.change()，修改端口名port.change()，修改FTP路径名port.change()。

在创建新的账号时，系统会展示默认用户属性，并询问是否修改属性。直至确认不修改，创建用户成功。

删除账号时，系统会询问是否删除用户对应的文件夹。确实删除时server才会删除账户对应的文件夹。

删除账号、显示、修改账号属性时，需要输入账户密码。若密码正确才会执行相应的操作。在显示账号信息时，若是以root账号登录，可以选择是否显示所有账号信息。

在终端执行相应操作使用方法如下图。

```
help      help
show      show account_name
add        add account_name account_pwd
del        del account_name
exit       exit the system
port       port port_number
host       host host_name
pwd        pwd account_name
attr       attr account_name
```

对于账号的管理是通过account.py与account.json。json文件中存储了账号的字典，如下图。server.py通过调用account.py中的函数，从而获取账号信息，也可对json文件中的账户信息进行修改。account.py中使用json模块，利用json.load()获取账号信息，利用json.dump()将修改后的账户信息写进json文件中。

```
"Michelle": {  
    "rename": "NO",  
    "passwd": "public",  
    "mkdir": "YES",  
    "upload": "YES",  
    "download": "YES",  
    "home": "home/mixie/project/myftp/server/Michelle",  
    "delete": "NO"  
},
```

对于端口名和FTP路径名的修改是通过host_port.py和host_port.json而实现的。json文件中存储了含有host和port信息的字典，如下图。server.py通过调用host_port.py中的函数，从而获取host和port信息，也可对json文件中的host和port进行修改。host_port.py中使用json模块，利用json.load()获取账号信息，利用json.dump()将修改后的host和port写进json文件中。

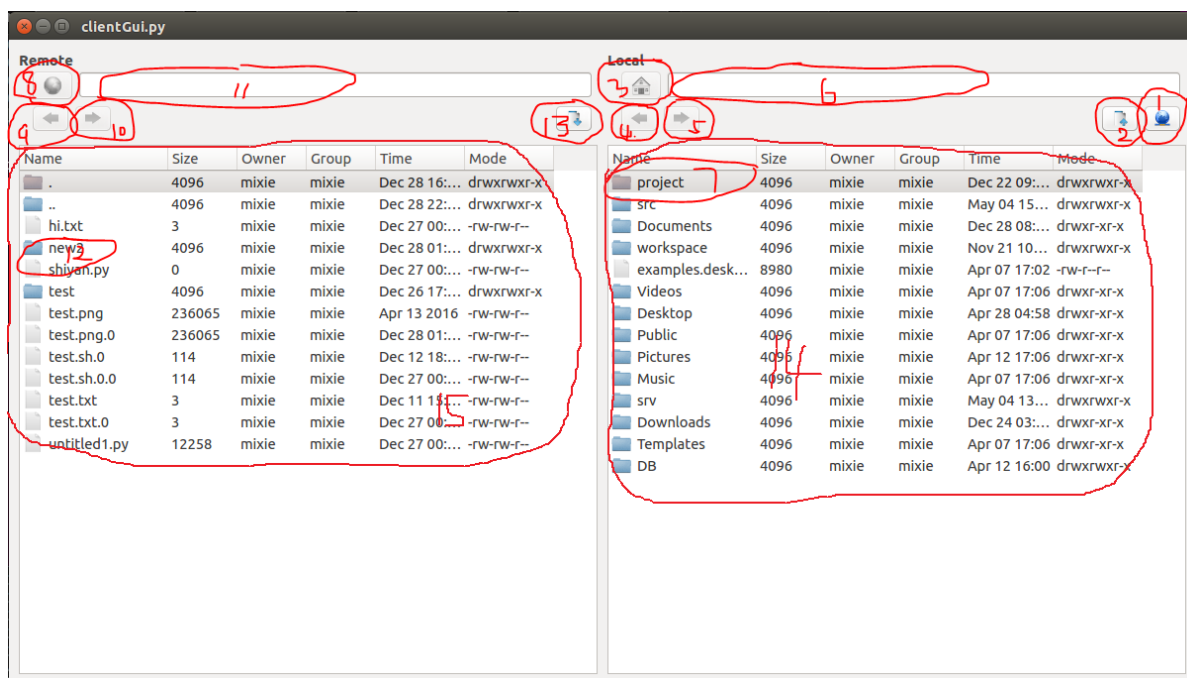
```
{  
    "host": "127.0.0.1",  
    "port": 9000  
}
```

2.2 client部分

对于client部分，我设计了两种实现客户端的方式，一种是有用户界面的，一种则没有。虽然无用户界面的client.py已经完全实现所有的功能，但有用用户界面的clientGui.py是基于client.py的基础而实现的。因此本部分就clientGui.py进行说明。

2.2.1 功能函数部分

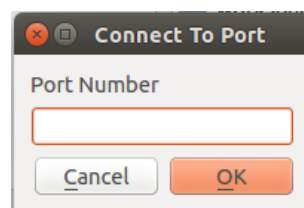
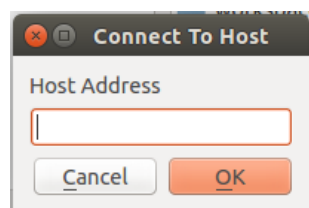
功能函数有连接函数connect()，身份认证函数login()，对于本地和远程的文件列表local与remote有对应的（此处针对remote进行说明，local中对应的函数类似）显示文件列表函数downloadToRemoteFileList()、cdToRemoteDirectory()以及返回主菜单cdToRemoteHomeDirectory()、返回上一菜单cdToRemoteBackDirectory()、返回下一菜单cdToRemoteNextDirectory()、通过输入路径名访问菜单列表cdToRemotePath()等函数，下载服务器中文件的函数download()，将本地文件上传服务器的函数upload()，在服务器端和本地创建文件夹的函数make_dir()与local_make_dir()，删除服务器端文件(夹)的函数delete()，重命名服务器端文件(夹)的函数rename()。



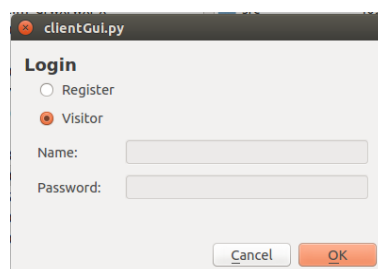
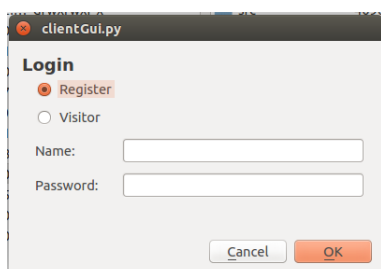
如图所示，先就图中每一部分的功能进行说明。鼠标点击相应button，执行相应的函数。

0.在运行clientGui.py之后，执行函数loadToLocalFileList()，实现本地菜单初始化。显示图中标记14。

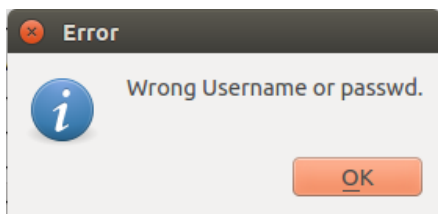
1.点击图中标记1处图标，执行connect()函数，实现与服务器端通过socket进行的连接。



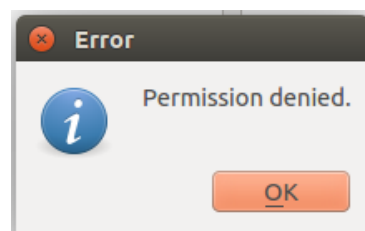
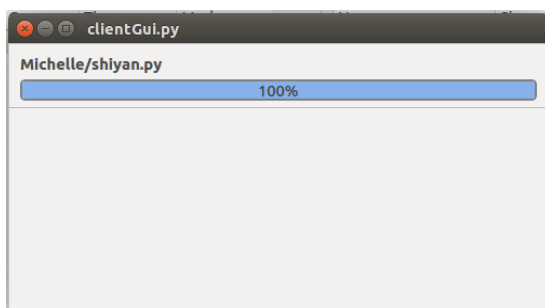
执行connect()函数时，弹出两个文本框，输入host和port后创建连接。连接成功后，执行auth()。此时弹出输入用户名、密码的窗口。



如上图, 用户登录有两种模式, 一是用户名登录, 一是匿名登录。匿名登录时, 用户名与密码默认为"anonymous"与"anonymous", 将信息传输给服务器, 加载匿名登录时对应的远程文件列表。用户名登录后, 在服务器端验证账户信息, 若验证成功, 则进行远程文件列表初始化, 执行downloadToRemoteFileList(), 显示图中标记15。若验证失败, 则弹出如下警告窗, 无初始化信息。



2. 在本地文件列表处选择指定文件夹中要上传的文件, 点击图中标记2处图标, 执行upload()函数, 若用户有上传的权限, 上传文件到指定文件夹, 显示上传进程, 如下图左。若无权限, 则弹出警告窗, 如下图右, 文件上传失败。需要注意的是, 上传只可以上传文件, 不可上传文件夹。



3. 进入本地的某个文件夹中, 点击图中标记3处图标, 执行cdToLocalHomeDirectory(), 返回主文件列表, 调用函数updateLocalFileList(), 刷新菜单栏。进入远程文件列表的某个文件夹中, 点击图中标记8处图标, 执行cdToRemoteHomeDirectory(), 返回主文件列表, 调用函数updateRemoteFileList(), 刷新文件列表。

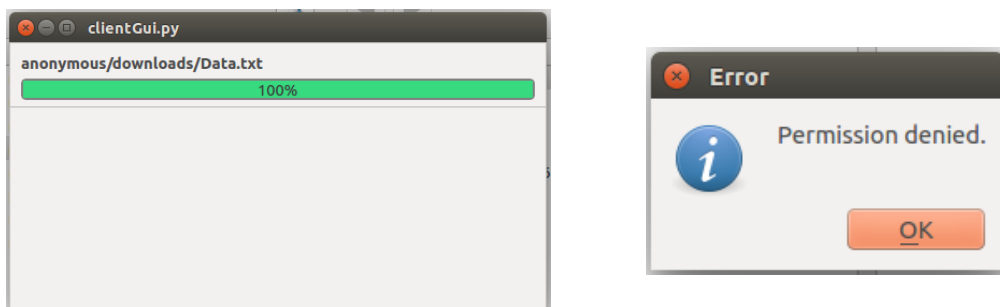
4. 进入本地的某个文件夹中, 点击图中标记4处图标, 执行cdToLocalBackDirectory(), 返回上一级文件列表。进入远程文件列表的某个文件夹中, 点击图中标记9处图标, 执行cdToRemoteBackDirectory()。

5. 返回本地的某个文件夹中, 点击图中标记5处图标, 执行cdToLocalNextDirectory(), 进入前一级文件列表。返回远程文件列表的某个文件夹中, 点击图中标记10处图标, 执行cdToRemoteNextDirectory()。

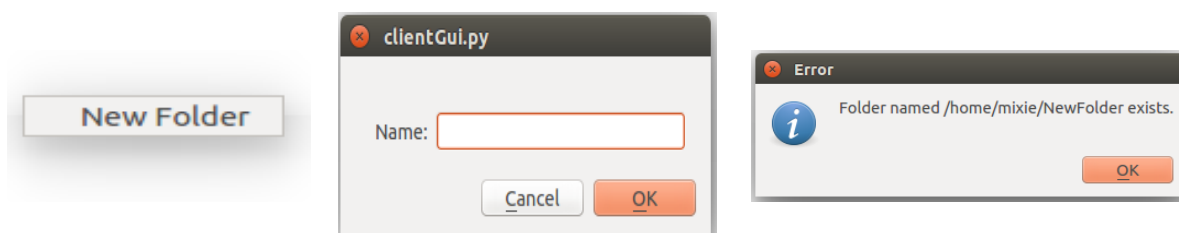
6. 文本输入框中(图中标记6处或11处)中输入文件夹名称, 执行cdToLocalPath()或cdToRemotePath()进入对应的文件夹中。

7. 单击图中标记7处文件夹图标, 执行函数cdToLocalDirectory(), 进入指定文件夹; 单击图中标记12处文件夹图标, 执行函数cdToRemoteDirectory(), 进入指定文件夹。

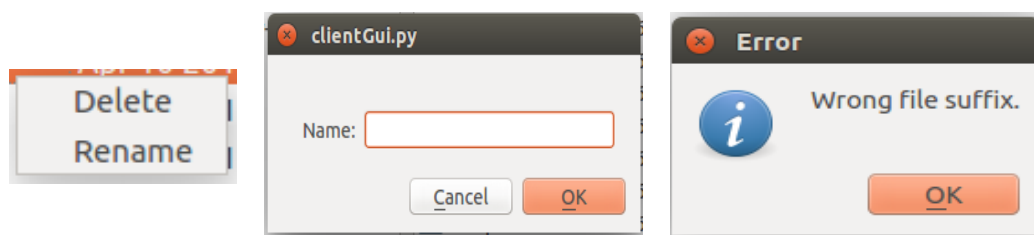
8. 在远程文件列表处选择指定文件夹中要下载的文件, 点击图中标记13处图标, 执行download()函数, 若用户有下载的权限, 下载文件到指定文件夹, 显示下载进程, 如下图左。若无权限, 则弹出警告窗, 如下图右, 文件下载失败。需要注意的是, 下载只可以上下载文件, 不可下载文件夹。



9.在本地文件列表或远程文件列表空白处右键，执行RightBlankMenuWidget()，显示菜单栏，如下图所示左。单击New Folder，执行local_make_dir()或make_dir()，弹窗输入新文件夹名称，如下图所示中。若未输入文件夹名，默认文件夹名为”NewFolder”。若文件夹已存在，弹出如下图所示右窗口。在远程文件列表中，若用户有权限，则创建相应文件夹，刷新文件列表栏，否则弹出警告窗口，服务器端不做修改。而在本地直接创建文件夹，刷新文件列表栏。可将文件下载到新文件夹中。



10.在远程文件列表处选中某一文件夹或者文件，鼠标右键，执行RightMenuWidget()，显示菜单栏，如下图所示左。若单击Delete，执行delete()，若用户有权限，可删除选定的文件夹或文件，若无权限，弹出警告窗口，服务器端不对其做修改。若单击Rename，执行rename()，若用户有权限，可对选定的文件夹或文件重命名，弹出下图中窗口。若命名格式与原文件格式不对，弹出下图右窗口，服务器端拒绝修改。修改好后远程文件列表刷新。



而get_fileProperty.py中，则通过调用os、pwd、stat等模块获取本地当前目录内容，以用于用户界面本地文件列表的加载。

2.2.2 图形界面部分

图形化界面设计主要在clientGui.py、dialog.py两个文件中。其中clientGui.py设计主界面，即菜单栏部分。dialog.py设计弹窗部分，包括输入用户信息、上传下载弹窗、显示进程部分。

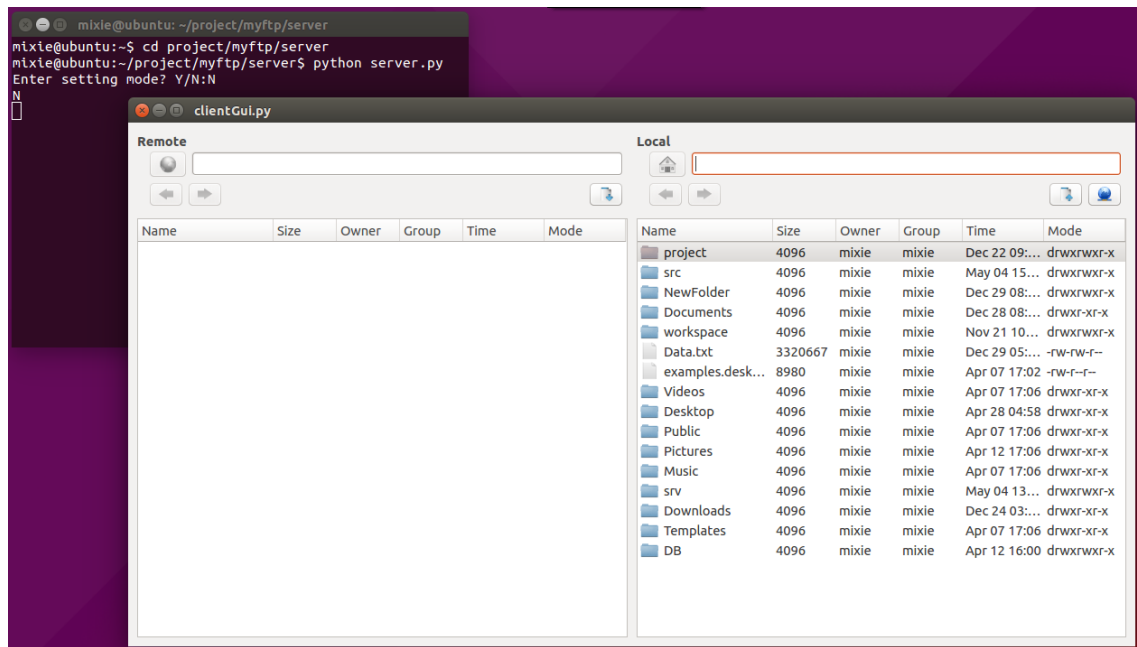
3 主要数据结构

在本程序中，由于需要存储的信息小，主要涉及到账户信息与host、port信息的存储。因而未采用数据库，而是将信息以字典的形式，存储到相应的json文件中。并通过json模块的json.load()与json.dump()函数对数据进行读取与修改。存储结构图如下。

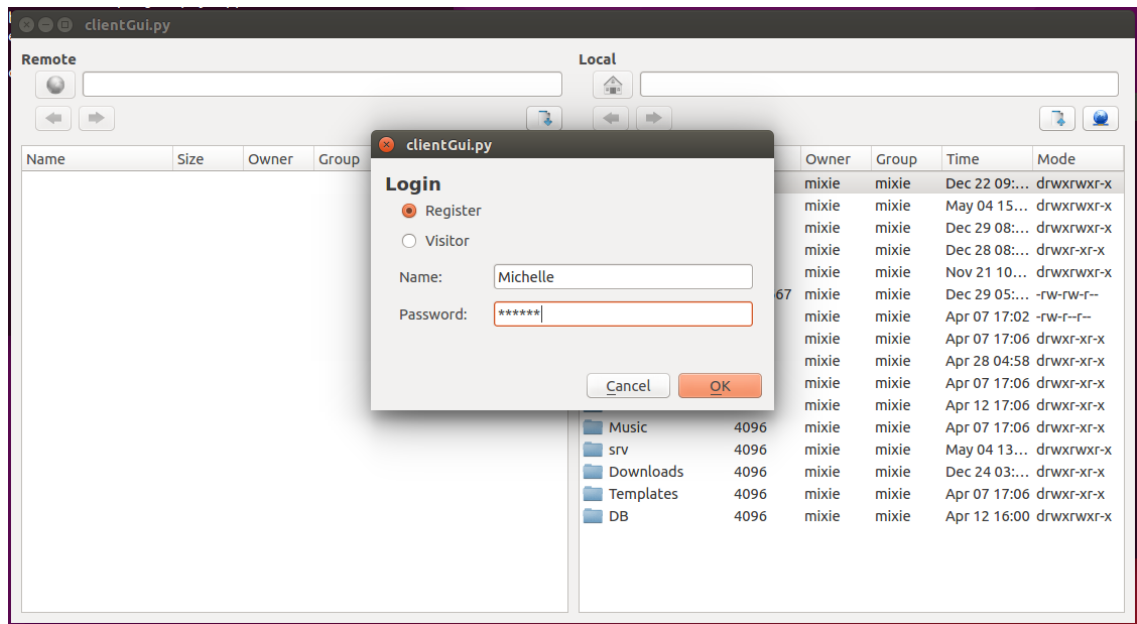
```
{
  "Michelle": {
    "rename": "NO",
    "passwd": "public",
    "mkdir": "YES",
    "upload": "YES",
    "download": "YES",
    "home": "home/mixie/project/myftp/server/Michelle",
    "delete": "NO"
  },
  "root": {
    "rename": "YES",
    "passwd": "root",
    "mkdir": "YES",
    "upload": "YES",
    "download": "YES",
    "home": "home/mixie/project/myftp/server/root",
    "delete": "YES"
  },
  "anonymous": {
    "rename": "NO",
    "passwd": "anonymous",
    "mkdir": "NO",
    "upload": "NO",
    "download": "YES",
    "home": "home/mixie/project/myftp/server/anonymous",
    "delete": "NO"
  },
}
```

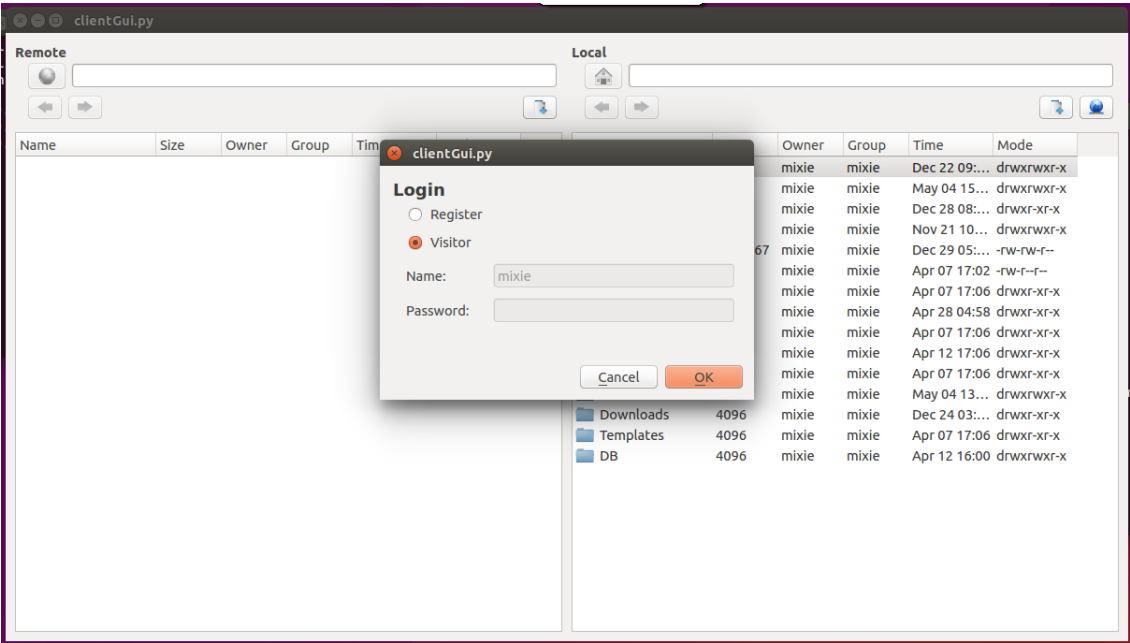
4 程序测试截图及说明

1.服务器端选择通信模式，执行clientGui.py。用户界面初始化。

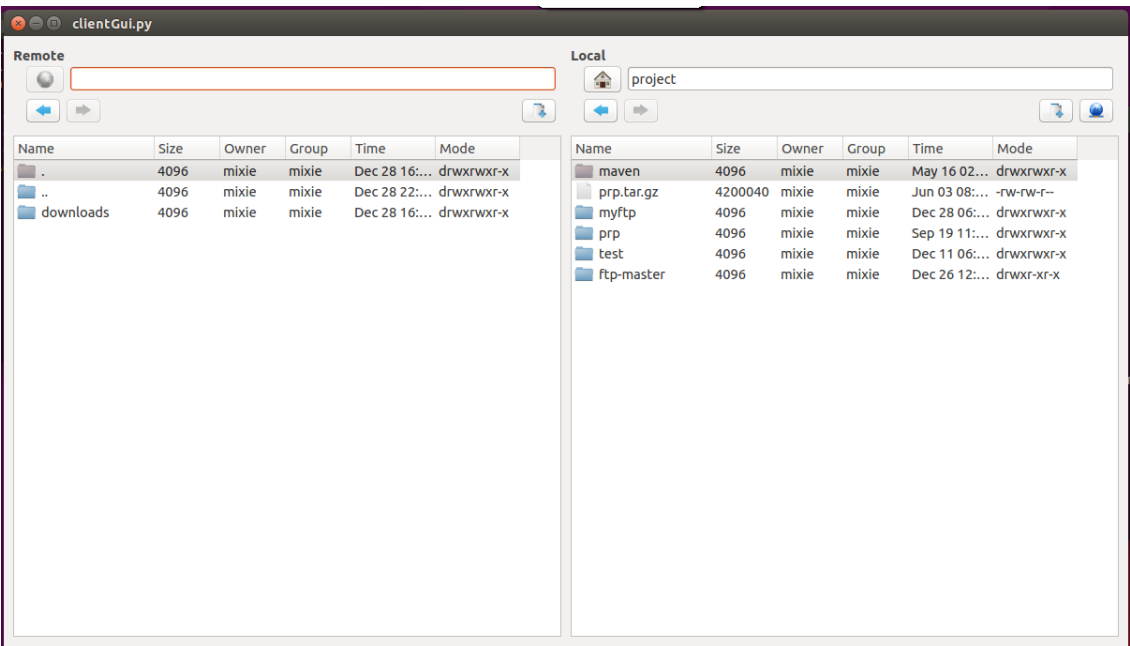


2.点击连接图标，输入FTP路径名和端口号后，弹出登录窗口。可以使用账户名登录，也可使用匿名登录。

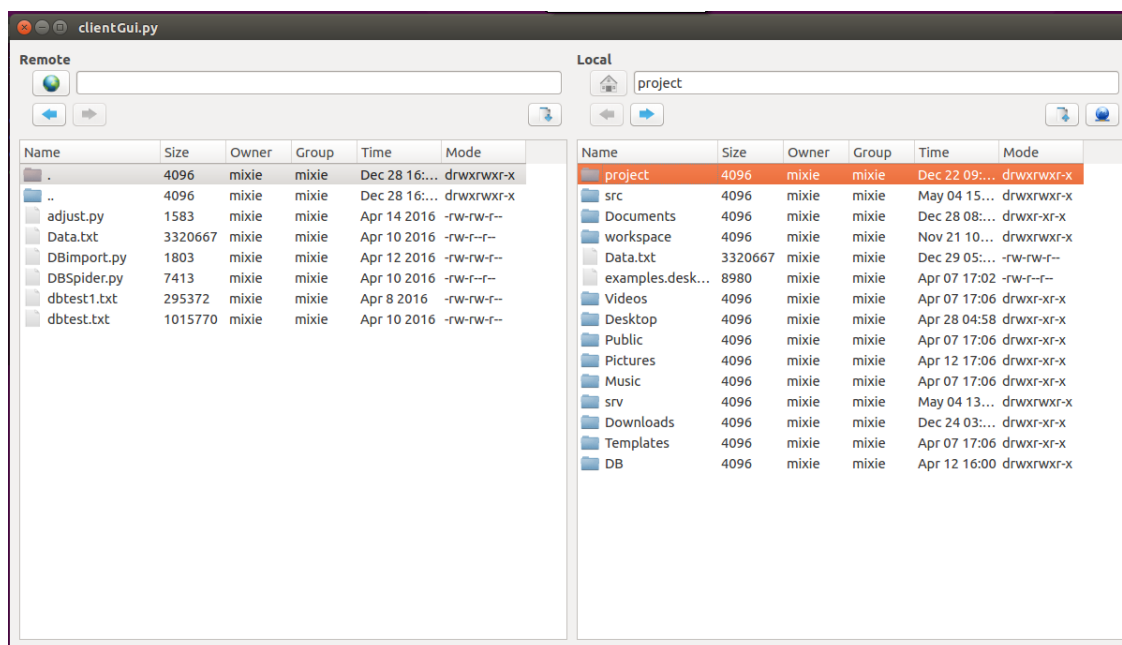




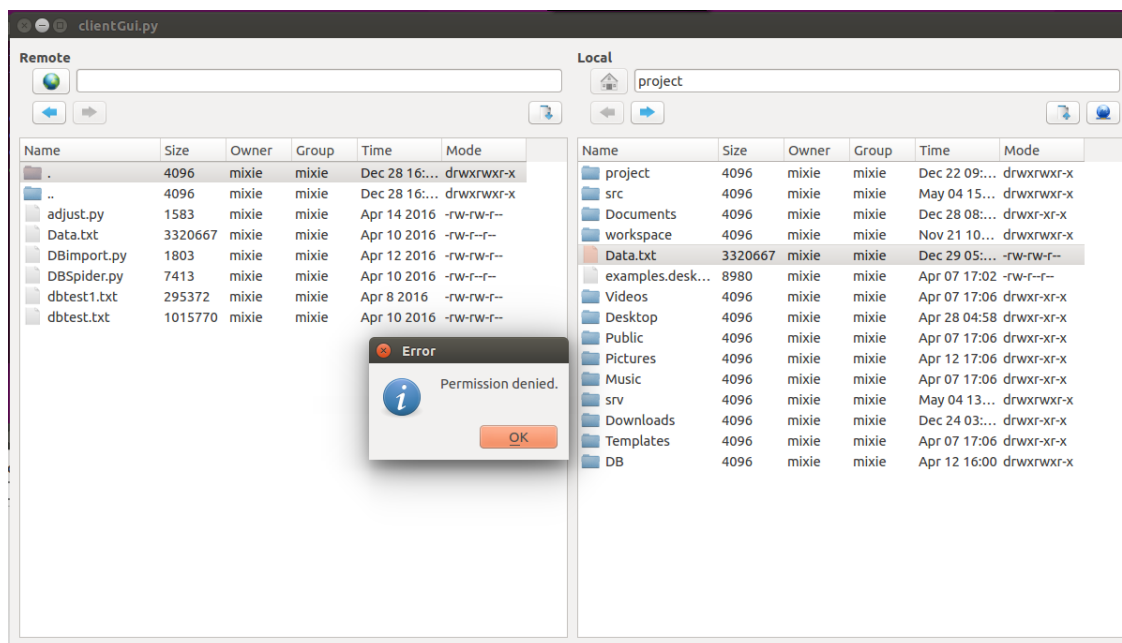
3.匿名登录后，远程文件夹列表中加载对应的文件列表。在本地输入框中输入文件夹名称，进入相应的文件夹中。此时返回上一级文件列表的箭头激活，可点击该箭头返回前一级文件列表中。



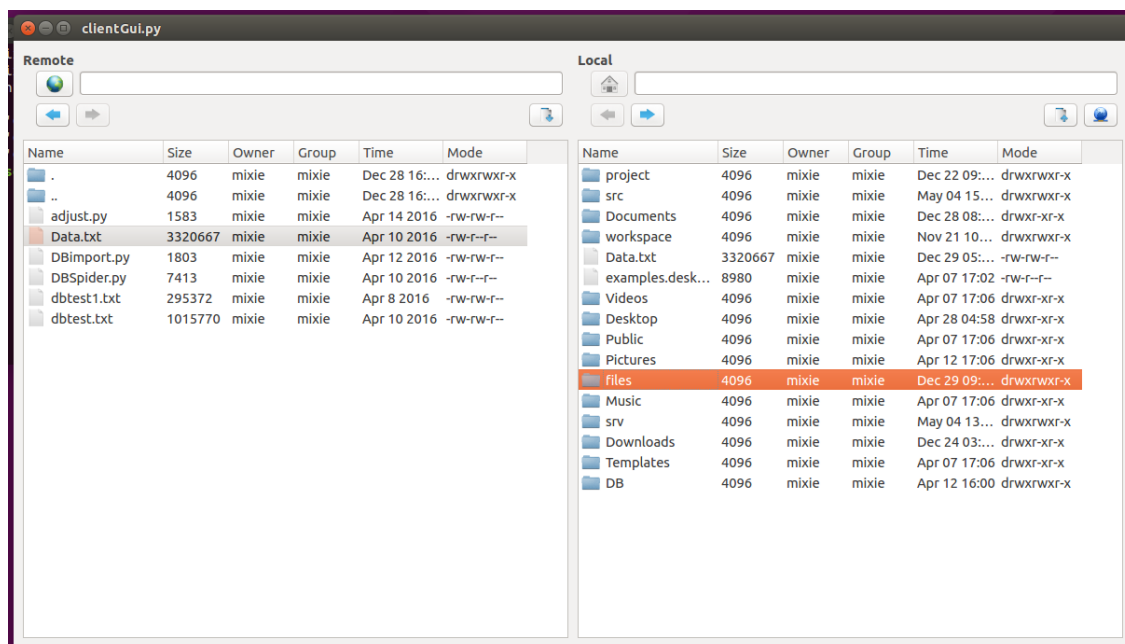
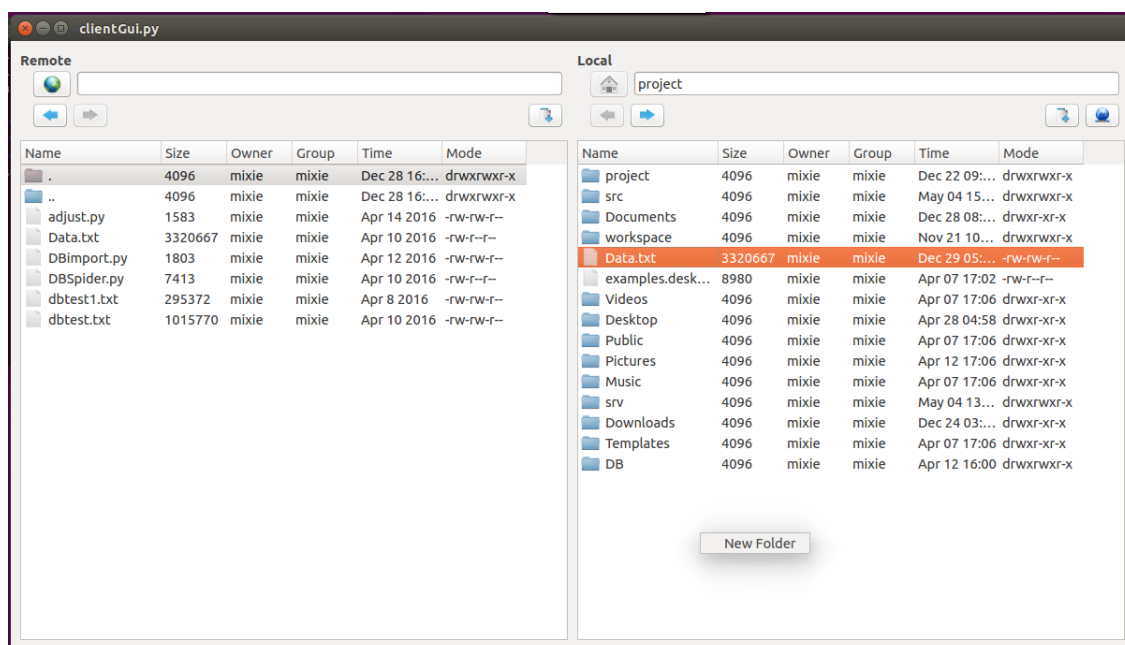
4. 双击文件夹名称，进入该文件夹中，如下图远程文件列表。返回上一级文件列表后，进入后一级文件列表的箭头激活，可以点击该箭头进入后以及文件列表中。如下图本地文件列表。

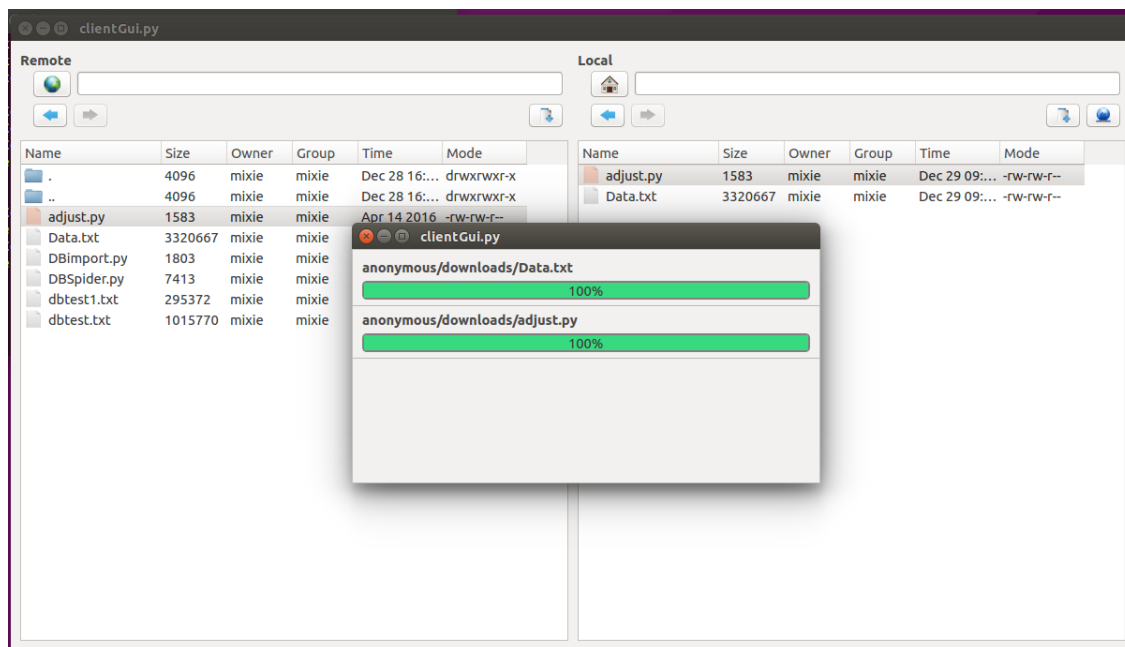


5. 由于匿名登录账户只有下载文件的权限，没有上传文件、修改远程文件（夹）名、删除远程文件（夹）、在远程端创建文件夹的权限，因而提示操作被禁止。

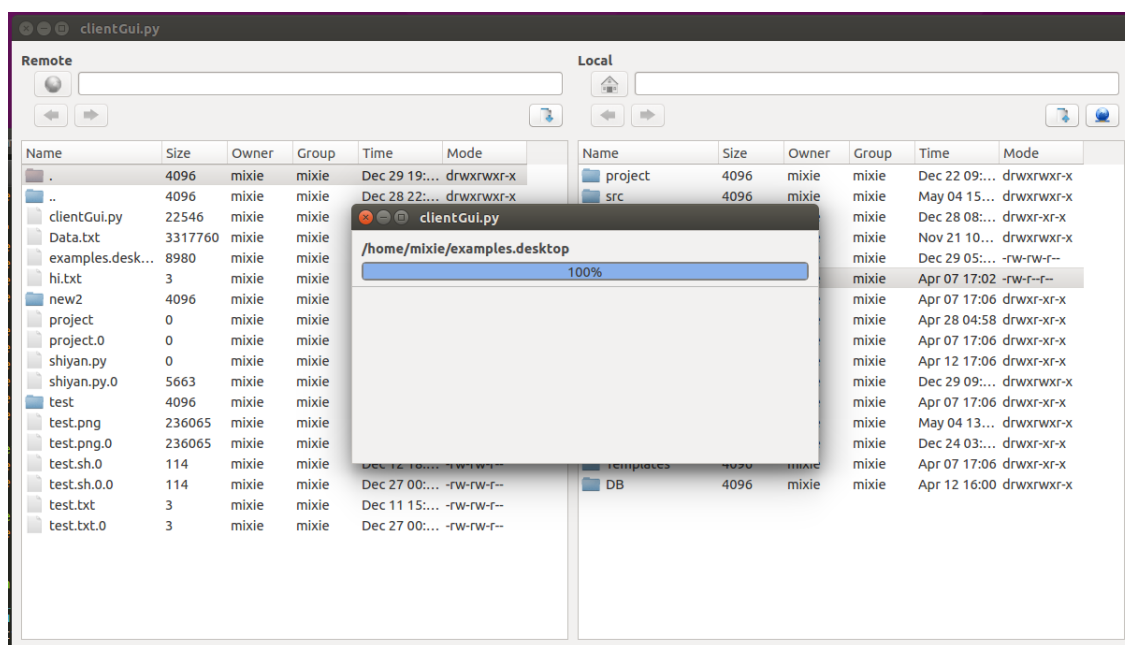


6.在本地创建新文件夹files，从FTP服务器中下载文件adjust.py与Data.txt。

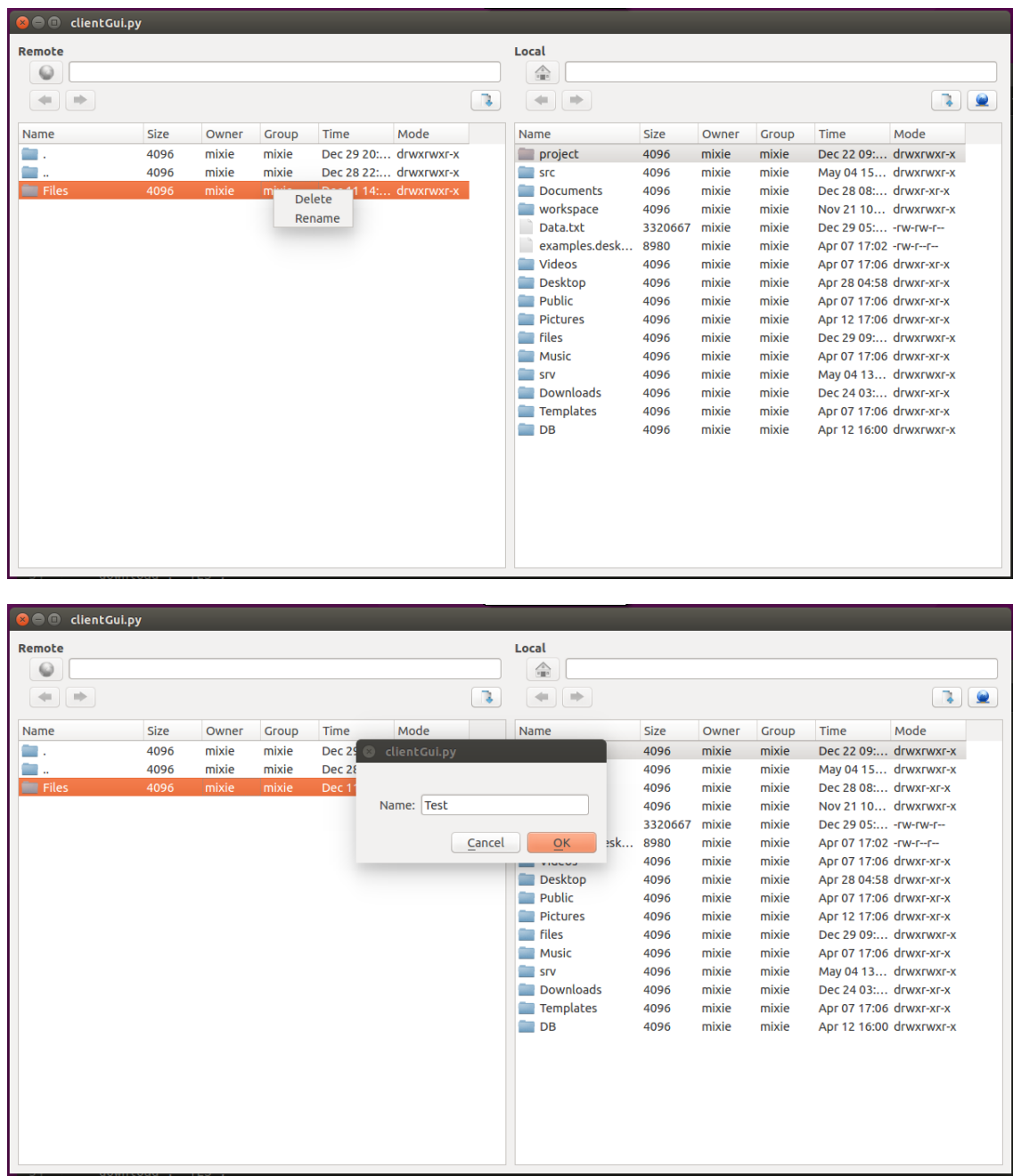


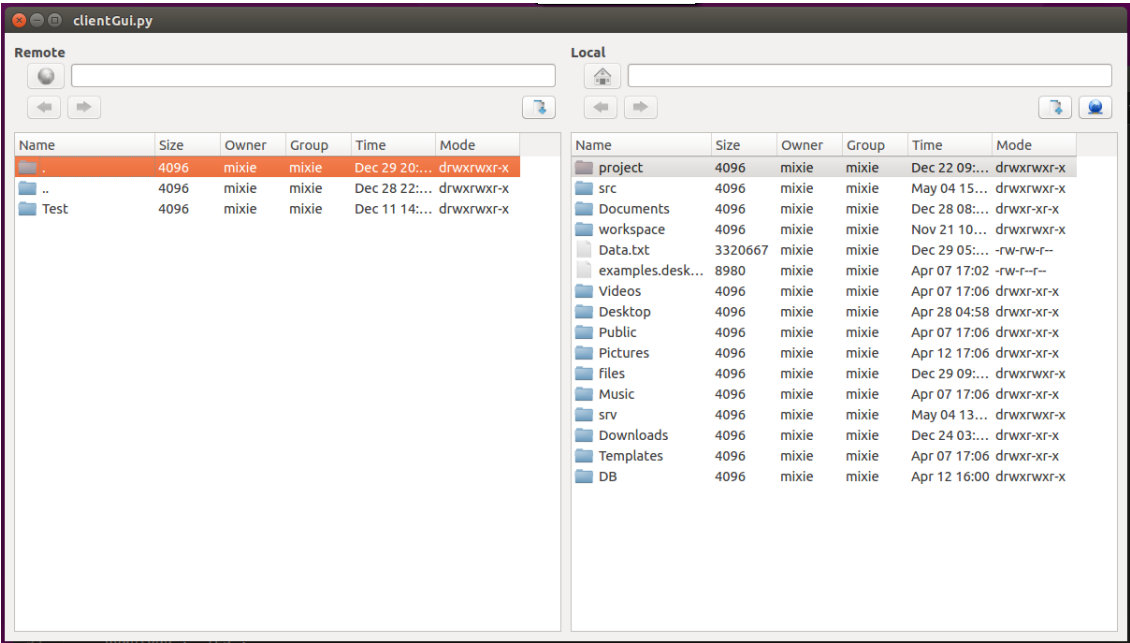


7.通过账号Michelle登录，进入对应的FTP服务器中。选择上传的文件进行上传。

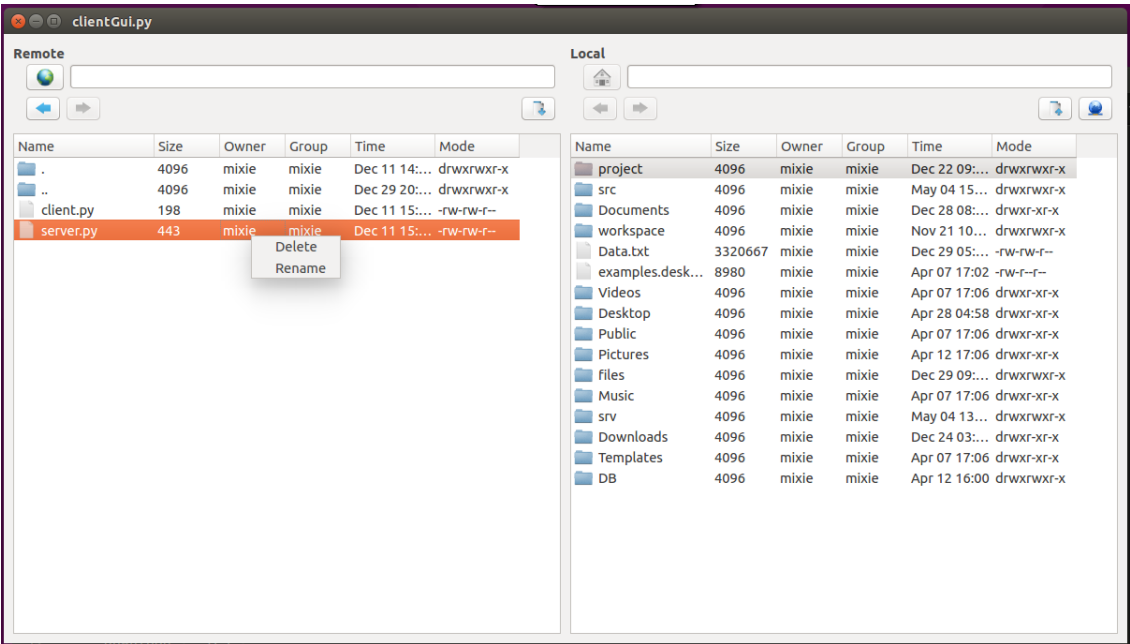


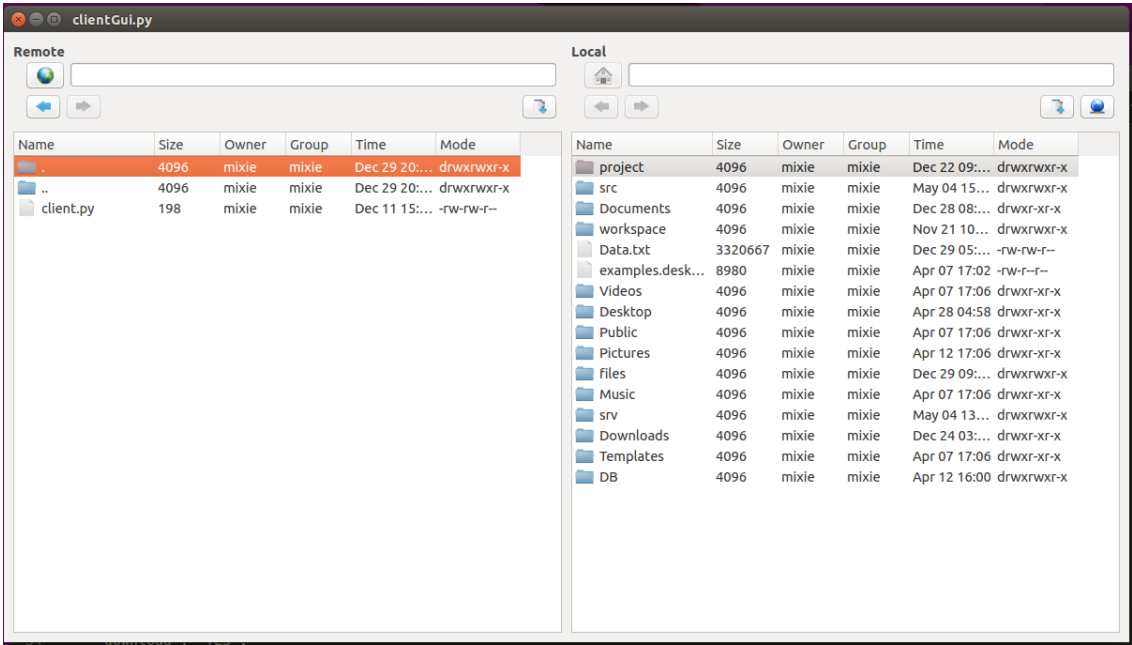
8.通过账号root登录，修改文件夹Files的名字为Test。



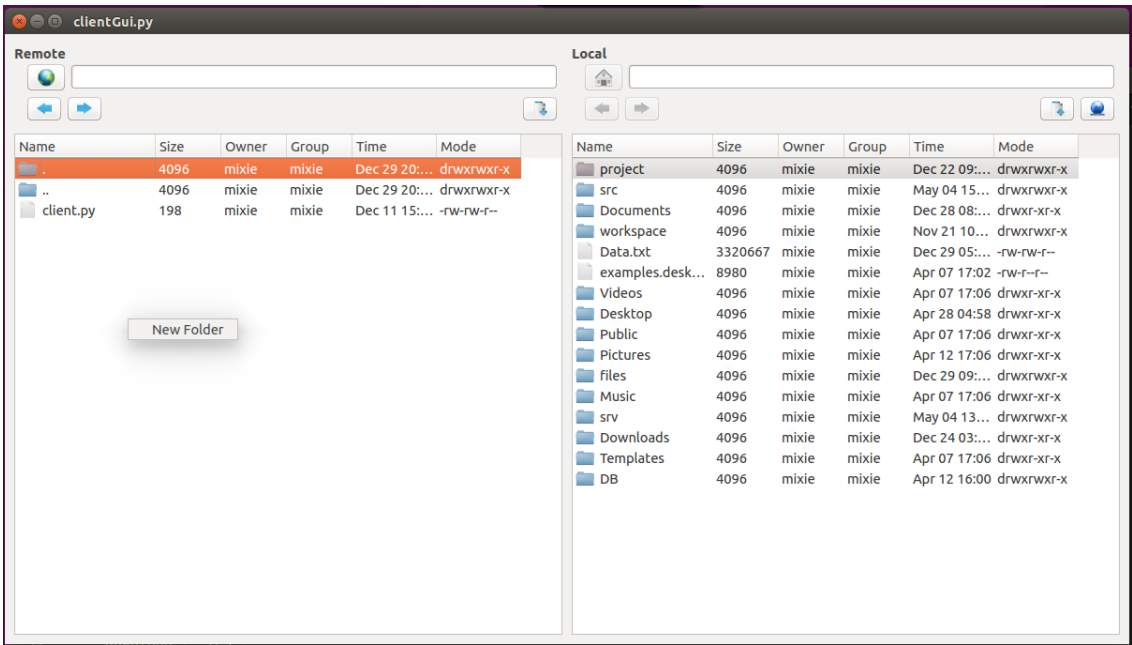


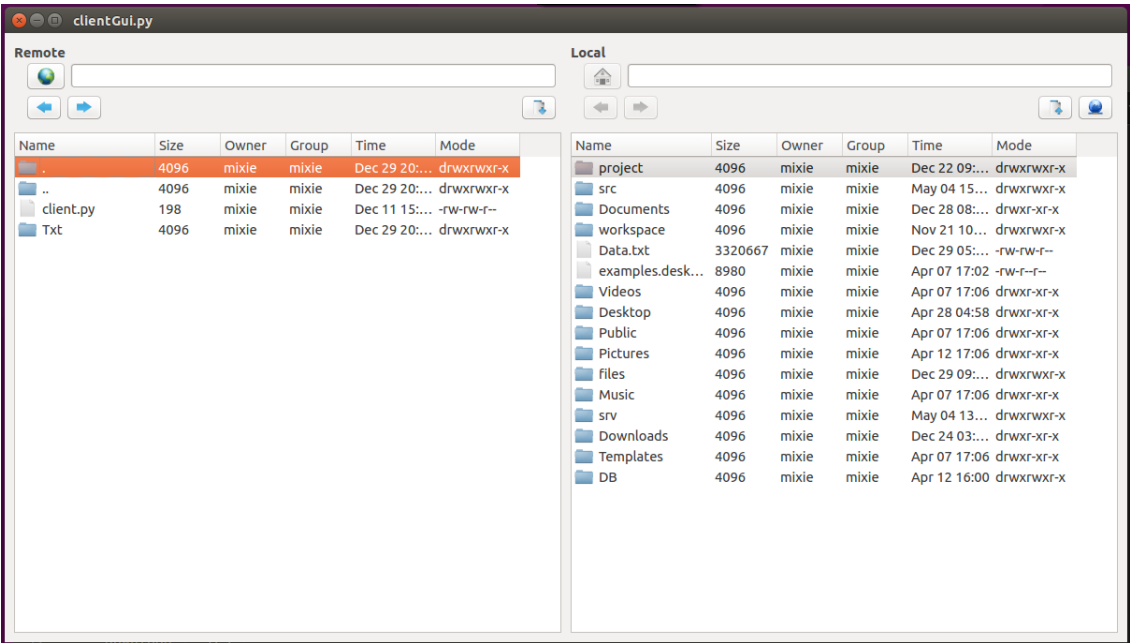
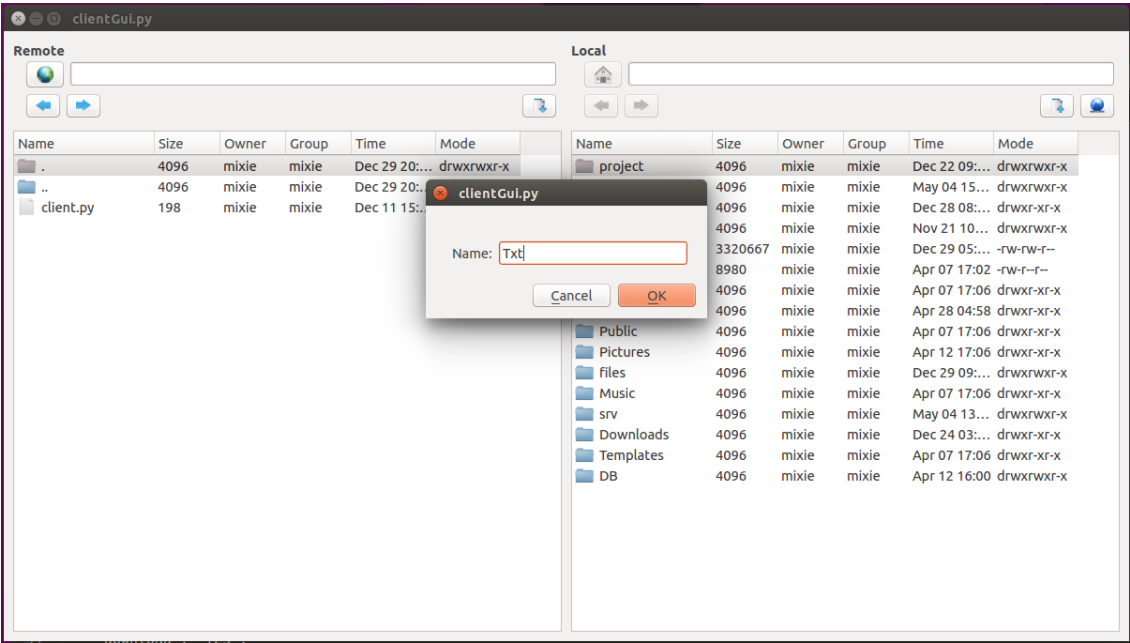
9.删除服务器端文件server.py。





10.在服务器端创建新的文件夹Txt。





11.在服务器端修改相应配置。键入help获得帮助信息。键入show root显示root账号的属性。键入attr Michelle修改该账号的属性。输入错误，返回错误信息，拒绝修改。

```

Enter setting mode? Y/N:Y
Y
>>:help
      help      help
      show      show account_name
      add      add account_name account_pwd
      del      del account_name
      exit      exit the system
      port      port port_number
      host      host host_name
      pwd      pwd account_name
      attr      attr account_name

>>:show root
Password:
show all users' information? Y/N: N

      Details information about root is:
      home      home/mixie/project/myftp/server/root
      passwd     root
      Authentication for
      rename     YES
      mkdir      YES
      upload     YES
      download   YES
      delete     YES

>>:attr Michelle
Password:
Enter the attribute(s) name that you want to change: n
Invalid attribute name n. Thus no change to it.

      Details information about Michelle is:
      home      home/mixie/project/myftp/server/Michelle
      Authentication for
      rename     NO
      mkdir      YES
      upload     YES
      download   YES
      delete     NO

```

12.删除账号Mark信息，确认删除其文件夹。显示账号Michelle属性，修改其rename的属性。修改其密码。

```

>>:del Mark
Password:
Delete directory of Mark? Y/N: Y
Directory:home/mixie/project/myftp/server/Mark has been deleted.
Account:Mark has been deleted.
>>:show Michelle
Password:

      Details information about Michelle is:
      home      home/mixie/project/myftp/server/Michelle
      passwd     public
      Authentication for
      rename     YES
      mkdir      YES
      upload     YES
      download   YES
      delete     NO

>>:attr Michelle
Password:
Enter the attribute(s) name that you want to change: rename
Value of attribute rename is YES. Change it into YES/NO: NO

      Details information about Michelle is:
      home      home/mixie/project/myftp/server/Michelle
      Authentication for
      rename     NO
      mkdir      YES
      upload     YES
      download   YES
      delete     NO

>>:pwd Michelle
Password:
Enter new password: public
Account:Michelle passwd has been changed.
>>:

```

13.增加账户Mark，在增加的过程中修改其属性。修改端口号。修改FTP路径号。

```
>>:add Mark public

Details information about Mark is:
home      home/mixie/project/myftp/server/Mark
passwd    public
Authentication for
rename    YES
mkdir     YES
upload    YES
download  YES
delete    YES

Confirm the information above about Mark? Y/N: N
Enter the attribute(s) name that you want to change: rename
Value of attribute rename is YES. Change it into YES/NO: NO

Details information about Mark is:
home      home/mixie/project/myftp/server/Mark
passwd    public
Authentication for
rename    NO
mkdir     YES
upload    YES
download  YES
delete    YES

Confirm the information above about Mark? Y/N: Y
Account:Mark has been added!
>>:port 80
Invalid ftp port number. 1024<port_number<65536.
>>:port 8000
Port number has been changed to 8000
>>:host 1
Invalid host name
>>:host 192.168.232.132
Host name has been changed to 192.168.232.132
```

5 遇到的问题及解决办法

1.最开始时不知道如何通过客户端修改服务器端中的文件、文件夹信息，在网上搜索无果，询问学姐后得知可以通过import commands模块，将修改信息传输到服务器端，调用相关的函数进行修改。

2.在扩充的功能里面，涉及到对账户信息的修改。一开始account.py中只有一个包含了账户信息的字典。server.py中import account后可以获取账户信息，但无法修改。因此一开始想用关系型数据库，但是由于数据量小，用数据库效率不高。和室友讨论后，决定使用json文件，以account.py为中间载体，在server.py中修改账户信息，通过account.py对account.json进行修改。从而达到修改账户信息的功能。

3.在实现用户界面时，由于Python本身的Tkinter设计丑陋，因而采用PyQt4进行设计。但是PyQt的教程较少，在设计的过程中遇到了很多问题。比如设计远程文件列表中的右键菜单栏时，想设计在文件处右键和在空白处右键得到的菜单栏不同，最开始时无法实现这种功能。在研究了一些C++中Qt的实现代码后，有些使用itemAt()函数判断鼠标右键时指针是否在item处来判断，因而将其移植到Python中，从而实现了目的。

4.修改文件（夹）名称、创建新的文件夹时，一般都是直接在对应的图标下方对名字进行修改。但是PyQt资料少，也找不到可以参考的代码，因此我选择了弹窗来输入文件名，实现了该功能。

5.在整个代码的设计过程中，我遇到了大大小小的问题，这些问题都是在上网查阅相关资料、求助学姐学长同学后得以解决。也才实现了最终的目的。

6 体会与建议

通过本次FTP服务器的设计，我对基于TCP/IP的socket编程有了很深刻的了解，接触了基于Python的图形化界面设计工具PyQt，对Python也有了更多的理解。本次设计周期是两个月，从一开始茫然，到后来有了思路，再一点点的设计，每天完善一点功能，才有了最终呈现的样子。虽然过程有些艰难，中间也遇到了各种各样的bug，但在实现之后又不免高兴激动一番。