

# ЛР 2. FTP

автор: Палатова Т.А.

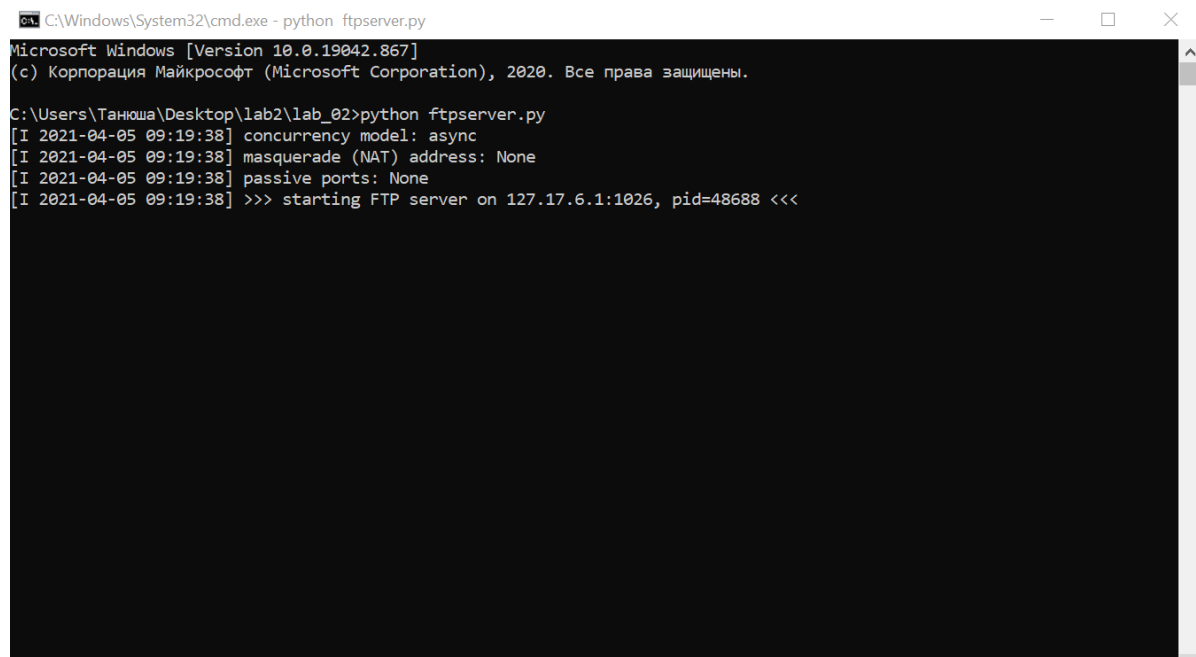
url: [https://github.com/palatova/Wide\\_Nets/tree/main/lab\\_2\\_ftpserver](https://github.com/palatova/Wide_Nets/tree/main/lab_2_ftpserver)

## Задание

1. Реализованный функционал должен полностью обеспечивать следующий сценарий работы с FTP-сервером (в пассивном и активном режиме):
  1. Аутентификация на сервере (использование не только анонимного доступа скачивание файлов, но и доступ авторизованных пользователей и загрузка файлов);
  2. Выбор директории и скачивание файла/ов (возможно из различных директорий);
  3. Выбор директории/создание директории и загрузка файлов (если на это есть права)
  4. Завершение сессии

## Результаты

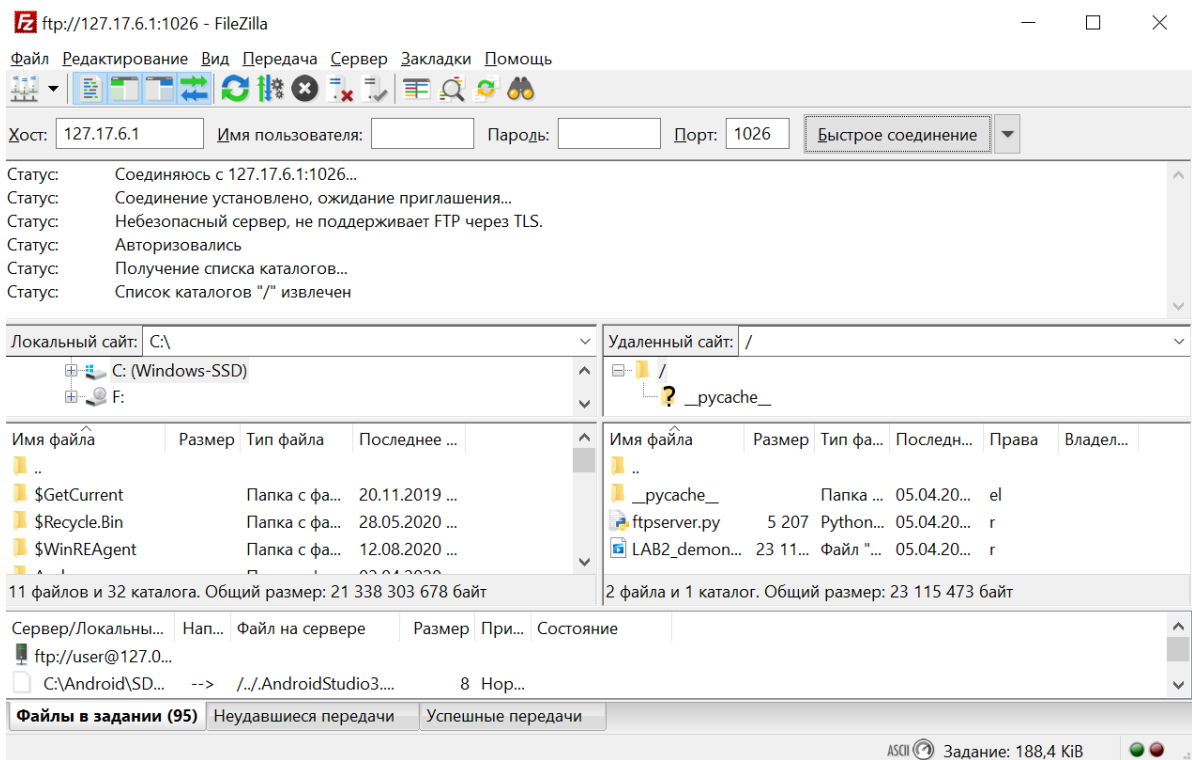
Загружаем сервер:



```
C:\Windows\System32\cmd.exe - python ftpserver.py
Microsoft Windows [Version 10.0.19042.867]
(c) Корпорация Майкрософт (Microsoft Corporation), 2020. Все права защищены.

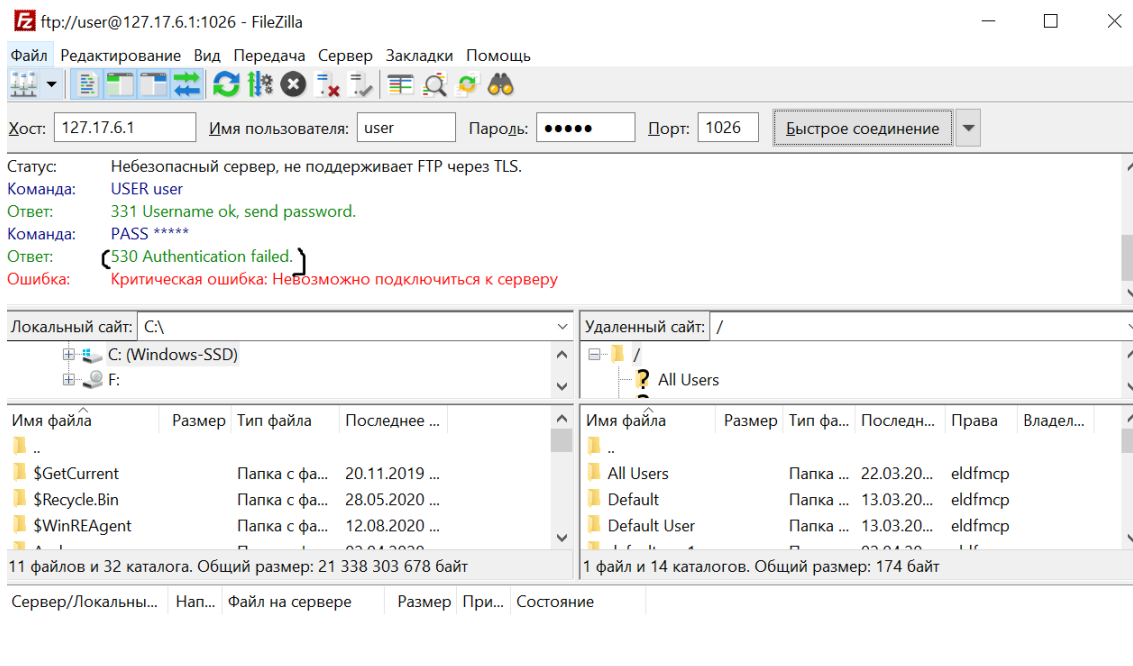
C:\Users\Танюша\Desktop\lab2\lab_02>python ftpserver.py
[I 2021-04-05 09:19:38] concurrency model: async
[I 2021-04-05 09:19:38] masquerade (NAT) address: None
[I 2021-04-05 09:19:38] passive ports: None
[I 2021-04-05 09:19:38] >>> starting FTP server on 127.17.6.1:1026, pid=48688 <<<
```

Аутентификация на сервере анонимного пользователя:

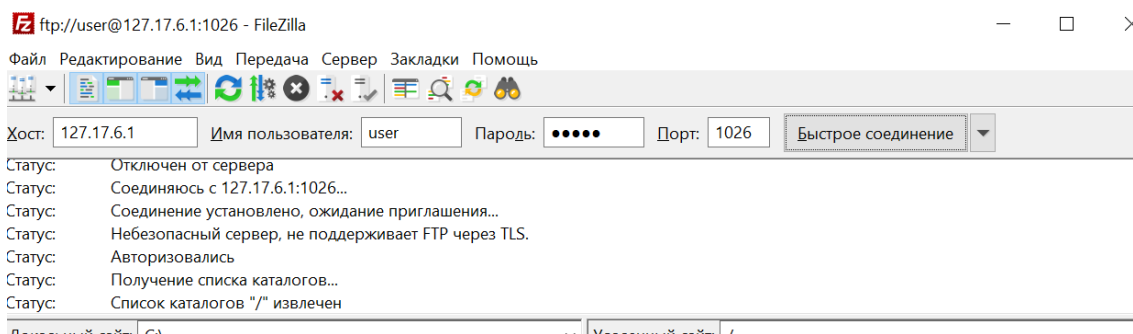


Аутентификация на сервере пользователя - user :

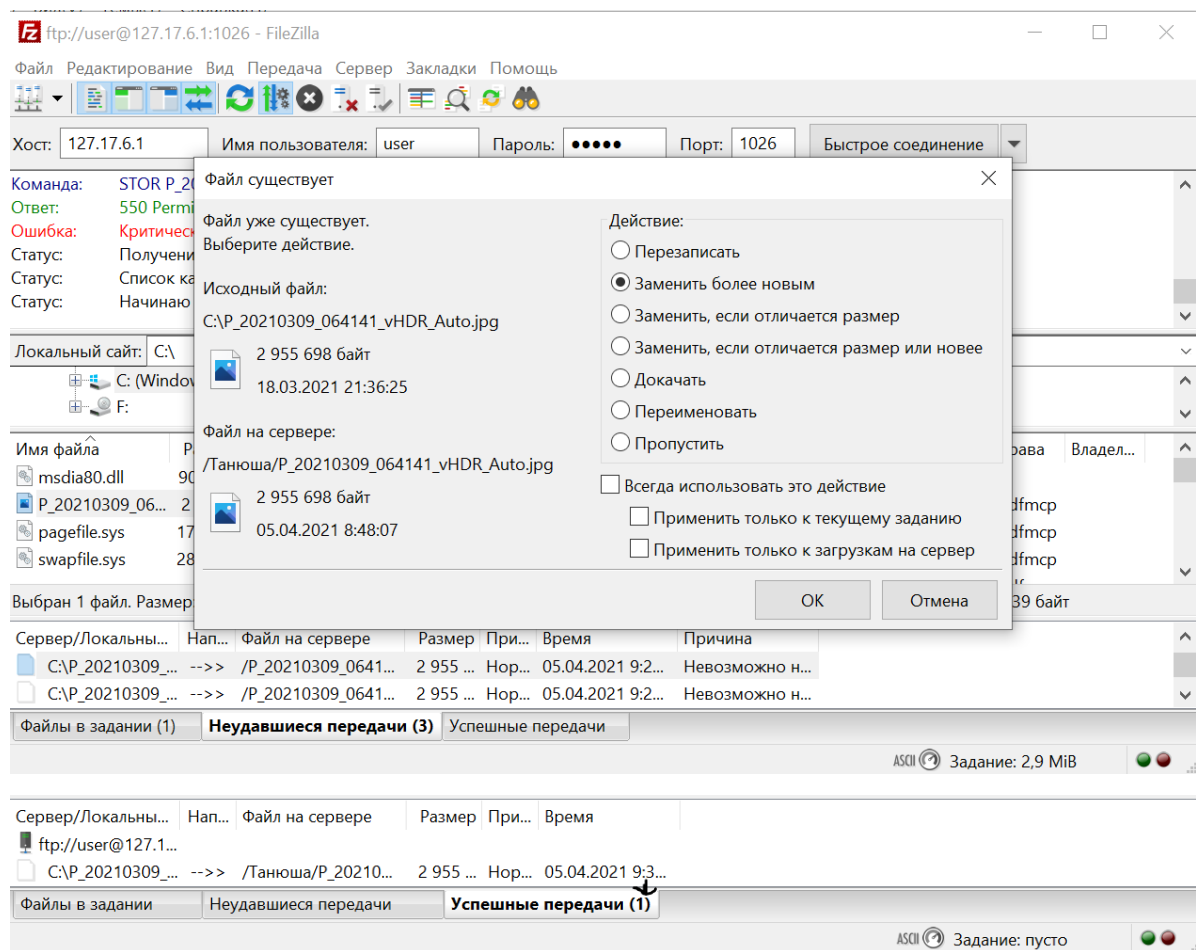
### 1. Обработка ошибки неправильная пара логин - пароль:



### 2. Успешная авторизация



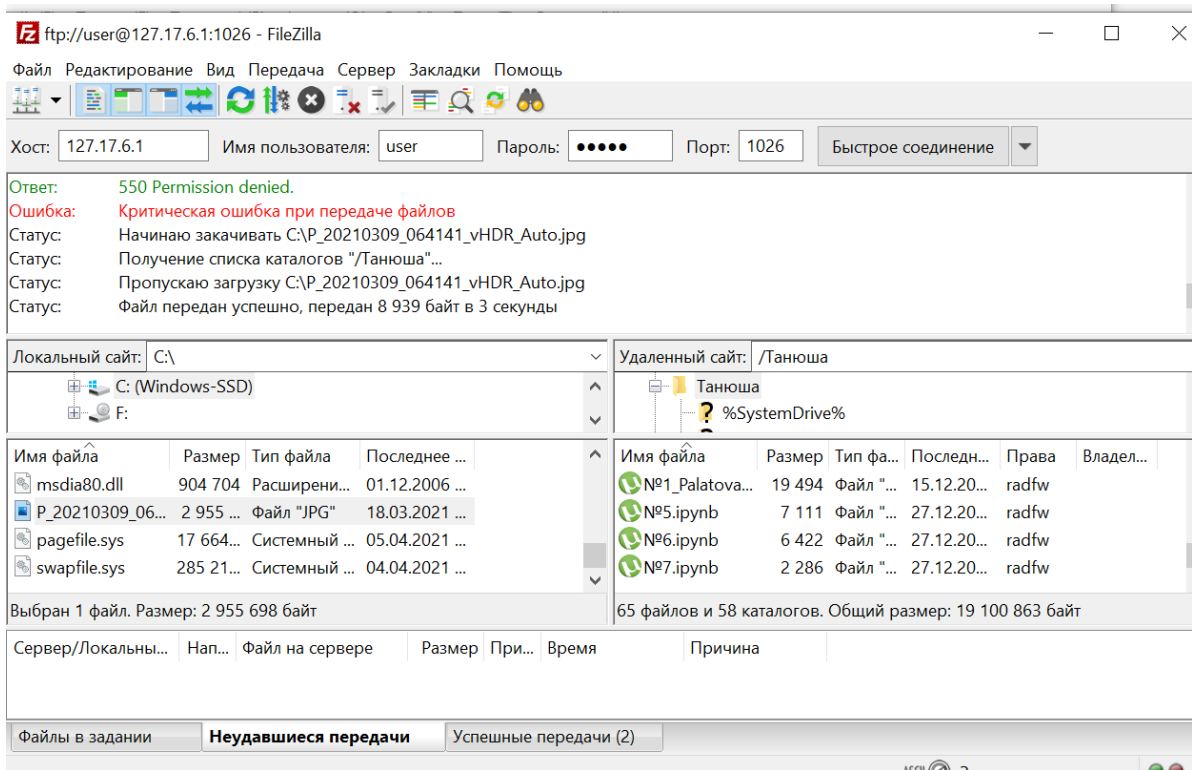
Выбор директории и скачивание файлов:



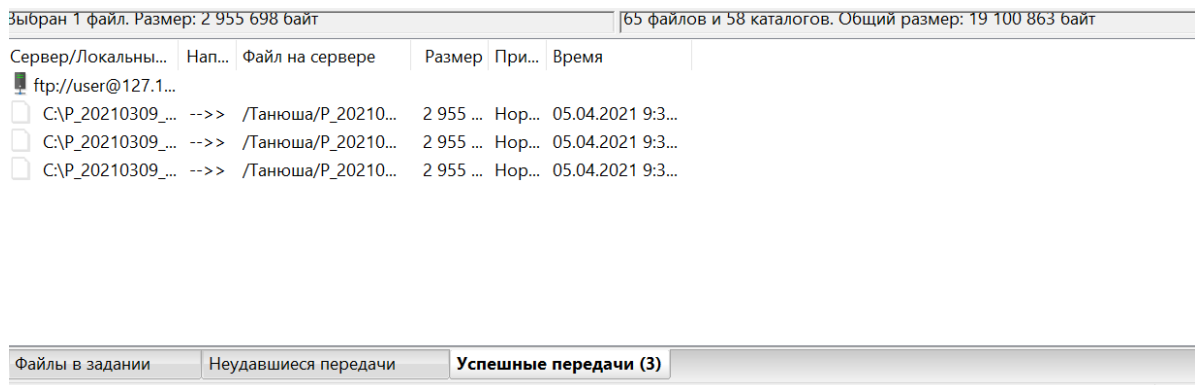
Обработка ошибки неправильное имя файла:

Команда: RETR C:/pagefile.sys1  
 Ответ: 550 Invalid argument.

Выбор директории директории и загрузка файлов:

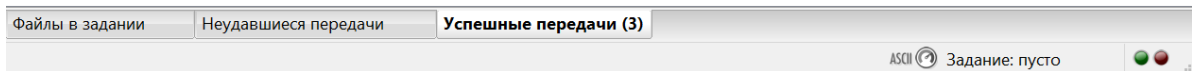
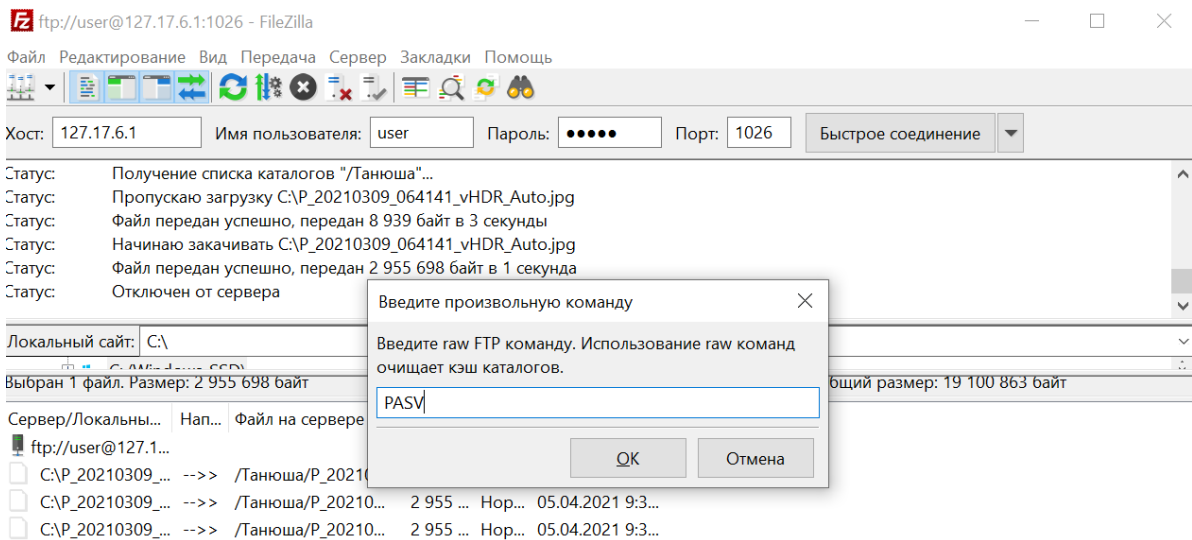


Передача файла с локального сайта на удаленный прошла успешно.

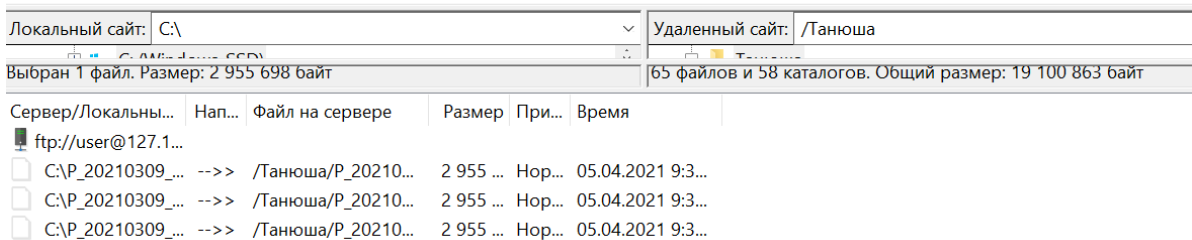


\

Пассивный режим:



Команда: PASV  
 Ответ: 227 Entering passive mode (127,17,6,1,230,73).



Выбор директории директории и загрузка файлов в пассивном режиме:

Статус: Получение списка каталогов "/Танюша/.config"..."  
 Статус: Файл передан успешно, передан 800 774 байт в 1 секунда

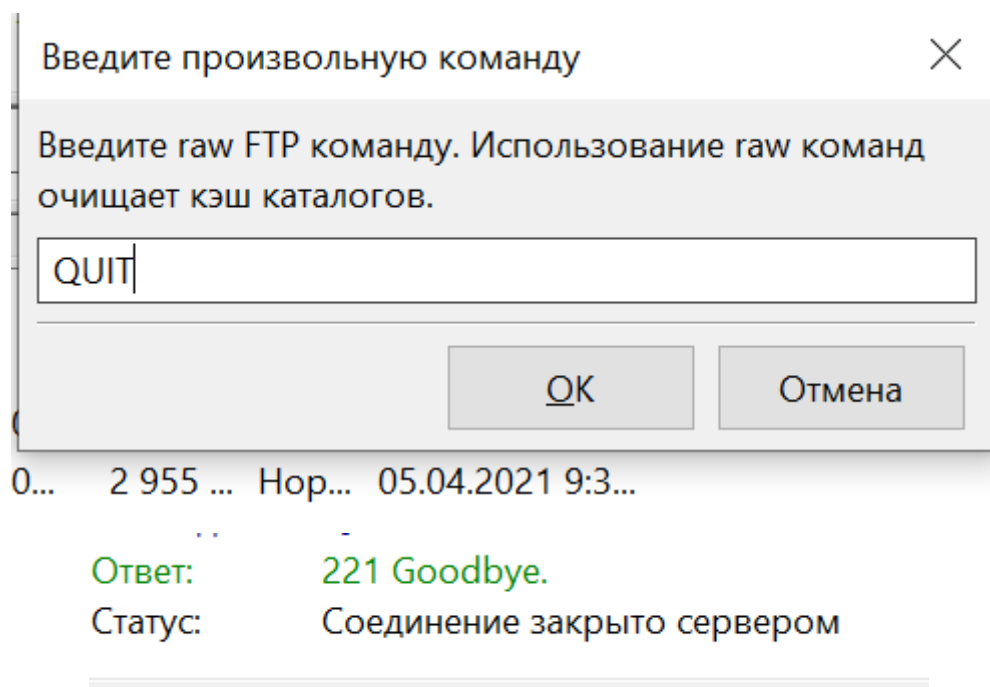
Выбор директории и скачивание файлов в пассивном режиме:

Статус: Начинаю закидывать C:\lab04.jpg  
 Статус: Файл передан успешно, передан 800 774 байт в 1 секунда

Попытка загрузить файл в режиме анонимного пользователя и обработка ошибки:

Команда: STOR cross\_0256x0256.png  
 Ответ: 550 Not enough privileges.

Завершение сессии:



## Текст программы

```
import os, sys, stat
import logging

from pyftplib.authorizers import DummyAuthorizer
from pyftplib.handlers import FTPHandler
from pyftplib.servers import FTPServer

authorizer = DummyAuthorizer()
authorizer.add_user("user", "12345", "C:/Users", perm="elradfmw")
authorizer.add_user("taty", "0805", "C:/Users", perm="elradfmw")
authorizer.add_anonymous(os.getcwd())

handler = FTPHandler
handler.authorizer = authorizer

server = FTPServer(("127.17.6.1", 1026), handler)
server.serve_forever()

class FTPThreadServer(threading.Thread):

    def QUIT(self, cmd):
        try:
            self.client.send('221 Goodbye.\r\n')
        except:
            pass
        finally:
            print ('Closing connection from ' + str(self.client_address) +
'...')

            self.close_datasock()
            self.client.close()
            quit()

    def LIST(self, cmd):
```

```

print ('LIST', self.cwd)
(client_data, client_address) = self.start_datasock()

try:
    listdir = os.listdir(self.cwd)
    if not len(listdir):
        max_length = 0
    else:
        max_length = len(max(listdir, key=len))

    header = '| %s | %9s | %12s | %20s | %11s | %12s |' % (max_length,
'Name', 'Filetype', 'Filesize', 'Last Modified', 'Permission', 'User/Group')
    table = '%s\n%s\n%s\n' % ('-' * len(header), header, '-' *
len(header))
    client_data.send(table)

    for i in listdir:
        path = os.path.join(self.cwd, i)
        stat = os.stat(path)
        data = '| %s | %9s | %12s | %20s | %11s | %12s |\n' %
(max_length, i, 'Directory' if os.path.isdir(path) else 'File',
str(stat.st_size) + 'B', time.strftime('%b %d, %Y %H:%M',
time.localtime(stat.st_mtime))
, oct(stat.st_mode)[-4:], str(stat.st_uid) + '/' +
str(stat.st_gid))
        client_data.send(data)

    table = '%s\n' % ('-' * len(header))
    client_data.send(table)

    self.client.send('\r\n226 Directory send OK.\r\n')
except (Exception, e):
    print ('ERROR: ' + str(self.client_address) + ': ' + str(e))
    self.client.send('426 Connection closed; transfer aborted.\r\n')
finally:
    client_data.close()
    self.close_datasock()

def PWD(self, cmd):
    self.client.send('257 "%s"\r\n' % self.cwd)

def CWD(self, cmd):
    dest = os.path.join(self.cwd, cmd[4:].strip())
    if (os.path.isdir(dest)):
        self.cwd = dest
        self.client.send('250 OK "%s"\r\n' % self.cwd)
    else:
        print ('ERROR: ' + str(self.client_address) + ': No such file or
directory.')
        self.client.send('550 "' + dest + '\": No such file or
directory.\r\n')

def CDUP(self, cmd):
    dest = os.path.abspath(os.path.join(self.cwd, '..'))
    if (os.path.isdir(dest)):
        self.cwd = dest
        self.client.send('250 OK "%s"\r\n' % self.cwd)
    else:

```

```

        print ('ERROR: ' + str(self.client_address) + ': No such file or
directory.')
        self.client.send('550 \'' + dest + '\": No such file or
directory.\r\n')

def MKD(self, cmd):
    path = cmd[4:].strip()
    dirname = os.path.join(self.cwd, path)
    try:
        if not path:
            self.client.send('501 Missing arguments <dirname>.\r\n')
        else:
            os.mkdir(dirname)
            self.client.send('250 Directory created: ' + dirname + '.\r\n')
    except (Exception, e):
        print ('ERROR: ' + str(self.client_address) + ': ' + str(e))
        self.client.send('550 Failed to create directory ' + dirname + '.')

def RMD(self, cmd):
    path = cmd[4:].strip()
    dirname = os.path.join(self.cwd, path)
    try:
        if not path:
            self.client.send('501 Missing arguments <dirname>.\r\n')
        else:
            os.rmdir(dirname)
            self.client.send('250 Directory deleted: ' + dirname + '.\r\n')
    except (Exception, e):
        print ('ERROR: ' + str(self.client_address) + ': ' + str(e))
        self.client.send('550 Failed to delete directory ' + dirname + '.')

def STOR(self, cmd):
    path = cmd[4:].strip()
    if not path:
        self.client.send('501 Missing arguments <filename>.\r\n')
        return

    fname = os.path.join(self.cwd, path)
    (client_data, client_address) = self.start_datasock()

    try:
        file_write = open(fname, 'w')
        while True:
            data = client_data.recv(1024)
            if not data:
                break
            file_write.write(data)

        self.client.send('226 Transfer complete.\r\n')
    except (Exception, e):
        print ('ERROR: ' + str(self.client_address) + ': ' + str(e))
        self.client.send('425 Error writing file.\r\n')
    finally:
        client_data.close()
        self.close_datasock()
        file_write.close()

```



```

def RETR(self, cmd):
    path = cmd[4:].strip()
    if not path:
        self.client.send('501 Missing arguments <filename>.\r\n')
        return

    fname = os.path.join(self.cwd, path)
    (client_data, client_address) = self.start_datasock()
    if not os.path.isfile(fname):
        self.client.send('550 File not found.\r\n')
    else:
        try:
            file_read = open(fname, "r")
            data = file_read.read(1024)

            while data:
                client_data.send(data)
                data = file_read.read(1024)

            self.client.send('226 Transfer complete.\r\n')
        except (Exception, e):
            print ('ERROR: ' + str(self.client_address) + ': ' + str(e))
            self.client.send('426 Connection closed; transfer aborted.\r\n')
        finally:
            client_data.close()
            self.close_datasock()
            file_read.close()

```