МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ

УНИВЕРСИТЕТ ИМЕНИ АКАДЕМИКА С.П. КОРОЛЕВА»

(САМАРСКИЙ УНИВЕРСИТЕТ)

Отчёт по лабораторной работе №3

«Исследование структуры FTP-сервера»  
по курсу «Сети ЭВМ и телекоммуникации»

Выполнил:  
Грибанов Д. Н.

гр.6301

Проверил:  
Еленев Д. В.

Самара 2020

**Задание:**

Разработать клиентское приложение, позволяющее пользователю получать следующую информацию о FTP-сервере:

- суммарный размер файлов, размещенных на сервере;

- суммарный размер размещенных на сервере файлов, сгруппированных по их типам (тип файлов определяется по его расширению);

- структуру каталогов FTP-сервера.

Адрес FTP-сервера, имя пользователя и пароль вводятся пользователем

**Код программы:**

**Сервер:**

from pyftpdlib.servers import FTPServer  
from pyftpdlib.authorizers import DummyAuthorizer  
from pyftpdlib.handlers import FTPHandler  
  
authorizer = DummyAuthorizer()  
authorizer.add\_user(**"lokesh"**, **"123"**, **"T:/University/Сеть ЭВМ/Лабораторные"**, perm=**"elradfmw"**)  
authorizer.add\_anonymous(**"T:/University/Сеть ЭВМ/Лабораторные"**, perm=**"elradfmw"**)  
  
handler = FTPHandler  
handler.authorizer = authorizer  
  
server=FTPServer((**"localhost"**,8080),handler)  
server.serve\_forever()

**Клиент:**

import os  
from ftplib import FTP  
  
  
class FTPController:  
  
 IGNORE\_LIST = [  
 **'idea'**,  
 **'\_\_pycache\_\_'**,  
 **'\_\_init\_\_'** ]  
  
 @staticmethod  
 def get\_directory\_list\_and\_size(host: str, port: int, user: str, passwd: str):  
 ftp = FTP()  
 ftp.connect(host, port=port)  
  
 #ftp.login(user=user, passwd=passwd)  
 ftp.sendcmd(**'USER '** + user)  
 ftp.sendcmd(**'PASS '** + passwd)  
  
 def get\_dirs\_ftp(folder=**""**, final\_size=0, size\_to\_each\_type={}):  
 contents = ftp.nlst(folder)  
 folders = []  
 for item in contents:  
 joined = os.path.join(folder, item)  
  
 # Ignore some folders  
 ignore = False  
 for ignore\_name in FTPController.IGNORE\_LIST:  
 if ignore\_name in item:  
 ignore = True  
  
 if ignore:  
 continue  
  
 # Save folder  
 if **"."** not in item:  
 folders.append(joined)  
 continue  
 # Write size of the file  
 if **'.'** in item:  
 ftp.voidcmd(**'TYPE i'**)  
 cur\_type = item.split(**'.'**)[-1]  
  
 # The SIZE command is defined in RFC-3659, i.e. with prefix '213'  
 #cur\_size = ftp.size(joined)  
 cur\_size = int(ftp.sendcmd(**'SIZE '** + joined)[3:].strip())  
  
 if size\_to\_each\_type.get(cur\_type) is None:  
 size\_to\_each\_type[cur\_type] = cur\_size  
 else:  
 size\_to\_each\_type[cur\_type] += cur\_size  
  
 final\_size += cur\_size  
 return folders, final\_size  
  
 def get\_all\_dirs\_ftp(folder=**"**\\**"**):  
 dirs = []  
 size\_to\_each\_type = {}  
  
 new\_dirs, final\_size = get\_dirs\_ftp(folder, final\_size=0, size\_to\_each\_type=size\_to\_each\_type)  
 while len(new\_dirs) > 0:  
 for dir in new\_dirs:  
 dirs.append(dir)  
  
 old\_dirs = new\_dirs[:]  
 new\_dirs = []  
 for dir in old\_dirs:  
 new\_folders, new\_size = get\_dirs\_ftp(dir, final\_size=0, size\_to\_each\_type=size\_to\_each\_type)  
 final\_size += new\_size  
 for new\_dir in new\_folders:  
 new\_dirs.append(new\_dir)  
 dirs.sort()  
 return dirs, final\_size, size\_to\_each\_type  
  
 allfiles, final\_size, size\_to\_each\_type = get\_all\_dirs\_ftp()  
  
 return allfiles, final\_size, size\_to\_each\_type

**Результат выполнения программы на примере:**

