МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИМЕНИ АКАДЕМИКА С.П. КОРОЛЕВА»

ФАКУЛЬТЕТ ИНФОРМАТИКИ

КАФЕДРА ПРОГРАММНЫХ СИСТЕМ

Отчет к лабораторной работе № 3

«Исследование структуры FTP-сервера»

по курсу «Сети ЭВМ и телекоммуникации»

Выполнил:

Медов Д.А.

гр. 6303-090301D

Проверил:

Пигусов А.С.

Самара 2021

Введение

Цель работы: Приобретение навыков работы с протоколом передачи файлов FTP и написания серверных приложений.

Разработать клиентское приложение, позволяющее получить суммарный размер файлов, размещенных на FTP сервере.

Листинг программы

import ftplib

import os

class ftp:

    def \_\_init\_\_(self, host):

        self.ftp = ftplib.FTP(host)

        self.list\_path = []

        self.list\_file = []

    def connect(self, login, passwd):

        self.ftp.login(login, passwd)

        self.\_parse\_ftp\_file(self.ftp.pwd())

    def \_parse\_ftp\_file(self, path):

        self.ftp.cwd(path)

        self.list\_path.append(path)

        list = self.ftp.nlst()

        for item in list:

            try:

                pwd = self.ftp.pwd()

                path = (pwd + "/" + item).replace("//", "/")

                self.ftp.cwd(path)

                self.\_parse\_ftp\_file(path)

                self.ftp.cwd(pwd)

            except ftplib.error\_perm:

                \_, file\_extension = os.path.splitext(item)

                self.list\_file.append({"file" : item, "size" : self.ftp.size(item), "extension" : file\_extension, "path" : pwd})

    def size\_all\_file(self):

        return sum((item["size"] for item in self.list\_file))

    def group\_file(self):

        group\_file = {}

        for item in self.list\_file:

            if item["extension"] in group\_file.keys():

                group\_file[item["extension"]].append(item)

            else:

                group\_file.setdefault(item["extension"], [item])

        return group\_file

    def get\_file(self):

        return self.list\_file

    def get\_path(self):

        return self.list\_path

    def disconect(self):

        self.ftp.close()

Результат выполнения программы:

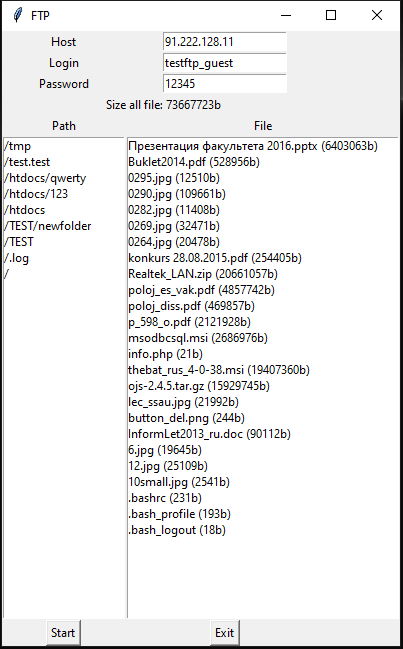


Рисунок 1 – Результат выполнения программы