Практическая работа №16
Разработка сетевых приложений на Python

1 Цель работы

1.1 Научиться создавать сетевые клиент-серверные приложения на Python;

1.2 Закрепить навык составления программ методами процедурного и событийно-ориентированного программирования.

2 Литература

2.1 Гуриков, С. Р. Основы алгоритмизации и программирования на Python : учебное пособие / С. Р. Гуриков. – Москва : ФОРУМ : ИНФРА-М, 2022. - URL:https://znanium.com/read?id=390096. – Режим доступа: для зарегистрир. пользователей. – Текст: электронный. – гл.12.

3 Подготовка к работе

3.1 Повторить теоретический материал (см. п.2).

3.2 Изучить описание практической работы.

4 Основное оборудование

4.1 Персональный компьютер.

5 Задание

5.1 Создание клиент-серверного сетевого приложения

5.1.1 Создание скрипта сервера

```
# Код сервера, принимающего подключение по порту 50007
# и пересылающего полученные данные отправителю
import socket
HOST = ''                 # Слушать все интерфейсы
PORT = 50007              # Порт, который необходимо слушать
# Создание интернет-сокета для использования по протоколу TCP
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

# прослушивание всех интерфейсов на порту PORT
s.bind((HOST, PORT))

# Перевод сокета в режим ожидания. Параметр – количество одновременных
# соединений (1 означает, что может быть одно соединение одновременно)
s.listen(1)
# ожидание соединения от клиента
conn, addr = s.accept()
print('Подключен клиент: ', addr) # addr – ip и номер порта клиента

# в цикле выполняется чтение данных от клиента
while True:
```

```
    # Считывание данных от клиента по 1024 байта
    data = conn.recv(1024)
    if not data:
        break
    # вывод текста в кодировке UTF-8
    print('Получено сообщение:', data.decode('utf-8'))
    # Отправка считанных данных обратно клиенту
    conn.sendall(data)

# Закрытие соединения
conn.close()
```

5.1.2 Создание скрипта клиента

```
# Код клиента, подключающегося к локальному серверу по порту 50007
# и пересылающего полученные данные отправителю
import socket
HOST = 'localhost'        # Имя сервера, к которому требуется подключиться
PORT = 50007              # Номер порта на сервере

# Создание интернет-сокета для использования по протоколу TCP
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
# Соединение с сервером
s.connect((HOST, PORT))

# отправка серверу байтовой строки
# (для преобразования в байты используется функция encode())
s.sendall('Hello, world'.encode())
# Прием от сервера данных
data = s.recv(1024)
# Закрытие соединения
s.close()
# Вывод на экран полученных данных
print('Сообщение сервера: ', repr(data))
```

5.1.3 Запустить для каждого скрипта свою копию IDLE (нужно для того, чтобы они работали в различных потоках) и осуществить сначала запуск сервера, потом запуск клиента.

5.2 Создание общего чата

5.2.1 Изменить код клиента следующим образом:

- убрать вывод строки, полученной с сервера;
- в бесконечном цикле запрашивать у пользователя ввод сообщения, которое должно отправляться на сервер;
- при вводе клиентом строки end работа клиента завершается.

5.2.2 Изменить код сервера следующим образом:

- реализовать постоянное прослушивание подключений (для этого добавить после s.listen(1) бесконечный цикл и перенести в него весь код, который написан после s.listen(1));

- убрать пересылку сообщения клиенту и вывод сообщения о том, что клиент подключился к серверу;

- реализовать вывод в командной строке полученных данных следующим образом "дата и время (адрес клиента): сообщение".

Для вывода даты и времени использовать следующий код:

```
from datetime import datetime
datetime.strftime(datetime.now(), "%Y.%m.%d %H:%M:%S")
```

5.2.3 Протестировать, запустив сервер и несколько клиентов

5.3 Создание общего чата с информированием о том, какие пользователи подключились и какой пользователь отправил сообщение

5.3.1 Изменить код клиента следующим образом:

- при запуске клиента запрашивать логин и отправлять его на сервер при подключении;

- в бесконечном цикле запрашивать у пользователя ввод сообщения и отправлять логин и сообщение серверу.

5.3.2 Изменить код сервера следующим образом:

- при подключении клиента должно выводиться сообщение «подключился логин» (логин д.б. тот, что указал клиент при подключении);

- реализовать вывод в командной строке полученных данных следующим образом "дата и время логин: сообщение".

5.3.3 Протестировать, запустив сервер и несколько клиентов

5.4 Создание оконного клиента

5.4.1 Создать оконное клиентское приложение с кнопкой "Отправить" и двумя полями ввода (логин и сообщение). При нажатии на кнопку "Отправить" приложение должно действовать аналогично клиенту из п.5.3.

5.4.2 Протестировать, запустив сервер из п.5.3 и несколько оконных клиентов

6 Порядок выполнения работы

6.1 Запустить Python IDLE и выполнить все задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Что такое «сервер»?

8.2 Что такое «клиент»?

8.3 Какой модуль применяется для разработки сетевых приложений на Python?
8.4 Как передать данные от клиента серверу в приложениях на Pyrhon?