Клиент-серверное приложение.

Возможно запустить сервер и клиентов отдельно через модули Server.py и Client.py, но удобней запустить через модуль start.py, так как этот модуль позволяет запустить сервер и клиентов клавишей ‘s’(Рис.1). А так же в самом модуле прописано количество клиентов (Рис. 3).

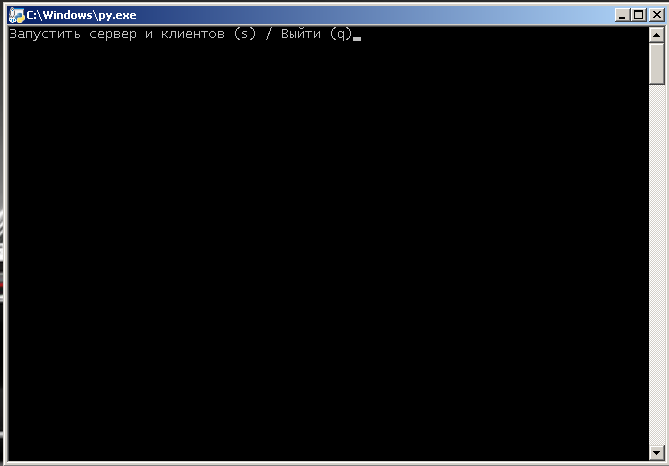


Рис 1.

После запуска через модуль start.py появляются консоль сервера с информацией о именах консолей клиентов, а так же ip и host клиентов; консоли пользователей с их именами. Для отправки сообщения нужно вести имя получателя, например User0 введет User1. Далее User0 можно отправлять свои сообщения (Рис. 2). Нажав q в главной консоли можно закрыть все процессы.

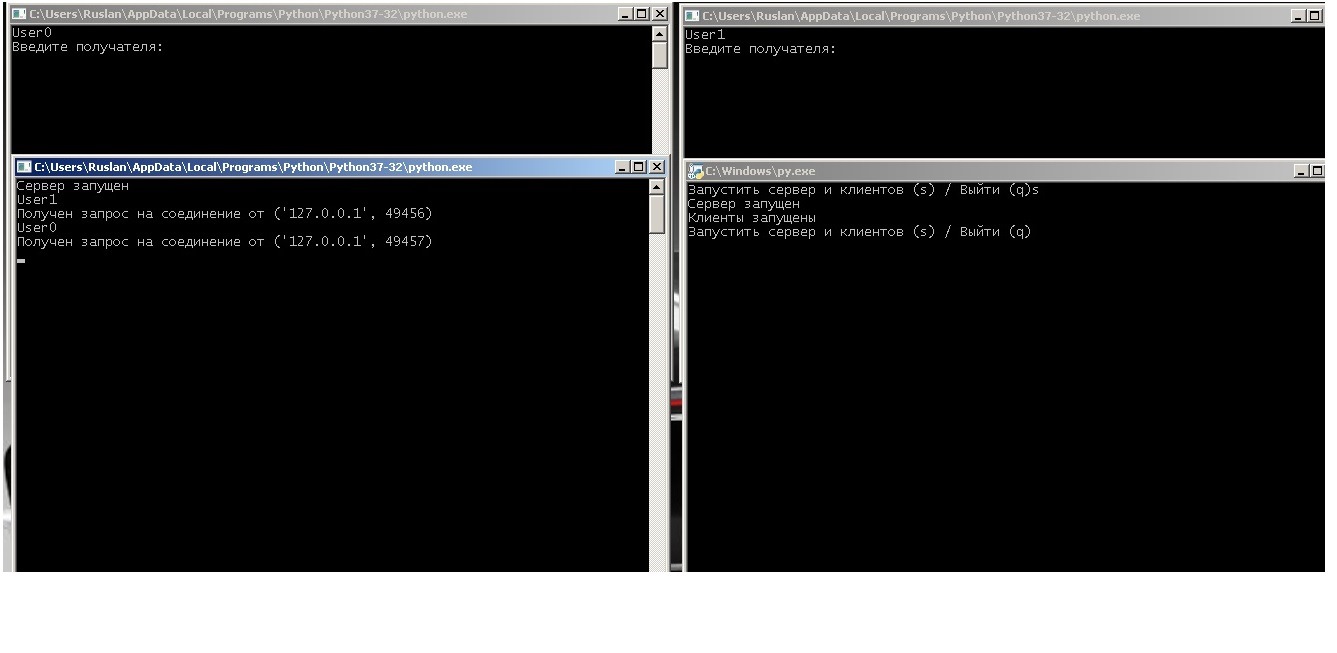


Рис. 2

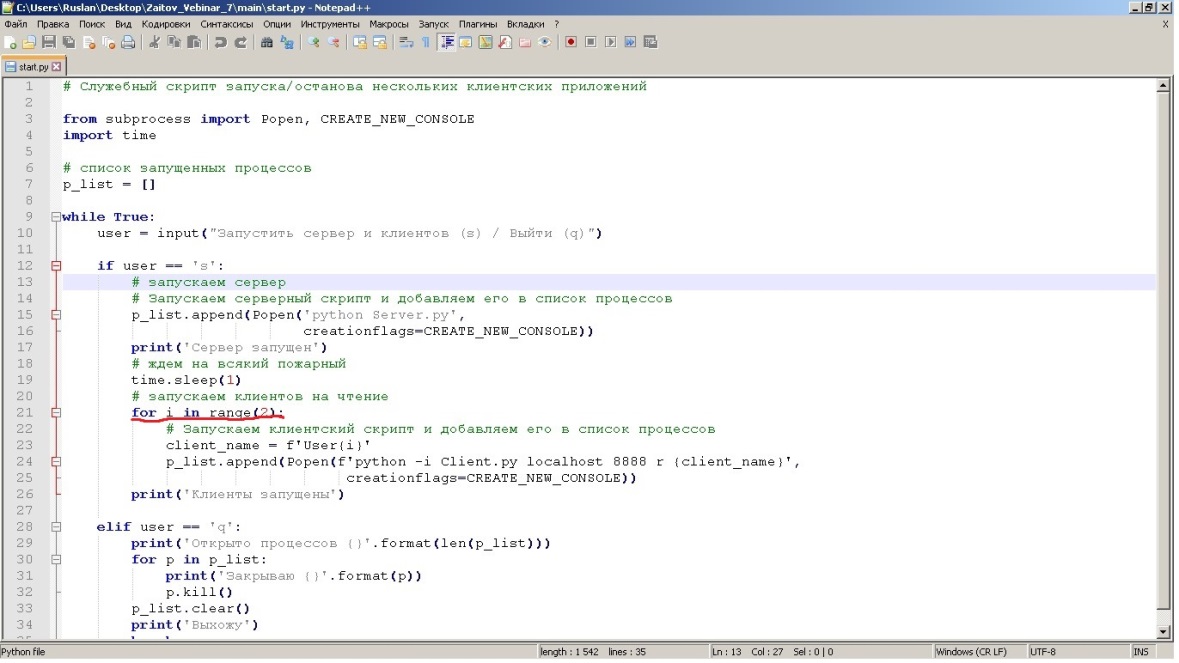


Рис. 3

Модуль Server.py.

read\_requests – собирает все сообщения пользователей в список и возвращает список

write\_responses – пересылает сообщение от отправителя к получателю

presence\_response – отправляет ответ клиенту в виде словаря при первом его подключении к серверу. {‘response:200’} при удачном, {‘response:400’} при неполадках. Можно усовершенствовать.

Далее идет основная часть создания сервера.

Реализована возможность запускать программу через консоль с параметрами.

Используется конструкция try-except для запуска программы с дефолтными параметрами.

В бесконечном цикле обрабатываются сообщения пользователей. Используется модуль select

Модуль Client.py.

read\_messages – клиент получает сообщения

create\_messages – клиент создает сообщение

create\_presence – клиент отправляет сообщение при первом подключении к серверу

Далее основная часть, где реализован запуск с параметрами через try-except

Взаимодействие клиентов идет через потоки.

Другие модули:

deco.py - декоратор для логгирования функций

errors.py – функции-ошибки, вызывающиеся при ошибках в основных действиях клиента

utils.py – основные функции программы, есть функции, которые пригодятся для графического интерфейса.

test\_client.py, test\_server.py – пробные юнит тесты