Пункты с 1 по 7 реализованы в файлах client.py и server.py 2 пункт. Клиент вводит exit и разрывает соединение

Изображение выглядит как текст

Автоматически созданное описание

3 пункт. Сервер продолжает слушать и можно подключиться повторно. Два клиента подключились и разорвали соединение

Изображение выглядит как текст

Автоматически созданное описание

4 пункт. номер порта и имя хоста (для клиента) спрашивается у пользователя. Безопасный ввод данных реализован модулем getpass и значения по умолчанию при пустом вводе.

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

при некорректном вводе ip или port соединение просто не установится и выведется ошибка, как таковая проверка ввода и не нужна.

5 пункт. Модифицировать код сервера таким образом, чтобы все служебные сообщения выводились не в консоль, а в специальный лог-файл. Используем модуль sys и изменяем файл вывода на наш файл.

Изображение выглядит как текст

Автоматически созданное описание





Изображение выглядит как текст

Автоматически созданное описание

6 пункт. Модифицируйте код сервера таким образом, чтобы он автоматически изменял номер порта, если он уже занят. Сервер должен выводить в консоль номер порта, который он слушает.

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

7 пункт. Сервер идентификации. Запросили у пользователя имя

Изображение выглядит как текст

Автоматически созданное описание

Записали имя пользователя и IP в файл

Изображение выглядит как текст

Автоматически созданное описание

Теперь при повторном подключении сервер нас поприветствует

Изображение выглядит как текст

Автоматически созданное описание

Пункты с 8 по 10 реализованы в файлах КЛИЕНТ.py и СЕРВЕР.py

8 пункт. Реализовать сервер аутентификации. Похоже на предыдущее задание, но вместе с именем пользователя сервер отслеживает и проверяет пароли. Дополнительные баллы за безопасное хранение паролей. Дополнительные баллы за поддержание сессии на основе токена наподобие cookies При первом подключении сервер попросил придумать пароль, далее он сохранил пароль и уже предложил авторизоваться, после авторизации. Пароли в файлах хранятся в хешированном виде с иcпользованием алгоритма шифрования md5. При авторизации сервер отправляет клиенту токен, который действует одну сессию, при отправке сообщения клиентом, к нему добавляется токен, сервер проверяет верен ли токен, если токен верен, сервер обрабатывает сообщение. информация о логине/пароле сохраняется в зашифрованном виде в csv файл

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

теперь сервер видит, что клиент зарегистрировался и авторизовался и может обмениваться с ним сообщениями

Изображение выглядит как текст

Автоматически созданное описание

9 пункт. Напишите вспомогательные функции, которые реализуют отправку и принятие текстовых сообщений в сокет. Функция отправки должна дополнять сообщение заголовком фиксированной длины, в котором содержится информация о длине сообщения. Функция принятия должна читать сообщение с учетом заголовка. В дополнении реализуйте преобразование строки в байтовый массив и обратно в этих же функциях. Дополнительно оценивается, если эти функции будут реализованы как унаследованное расширение класса socket библиотеки socket.

работа отправки/ приемки сообщений приведена в пункте 8, там можно увидеть, что выводится длина сообщений, а сообщения передаются как байтовые массивы, причем вспомогательные куски на экран не выводятся. Функции реализованы отдельно и созданы как унаследованное расширение класса socket. Функции приема отправки клиента

Изображение выглядит как текст

Автоматически созданное описание

Функции приема отправки сервера

Изображение выглядит как текст

Автоматически созданное описание

10 пункт. Дополните код клиента и сервера таким образом, чтобы они могли посылать друг другу множественные сообщения один в ответ на другое. Как видим сервер и клиент могут переписываться последовательно. Сначала сообщения сервера, потом клиента.

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

Если клиент отсоединится сервер будет ждать новое подключение.

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание