Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное

учреждение высшего образования

«СибирскИЙ государственнЫЙ Университет

геоСИСТЕМ И ТЕХНОЛОГИЙ»

(СГУГИТ)

ОТЧЕТ

ЛАБОРАТОРНАЯ РАБОТА 8. КЛИЕНТ-СЕРВЕРНОЕ ПРОГРАММИРОВАНИЕ В PYTHON

Выполнил обучающийся

Группы БИ-33

Ахиярова И.М.

Проверил

Воронкин Е.Ю.

Новосибирск – 2021

**Цель работы:** научиться разрабатывать клиент-серверные приложения.

**Задачи работы**

Ознакомиться с этапами разработки клиент-серверных приложений.

**Перечень обеспечивающих средств**

Для выполнения работы необходимо иметь компьютер с установленной операционной системой семейства Windows, установленным python и IDE PyCharm Professional.

**Задание**

Разработать серверную и клиентскую модель, для обмена сообщениями. Программа сервера сокетов запускается сначала и ждет любого запроса. Клиентская программа сначала инициирует диалог. Затем серверная программа будет реагировать на запросы клиента соответственно. Клиентская программа будет завершена, если пользователь введет сообщение «до свидания». Серверная программа также завершится, когда завершится клиентская программа, это необязательно, и мы можем поддерживать выполнение серверной программы на неопределенный срок или завершить работу с помощью какой-либо конкретной команды в клиентском запросе.

**Технология выполнения работы**

Создадим два python файла: один для клиента, другой для сервера. В обоих случаях импортируем встроенную библиотеку socket.

Сначала создадим серверное приложение. Получим адрес и установим порт. Затем создадим сокет и свяжем адрес с портом. Установим у сервера количество подключений и дадим серверу команду принимать подключения. Создадим основной цикл, в котором с помощью созданного подключения сервер сможет принимать и отправлять сообщения. Создадим условие. Если сервер не получил данных, основной цикл заканчивается и сервер отключается.

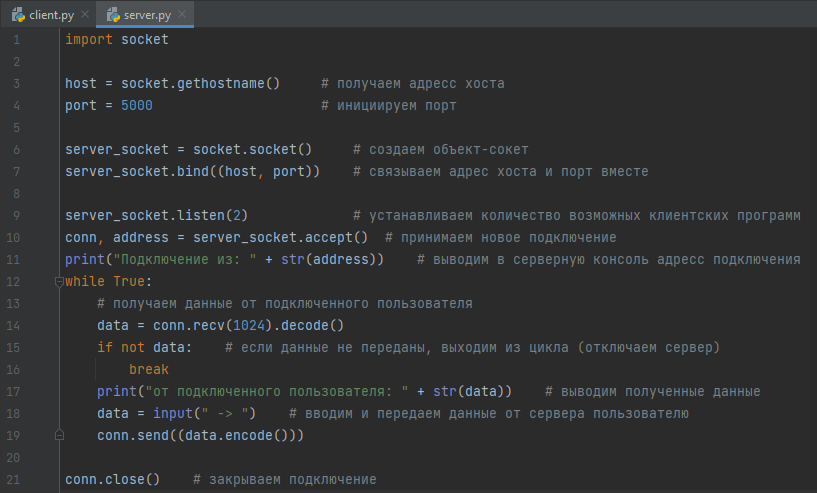


Рисунок 1 – Серверное приложение

Далее создадим клиентское приложение. Так как сервер и клиент запускаются с одного компьютера, мы можем получить адрес хоста сразу, используя ту же команду gethostname, что и в серверном приложении. Создадим сокет и подключимся к серверу. С помощью созданного подключения мы получаем возможность обмениваться с сервером сообщениями.

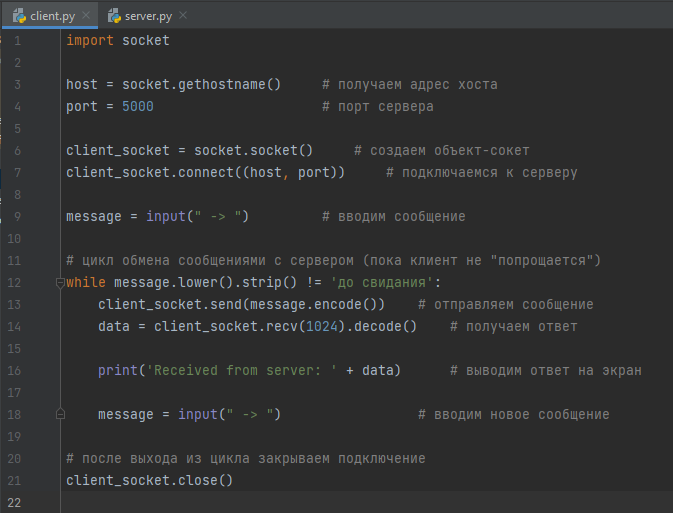


Рисунок 2 – Клиентское приложение

Теперь запустим оба приложения и проверим правильность их работы.

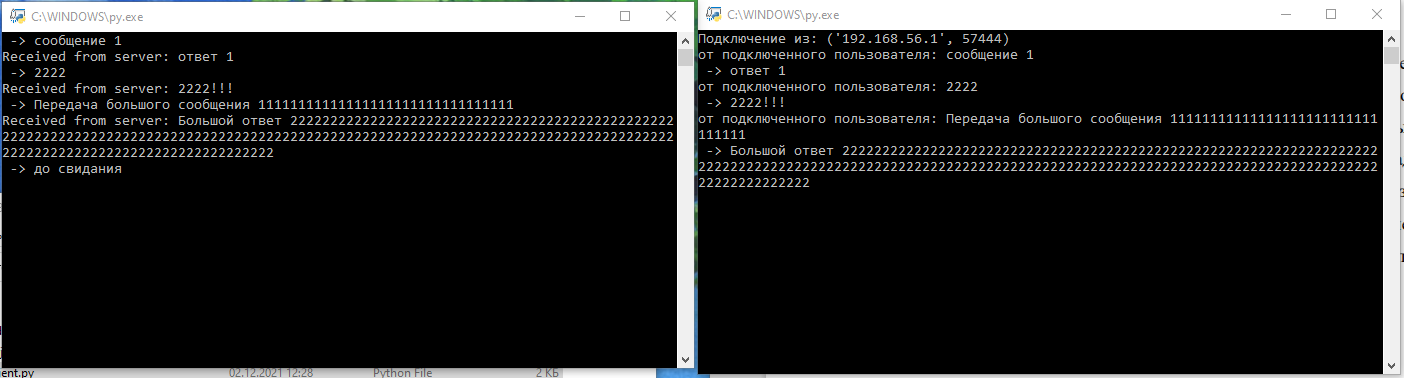


Рисунок 3 – Обмен сообщениями между сервером и клиентом

После нажатия Enter оба окна закроются.

**Вывод:** я научилась разрабатывать клиент-серверные приложения.

**Контрольные вопросы**

1. Назовите этапы разработки клиент-серверного приложения?

Этапы:

1. Определение целей приложения;
2. Формулирование требований к функционалу;
3. Выбор языка программирования и средств разработки;
4. Построение:
   1. Разработка дизайна клиентских компонентов и проектирование интерфейсов клиент-серверного взаимодействия;
   2. Реализация серверных компонентов и интерфейсов клиент-серверного взаимодействия;
5. Тестирование;
6. Внедрение.
7. Основные особенности клиент-серверной архитектуры?

Особенности:

1. Асимметричность протоколов. Между клиентами и сервером существуют отношения «один ко многим». Инициатором диалога с сервером обычно является клиент.
2. Инкапсуляция услуг. После получения запроса на услугу от клиента, сервер решает, как должна быть выполнена данная услуга.
3. Целостность. Программы и общие данные для сервера управляются централизованно, что снижает стоимость обслуживания и защищает целостность данных. В то же время, данные клиентов остаются персонифицированными и независимыми.
4. Местная прозрачность. Сервер – это программный процесс, который может исполняться на той же машине, что и клиент, либо на другой машине, подключенной по сети. Программное обеспечение «клиент-сервер» обычно скрывает местоположение сервера от клиентов, перенаправляя запрос на услуги через сеть.
5. Обмен на основе сообщений. Клиенты и сервер являются нежёстко связанными («loosely-coupled») процессами, которые обмениваются сообщениями: запросами на услуги и ответами на них.
6. Модульный дизайн, способный к расширению. Модульный дизайн программной платформы «клиент-сервер» придаёт ей устойчивость к отказам, то есть, отказ в каком-то модуле не вызывает отказа всего приложения.
7. Независимость от платформы. Идеальное приложение «клиент-сервер» не зависит от платформ оборудования или операционной системы. Клиенты и серверы могут развёртываться на различных аппаратных платформах и разных операционных системах.
8. Масштабируемость. Системы «клиент-сервер» могут масштабироваться как горизонтально (по числу серверов и клиентов), так и вертикально (по производительности и спектру услуг).
9. Разделение функционала. Система «клиент-сервер» — это соотношение между процессами, работающими на одной или на разных машинах. Сервер – это процесс предоставления услуг. Клиент – это потребитель услуг.
10. Общее использование ресурсов. Один сервер может предоставлять услуги множеству клиентов одновременно, и регулировать их доступ к совместно используемым ресурсам.
11. Что такое сокет?

Сокет – это конечная точка двунаправленного канала связи между сервером и клиентом.

1. Что такое сервер?

Сервер – это программное обеспечение, которое ожидает запросов клиентов и обслуживает или обрабатывает их соответственно.

1. Что такое порт?

Порт – это числовой идентификатор программы или процесса, которые обслуживают сетевые соединения на заданном сетевом адресе.

1. Что такое клиент?

Клиент – компьютерное устройство, которое отсылает запросы серверу, касающиеся выполнения определенных задач или предоставления конкретной информации.