Министерство высшего и профессионального образования РФ

Южно Уральский государственный университет

Кафедра «Системного программирования»

Лабораторная работа №1

Простое сетевое приложение

Выполнил:

студенты группы ВМИ-312

Меньшиков А.П. \_\_\_\_\_\_\_\_\_\_\_\_

Проверил:

Громов И.А. \_\_\_\_\_\_\_\_\_\_\_

Челябинск

2013

## Цель работы

Ознакомиться с языком программирования Python

Ознакомиться с работой системы контроля версий Git

Написать сетевое приложение и сохранить результат работ в репозитории Git

## Техническое задание

Написать эхо-сервер. Сервер должен слушать указанный TCP порт, транслировать данные из входящего соединения в исходящее

## Ход работы

1. Репозиторий Git для выполнения лабораторной работы я создал на сайте github.com
2. *git init*
3. *git add README.md*
4. *git commit -m "first commit"*
5. *git remote add web https://github.com/AlexPMenshikov/web.git*
6. *git push -u origin web*
7. Код программы
8. *import socket*
9. Создается новое сетевое соединение и возвращается объект класса socket под именем server\_socket, константа AF\_INET задает вид адреса и константа SOCK\_STREAM задает тип соединения
10. *server\_socket = socket.socket(socket.AF\_INET,socket.SOCK\_STREAM)*
11. Задаем адрес и порт
12. *host = 'localhost'*
13. *port = 8000*
14. Привязываем соединение к адресу и порту
15. *server\_socket.bind((host, port))*
16. Сколько клиентов может подключится к серверу
17. *server\_socket.listen(1)*
18. *while 1:*
19. Принимаем соединение и возвращаем conn – объект готовый для получения и посылки данных
20. *conn , addr = server\_socket.accept()*
21. *print 'client is at', addr*
22. Получаем данные от клиента в строку data
23. *data =conn.recv(1024)*
24. Отсылаем данные клиенту
25. *conn.send(data)*
26. *break*
27. Закрываем соединение
28. *conn.close()*

## Результат

Получил начальные знания о Python, а также о создании сервера на Python. А также расширил знания о Git.