Министерство образования Республики Беларусь

Учреждение образования

«Брестский Государственный технический университет»

Кафедра ИИТ

**Лабораторная работа №1**

По дисциплине «Аппаратное и программное обеспечение сетей»

Тема: «. Организация TCP – сервера и TCP – клиента»

**Выполнил:**

Студент 3 курса

Группы ИИ-21

Карагодин Д.Л.

**Проверил:**

Степанчук В.И.

Брест 2023

**Цель работы:** Изучить основы программирования сетевых приложений; Приобрести навыки по практическому использованию библиотеки для реализации сетевых приложений на базе протоколов TCP.

**Ход работы:**

Разработать программу работы TCP-эхо-сервера и TCP-клиента, выполняющую функции согласно варианта задания (см. приложения). Продемонстрировать работу системы преподавателю.

**Вариант 12:**

|  |  |  |
| --- | --- | --- |
| **Вариант задания для бригады студентов** | Номер задания для реализации TCP (UDP) сервера | Номер задания для реализации TCP (UDP) клиента |
| **12** | 4 | 1 |

|  |  |
| --- | --- |
| Номер задания | Функция, выполняемая TCP-сервером |
| 4 | После приема каждой группы из 64 символов от клиента сервер формирует и отсылает клиенту цепочку <«символ1-количество1», «символ2-количество2» и т.д.>, где «символ» - встреченный в группе символ, а «количество» - их количество в группе. Если в очередной группе число разных символов меньше 3, то отсылается соответствующее сообщение клиенту и сервер разрывает соединение. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Вариант | Ввод символов с отсылкой введенной строки по нажатию на клавишу | Ведение файла протокола событий, включающих:  1) время начала и окончания соединения;  2) передаваемую серверу строку и время передачи строки; | Возможность разрыва соединения при помощи команды:  **disconnect <адрес> <порт>** | 2) Автоматическое подключение к серверу с заданным по умолчанию адресом при запуске клиента |
| 1 | Home | 1), 2) | + | 2) |

**Код программы Эхо-сервера:**

import socket

def process\_group(group):

    unique\_chars = set(group)

    if len(unique\_chars) < 3:  # Если число разных символов меньше 3, разрываем соединение

        return None

    result = []

    for char in unique\_chars:

        count = group.count(char)

        result.append(f"{char}-{count}")

    return ', '.join(result)

def run\_server(host, port):

    with socket.socket(socket.AF\_INET, socket.SOCK\_STREAM) as server\_socket:

        server\_socket.bind((host, port))

        server\_socket.listen(1)

        print(f"Server listening on {host}:{port}")

        while True:

            client\_socket, addr = server\_socket.accept()

            print(f"Connection established from {addr}")

            data = client\_socket.recv(64).decode()

            while data:

                result = process\_group(data)

                if result is None:  # Если число разных символов меньше 3

                    client\_socket.send(b"Connection closed")

                    break

                client\_socket.send(result.encode())

                data = client\_socket.recv(64).decode()

            client\_socket.close()

            print(f"Connection closed from {addr}")

# Запуск сервера

run\_server('localhost', 8000)

**Код программы клиента:**

import socket

import datetime

def create\_event\_log(connection\_start\_time, connection\_end\_time, transmitted\_string, transmission\_time):

    log = f"Connection start time: {connection\_start\_time}\n"

    log += f"Connection end time: {connection\_end\_time}\n"

    log += f"Transmitted string: {transmitted\_string}\n"

    log += f"Transmission time: {transmission\_time}\n"

    return log

def run\_client(host, port):

    with socket.socket(socket.AF\_INET, socket.SOCK\_STREAM) as client\_socket:

        client\_socket.connect((host, port))

        print(f"Connected to {host}:{port}")

        connection\_start\_time = datetime.datetime.now()

        transmitted\_string = ""

        while True:

            user\_input = input("Enter characters: ")

            if user\_input == "disconnect":

                client\_socket.send(user\_input.encode())

                break

            transmitted\_string += " " + user\_input

            client\_socket.send(user\_input.encode())

            data = client\_socket.recv(1024).decode()

            print("Received:", data)

            if data == "Connection closed":

                break

        connection\_end\_time = datetime.datetime.now()

        transmission\_time = connection\_end\_time - connection\_start\_time

        event\_log = create\_event\_log(connection\_start\_time, connection\_end\_time, transmitted\_string, transmission\_time)

        with open(".\event\_log.txt", "w") as file:

            file.write(event\_log)

        client\_socket.close()

        print("Connection closed")

# Запуск клиента

run\_client('localhost', 8000)

**Результат программы Эхо-сервера:**

**Результат программы клиента:**

**Вывод:** изучил основной элемент нейронной сети — нейрона — и принципов построения на основе нейрона простейшей нейронной сети — персептрона.