МИНЕСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧЕРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«Брестский государственный технический университет»

Кафедра «Интеллектуальные информационные технологии»

Лабораторная работа №1

По дисциплине «Аппаратное и программное обеспечение сетей»

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

Выполнил:

студент 3 курса

группы ИИ-21(2)

Литвинюк Т. В.

Проверил:

Степанчук В.И.

Брест 2023

Цель работы: 1) изучить основы программирования сетевых приложений Windows на базе библиотеки WINSOCK2.H; 2) приобрести навыки по практическому использованию библиотеки для реализации сетевых приложений в среде С++ на базе протоколов TCP.

Вариант 7.

|  |  |  |
| --- | --- | --- |
| Вариант задания для бригады студентов | Номер задания для реализации TCP (UDP) сервера (см. табл. 2). | Номер задания для реализации TCP (UDP) клиента (см. табл. 3). |
| 7 | 6 | 2 |

|  |  |
| --- | --- |
| Номер задания | Функция, выполняемая TCP-сервером |
| 6 | После приема каждой группы из 48 символов сервер отсылает подтверждение в виде контрольной суммы ASCII-кодов символов группы и количество символов, принятых сервером от начала сеанса. Если в потоке принятых символов встречается последовательность «~#~», то сервер отсылает сообщение об окончании сеанса и разрывает соединение. |

Таблица 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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;  3) принимаемую от сервера строку и время приема строки. | Возможность разрыва соединения при помощи команды:  **disconnect <адрес> <порт>** | 1) Задание в программе клиента специальной команды и параметров: подключения к серверу: **сonnect**  **<адрес> <порт>**  2) Автоматическое подключение к серверу с заданным по умолчанию адресом при запуске клиента |
| 2 | PgUp | 1), 3) | - | 1) |

Таблица 3

Реализация TCP-сервера:

import socket

server = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

server.bind(('localhost', 8080))

server.listen(5)

user\_socket, address = server.accept()

print("user connected")

all\_messages = ""

while True:

data = user\_socket.recv(128)

all\_messages += data.decode()

if len(all\_messages) > 48:

user\_socket.send(f"!control\_sum:{sum(map(lambda char: ord(char), all\_messages[::48]))}").encode() # отправить сумму ASCII

if '~#~' in all\_messages:

user\_socket.send(f"!close\_connection".encode())

break

print(data.decode())

user\_socket.send(data)

Реализация TCP-клиента:

from datetime import datetime

import socket

import threading

import keyboard

file = open("log.txt", 'w')

client = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

while True:

command = input(">>> ")

if command.startswith("connect"):

ip\_address, host = command.split()[1:]

break

client.connect((ip\_address, int(host)))

file.write(str(datetime.now()) + " --- connection to server\n")

FLAG = True

def exit():

global FLAG

keyboard.wait('esc')

FLAG = False

file.write(str(datetime.now()) + " --- disconnect from server\n")

file.close()

def input\_handler():

global FLAG

while FLAG:

command = input()

client.send(command.encode())

message = client.recv(128).decode()

file.write(message + f" --- {datetime.now()}\n")

print(message)

exit\_thread = threading.Thread(target=exit)

input\_thread = threading.Thread(target=input\_handler)

exit\_thread.start()

input\_thread.start()

exit\_thread.join()

input\_thread.join()

client.close()

file.write(str(datetime.now()) + " --- disconnect from server\n")

file.close()

**Вывод:** 1) изучил основы программирования сетевых приложений Windows на базе библиотеки WINSOCK2.H; 2) приобрёл навыки по практическому использованию библиотеки для реализации сетевых приложений в среде С++ на базе протоколов TCP.