МIНIСТЕРСТВО ОСВIТИ I НАУКИ УКРАЇНИ

НАЦIОНАЛЬНИЙ ТЕХНIЧНИЙ УНIВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛIТЕХНIЧНИЙ IНСТИТУТ»

Кафедра прикладної математики

ФАЙЛОВА АБСТРАКЦIЯ В ОС WINDOWS

лабораторна робота №4 з дисциплiни:

«Операцiйнi системи»

|  |  |
| --- | --- |
| Виконав: | Керівник: |
| студент групи КМ-63 | *асистент Громова В. В.* |
| *Артеменко Я.К.* |  |

Київ — 2019

# **ЗМІСТ**

[**ЗМІСТ** 1](#_Toc3730304)

[**Вступ** 2](#_Toc3730305)

[**1 ПОСТАНОВКА ЗАДАЧІ** 3](#_Toc3730306)

[**2** **ОПИС ПРОГРАМИ** 5](#_Toc3730307)

[**ВИСНОВКИ** 6](#_Toc3730308)

[**Додаток А (скріншоти результату)** 7](#_Toc3730309)

[**Додаток Б (текст програми)** 8](#_Toc3730310)

# **Вступ**

У традицiйному розумiннi, файл — це iменована область даних у зов-

нiшнiй пам’ятi. В операцiйнiй системi (ОС) Windows та у Windows API,

пiд файлом розумiють як традицiйнi файли, так i комунiкацiйнi ресур-

си, дисковi пристрої, консолi, iменованi конвеєри, поштовi ящики тощо.

Рiзнi типи файлiв подiбнi один на одний, але вiдрiзняються в окремих

властивостях та областях застосування.

Windows API надає програмiсту функцiї, якi дозволяють аплiкацiям

отримувати доступ до файлiв (у широкому сенсi слова) незалежно вiд

файлової системи та пристрою зберiгання файлiв.

У данiй роботi розглядатимемо засоби роботи з двома видами фай-

лової абстракцiї — iменованими конвеєрами та поштовими ящиками — в

ОС Windows версiї 2000 та вище.

# **1 ПОСТАНОВКА ЗАДАЧІ**

У данiй лабораторнiй роботi потрiбно ознайомитися з концепцiєю

файлової абстракцiї в ОС Windows, навчитися створювати аплiкацiї, якi

для обмiну даними використовують засоби Windows API для роботи з

iменованими конвеєрами або поштовими ящиками.

У рамках виконання лабораторної роботи потрiбно:

а) ознайомитися з теоретичними вiдомостями, викладеними в роз-

дiлi 2;

б) написати за допомогою Windows API будь-якою мовою програ-

мування двi аплiкацiї:

1) сервер поштового ящика або сервер iменованого конвеєра (на

вибiр);

2) клiєнт поштового ящика або клiєнт iменованого конвеєра, вiд-

повiдно;

сервер та клiєнт повиннi використовувати тiльки засоби Windows

API i не повиннi використовувати iншi можливостi систем програмування;

в) вiдлагодити програму в ОС Windows версiї 2000 та вище;

г) пiдготувати звiт iз лабораторної роботи вiдповiдно до вимог роз-

дiлу 3.

Кожен студент повинен написати сервер та клiєнт вибраної файло-

вої абстракцiї для виконання деякої конкретної задачi. Сервери та клiєнти

файлової абстракцiї рiзних студентiв не можуть мати однаковi функцiо-

нальнi можливостi.

# **ОПИС ПРОГРАМИ**

В межах даної лабораторної роботи було розроблено дві програми, одна з яких відповідає за сервер, а інша за клієнтську частину. Було організовано чат з підключенням до 100 клієнтів.

*Загальні відомості про програму:*

Програма була написана мовою C з застосуванням засобів для створення серверу та роботи в сервері.

Серверна частина має назву Server, а клієнтська – Client. В серверній, за допомогою циклу **for** було організовано можливість підключення 100 клієнтів. Створення підключення клієнту відбувається за допомогою newConnection = accept(sListen, (SOCKADDR\*)&addr, &sizeofaddr). Коли клієнт підключився, то з серверу приходить повідомлення «Hello. This is server» за допомогою функції send(newConnection, msg, sizeof(msg), NULL). Для можливості відправки повідомлень між користувачами використовується багатопоточність, яка створюється завдяки CreateThread(NULL, NULL, (LPTHREAD\_START\_ROUTINE)ClientHandler, (LPVOID)(i), NULL, NULL).

В клієнтській частині було організовано відправку повідомлень на сервер завдяки send(Connection, msg1, sizeof(msg1), NULL);, підключення до серверу за допомогою connect(Connection, (SOCKADDR\*)&addr, sizeof(addr)) та отримання повідомлень за допомогою recv(Connection, msg, sizeof(msg), NULL);

Функції, які використовує програма:

1. **ClientHandler(сервер) –** відправляє повідомлення всім користувачам, які підключені, окрім того, що відправив це повідомлення.
2. **ClientHandler(клієнт)**– отримує повідомлення з серверу, яке відправив інший користувач.
3. **main(сервер) –** в даній функції створюється сервер, виконується підключення клієнтів до нього та організація відправки повідомлень клієнтам;
4. **main(клієнт) –** відбувається з’єднання з сервером та відправка повідомлень іншим користувачам через сервер.

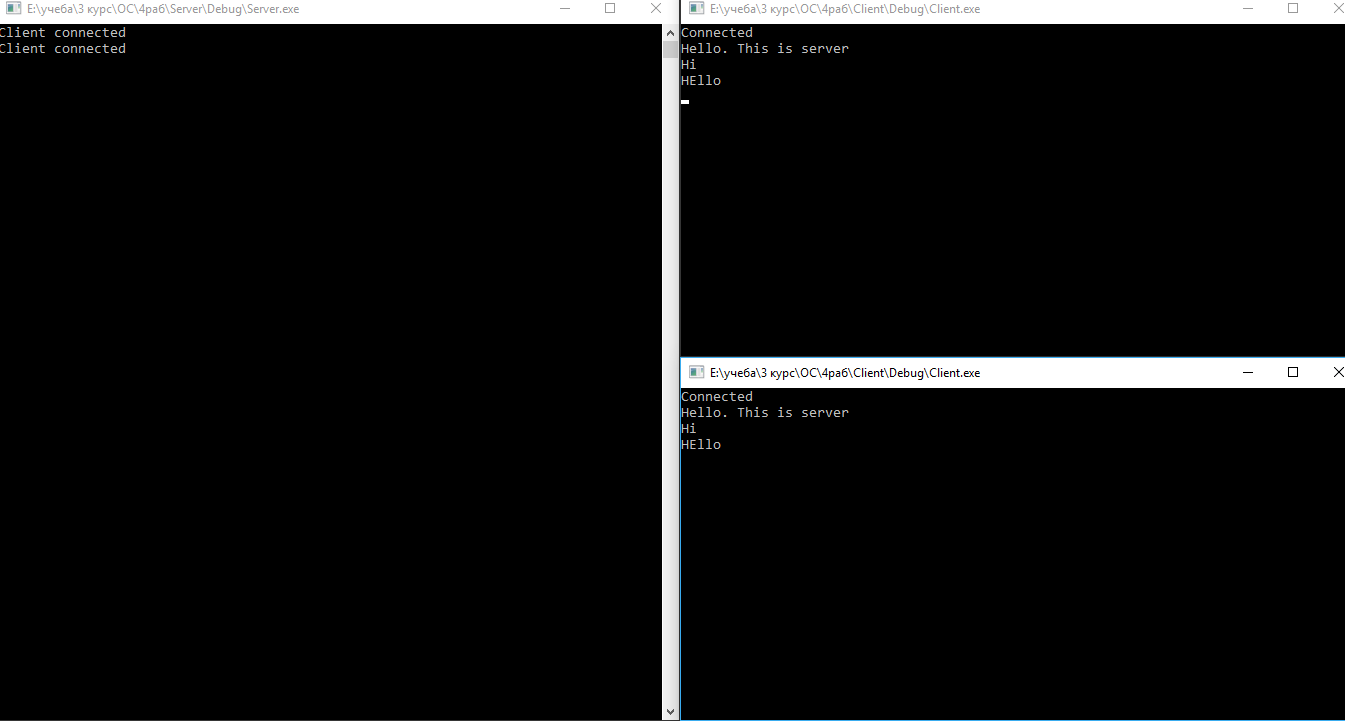
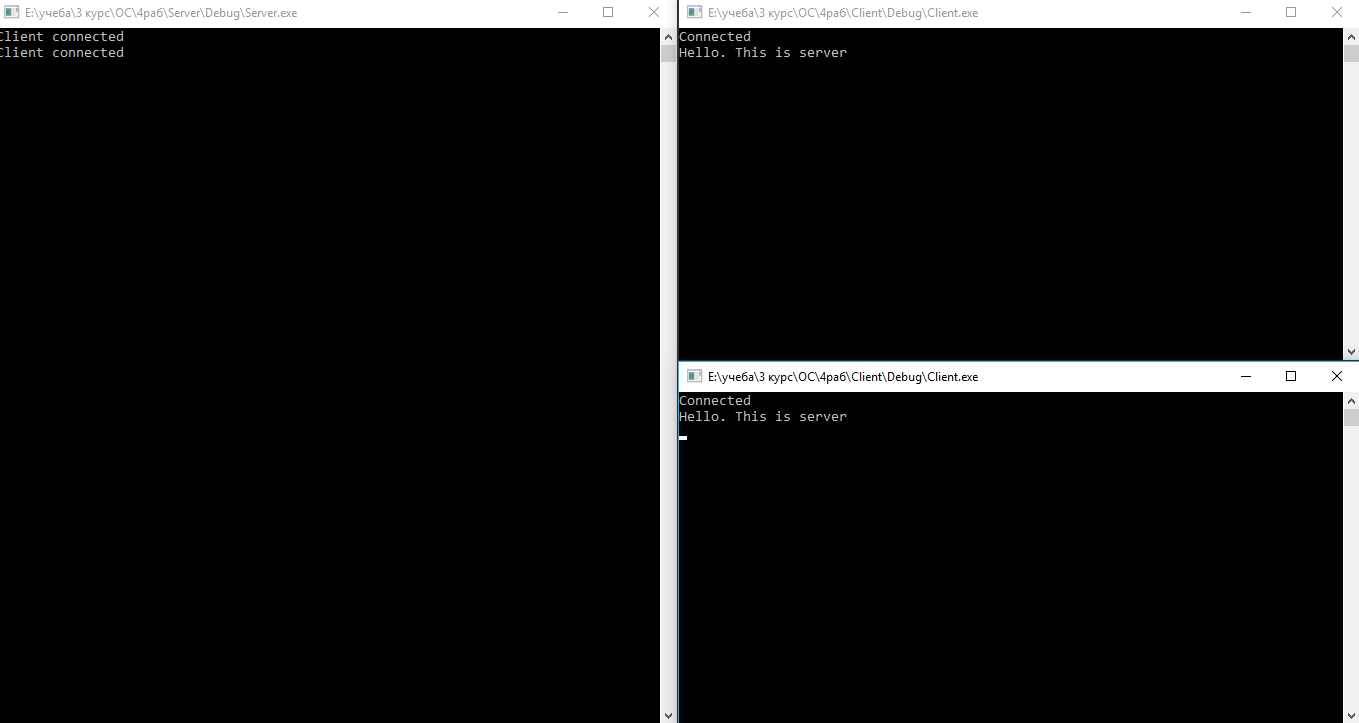
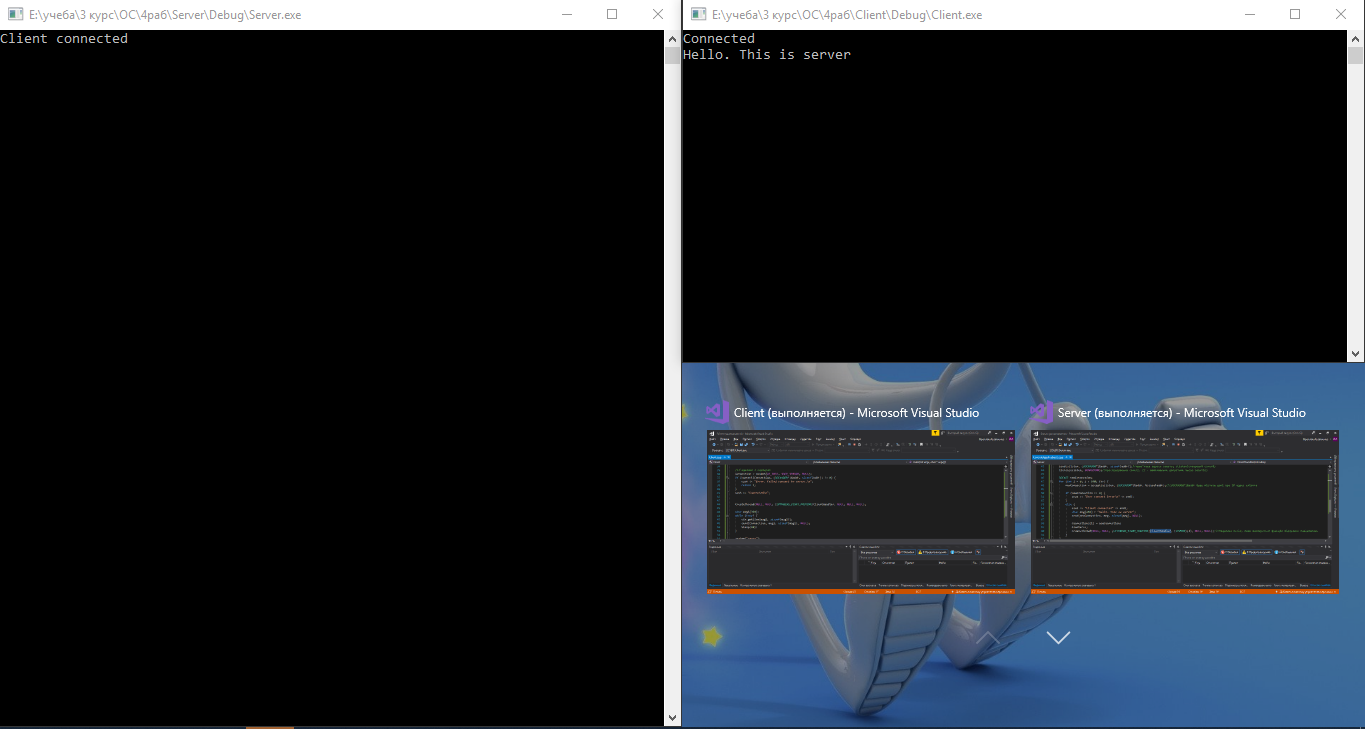
# 

# **ВИСНОВКИ**

На цій лабораторній роботі було здобуто навички по створенню серверної та клієнтської частини, та організовано роботу чату з підтримкою до 100 користувачів.

# 

# **Додаток А (скріншоти результату)**



# **Додаток Б (текст програми)**

Серверна частина

#include "pch.h"

#pragma comment(lib, "ws2\_32.lib")

#include <WinSock2.h>

#include <iostream>

#pragma warning(disable: 4996)

using namespace std;

SOCKET Connections[100];

int Counter = 0;

void ClientHandler(int index) {

char msg[256];

while (true) {

recv(Connections[index], msg, sizeof(msg), NULL);

for (int i = 0; i < Counter; i++) {

if (i == index) {

continue;

}

send(Connections[i], msg, sizeof(msg), NULL);

}

}

}

int main(int argc, char\* argv[])

{

WSADATA wsaData;

WORD DLLVersion = MAKEWORD(2, 1);

if (WSAStartup(DLLVersion, &wsaData) != 0) {

cout << "Library error" << endl;

exit(1);

}

SOCKADDR\_IN addr;

int sizeofaddr = sizeof(addr);

addr.sin\_addr.s\_addr = inet\_addr("127.0.0.1");

addr.sin\_port = htons(1111);

addr.sin\_family = AF\_INET;

SOCKET sListen = socket(AF\_INET, SOCK\_STREAM, NULL);

bind(sListen, (SOCKADDR\*)&addr, sizeof(addr));

listen(sListen, SOMAXCONN);

SOCKET newConnection;

for (int i = 0; i < 100; i++) {

newConnection = accept(sListen, (SOCKADDR\*)&addr, &sizeofaddr);

if (newConnection == 0) {

cout << "User connect Error\n" << endl;

}

else {

cout << "Client connected" << endl;

char msg[256] = "Hello. This is server";

send(newConnection, msg, sizeof(msg), NULL);

Connections[i] = newConnection;

Counter++;

CreateThread(NULL, NULL, (LPTHREAD\_START\_ROUTINE)ClientHandler, (LPVOID)(i), NULL, NULL);

}

}

system("pause");

return 0;

}

Клієнтська частина

#include "pch.h"

#pragma comment(lib, "ws2\_32.lib")

#include <WinSock2.h>

#include <iostream>

#pragma warning(disable: 4996)

using namespace std;

SOCKET Connection;

void ClientHandler() {

char msg[256];

while (true) {

recv(Connection, msg, sizeof(msg), NULL);

cout << msg << endl;

}

}

int main(int argc, char\* argv[])

{

WSADATA wsaData;

WORD DLLVersion = MAKEWORD(2, 1);

if (WSAStartup(DLLVersion, &wsaData) != 0) {

cout << "Library error" << endl;

exit(1);

}

SOCKADDR\_IN addr;

int sizeofaddr = sizeof(addr);

addr.sin\_addr.s\_addr = inet\_addr("127.0.0.1");

addr.sin\_port = htons(1111);

addr.sin\_family = AF\_INET;

Connection = socket(AF\_INET, SOCK\_STREAM, NULL);

if (connect(Connection, (SOCKADDR\*)&addr, sizeof(addr)) != 0) {

cout << "Error. Failed connect to server.\n";

return 1;

}

cout << "Connected\n";

CreateThread(NULL, NULL, (LPTHREAD\_START\_ROUTINE)ClientHandler, NULL, NULL, NULL);

char msg1[256];

while (true) {

cin.getline(msg1, sizeof(msg1));

send(Connection, msg1, sizeof(msg1), NULL);

Sleep(10);

}

system("pause");

return 0;

}