

# Лабораторна робота

## "Взаємодія розподілених процесів через механізм сокетів"

### Зміст:

#### [1. Постановка задачі.](#)

#### [2. Варіанти індивідуальних завдань](#)

В даній лабораторній роботі необхідно освоїти механізм (технологію) сокетів стеку протоколів TCP/IP, зокрема його реалізацію в MS Windows. Індивідуальний варіант роботи полягає в розробці двох програм (клієнта та сервера, які запускаються на різних станціях мережі), розробці протоколу обміну даними між ними та демонстрації роботи програм.

Від програм не вимагається досконалого дизайну та функціональності, - головне - розробка мережевого протоколу та його реалізація з використанням сокетів.

## Постановка задачі

Потрібно:

1. Вивчити роботу із сокетами (принципи, функції підтримки роботи з WinSock, див. теоретичні відомості).
2. Розробити протокол обміну даними для свого варіанту: сценарій обміну даними та його команди (через сокет передається просто повідомлення, а що з ним робити визначають команди).
3. Розробити дві програми, так звані програма-клієнт (далі просто **клієнт**) та програма-сервер (далі просто **сервер**). Ці дві програми, працюючи на різних робочих станціях (вузлах) мережі, мають обмінюватися командами, згідно розробленого автором протоколу, використовуючи механізм сокетів у профілі TCP/IP.
4. Має бути передбачена реакція на помилкові команди.
5. Номер порта призначення вибирати так: №\_порта = 1025 + №\_варіанту.
6. Всі варіанти мають підтримувати команду із ім'ям *Who*, у відповідь на яку сервер дає інформацію про автора та номер і коротку назву варіанту.
7. Клієнт та сервер мають для кожного відісланого/прийнятого сокет-повідомлення робити запис у системний журнал (текстовий файл) про: час дії, команда(и) повідомлення.
8. Формат сокет-повідомлень такий:

Заголовок	Дані
-----------	------

9. Формат заголовка та розміри полів:

<b>Довжина заголовка (в байтах)</b>	<b>Команда1 [; Команда2; ...]</b>
1 байт	до 255 байт

10. Зрозуміло, якщо у заголовку не буде команд, то все одно має бути один байт довжини заголовка (і дорівнювати 1). Як правило, у повідомленні буде одна команда (у [...] необов'язкові елементи). Команда може мати операнди, тоді синтаксис таких команд розробити самостійно.

**У звіті до лабораторної має бути:**

- описаний протокол (структура пакета обміну, команди, формати даних, діаграма станів сценарію пари клієнт/сервер),
- тексти програм (клієнта, сервера),
- тексти файлу протоколу (системного журналу дій програми) для клієнта та для сервера (для звіту блоки однакових команд у журналі доцільно скоротити).

## Варіанти індивідуальних завдань

(Закріплення варіантів за студентом - у таблиці балів)

№ вар.	Варіант
1	<b>Редактор текстових рядків.</b> Після приєднання клієнта, сервер передає йому набір текстових рядків (не менше 5). У клієнті можна змінювати символи, вставляти нові чи видаляти, не порушуючи кількість рядків та їх максимальний розмір. Будь-яка зміна у рядках клієнта (заміна символу, видалення, вставка нового) передаються серверу для збереження.
2	<b>Передача файлів.</b> Клієнт передає серверу текстовий файл (із збереженням його оригінального імені). Сервер зберігає переданий файл в своєму каталозі. Клієнт та сервер мають відображати хід передачі/прийому: кількість переданих/прийнятих байт, відсоток виконання операції та поточну швидкість передачі. Клієнт в кінці передачі має відобразити час виконання операції.
3	<b>Видалена консоль.</b> Клієнт передає на сервер певну кількість команд (командного процесора ОС), які на клієнті можуть вводитися вручну чи братися із текстового файлу. Сервер, після отримання останньої команди, виконує весь пакет команд та відсилає підтвердження дії. (З клієнта краще передавати команди, що не потребують діалогу з оператором (такі як: <i>dir</i> , <i>type</i> , <i>echo</i> ).
4	<b>Обмін репліками .</b> Дві програми на різних хостах дають можливість користувачам обмінюватися репліками. Повідомлення, довші за 20 символів, повинні розбиватися на кілька реплік.

5	<p><b>Електронна "дошка об'яв".</b></p> <p>На сервері вікно із 9 рядків розбите на 3 області - "об'яви" з розміром не більше рядка. З клієнтської програми потрібно мати можливість передавати "об'яви" в одну з областей і в заданий рядок в ній.</p>
6	<p><b>Список файлів.</b></p> <p>Сервер дає можливість клієнтській програмі отримати список файлів і папок поточної директорії сервера. Запит на отримання списку може містити набір символів (фільтр), які мають міститися в іменах файлів/папок.</p>
7	<p><b>Контейнер іменованих об'єктів.</b></p> <p>На сервері зберігаються іменовані об'єкти клієнтів: цілі числа, символьні рядки тощо. Для зберігання/зміни на сервері об'єкта потрібно передати трійку &lt;Ім'я; Тип; Значення&gt;. Для зчитування потрібно лише ім'я, а сервер має дати відповідь трійкою &lt;Ім'я; Тип; Значення&gt;. Клієнт може зчитати, змінити та отримати перелік імен об'єктів.</p>
8	<p><b>Перекладач.</b></p> <p>Клієнт посилає слово на сервер, які вводить користувач. Сервер повертає або словникову статтю, або відмову. На сервері достатньо реалізувати "словник" із якогось десятка слів.</p>
9	<p><b>Калькулятор.</b></p> <p>Сервер реалізує калькулятор на чотири арифметичні операції. З клієнта за одне звернення до сервера передається введена користувачем операція та два її операнди, а сервер повертає результат. В кінці сеансу сервер передає до клієнта: min, max та середнє операндів, кількість виконаних дій.</p>
10	<p><b>Контекстний пошук у файлі(ах).</b></p> <p>На сервері зберігається файл (достатньо лише текстовий). Користувач в клієнті задає контекст (лише із символів клавіатури), клієнт відсилає останній до сервера для виконання контекстного пошуку рядків. Результати пошуку порядково відсилаються до клієнта. Якщо пошук пустий, то сервер все одно відсилає відповідне попередження.</p>
11	<p><b>Програмуємий будильник.</b></p> <p>1) Користувач з клієнта задає інтервал в секундах відображення часу на сервері (за змовчанням це 1 секунда). Сервер в зазначені інтервали передає до клієнта свій час, а клієнт його відображає.</p> <p>2) Користувач з клієнта може виставити до 3-х таймерів (будильників), частина з яких прямого обліку часу, частина зворотнього. Сервер, крім системного часу, передає поточні значення для запущених таймерів та сигнал (чи транспорант) при спрацьовуванні таймера. (Клієнт лише відображає дані сервера, але сам він ніякий час не рахує.)</p>
12	<p><b>Гра "хрестики-нолики".</b></p> <p>Гра "хрестики-нолики" на полі 5x5 клітинок. Один гравець - на сервері, другий - на клієнті. Поле (матрицю символів) до клієнта передає сервер. Сервер робить хід у будь-яку пусту клітинку, стратегія не потрібна, перемогу/програш не відслідковує. Після кожного ходу клієнта сервер передає поле, додавши свій хід. Клієнт в ході гри може її завершити в будь-який момент.</p>
13	<p><b>Підбір пароля.</b></p> <p>Клієнт підбирає пароль, який зберігається на сервері. При запуску сервера задається його пароль. Сервер передає клієнту лише довжину пароля. З боку клієнта має бути можливість ручного підбору користувачем та автоматичного підбору на протязі заданого проміжку часу в секундах. Клієнт відображає</p>

	номер поточної спроби та її вигляд. При вгадуванні пароля процедура автоматичного підбору зупиняється. Якщо був заданий проміжок часу на автобідбір, то сервер повертає кількість виконаних спроб.
14	<b>Порівняння/корекція файлів.</b> Спочатку на клієнті та сервері зберігаються однакові текстові файли (на кілька рядків). На сервері можлива зміна у файлі. Клієнт по команді користувача корегує свій файл згідно змін на сервері. За кожним зверненням клієнта сервер передає лише одну зміну у форматі <i>&lt;Номер_рядка; Нове_значення_рядка&gt;</i> , або інформує про відсутність змін. На сервері у діалозі можна "підказувати" номер рядка, який був змінений.
15	<b>Сортування рядків.</b> Клієнт передає на сервер введені користувачем рядки або зчитані із текстового файлу. Сервер виконує сортування їх та повертає відсортованими до клієнта по одному рядку у сокетному повідомленні. Ключ сортування задається користувачем як стартова позиція у рядку та довжина ключа. Клієнт може окремо отримати від сервера дані про час, витрачений сервером для останнього сортування.
16	<b>Гра "морський бій".</b> Грає лише користувач на клієнті. Сервер відповідає лише про результат попадання/знищення кораблика. Достатньо на сервері мати єдине статичне представлення поля з кораблями. Клієнт в матриці 10x10 повинен передавати координати вистрілу. Сервер (без зайвої графічної реалізації, - достатньо використовувати кілька символів для позначень у матриці із символів) передає матрицю 10x10 із необхідними позначками вистрілів та попадань. Клієнт в ході гри може її завершити гру раніше. Після завершення сеансу сервер до клієнта передає статистику гри: кількість стрільб, кількість попадань та промахів.
17	<b>Реплікація даних.</b> Клієнт і сервер мають однаковий набір змінних (числові, строкові тощо). Ці дані незалежно можуть змінюватися як на сервері, так і на клієнті. Але в певні моменти часу, нехай з ініціативи сервера, виконується синхронізація даних (репліковані дані в розподілених системах - майже завжди, з точністю до моменту синхронізації, співпадаючі дані). Отже, після синхронізації знову в обох програмах однакові дані. При синхронізації кожна з програм передає лише дані (змінні), які були змінені після попередньої синхронізації. Разом із новим значенням потрібно передавати час зміни ("штамп" часу). Якщо однойменні дані в обох програмах були змінені, то за нове значення береться те, що має останній "штамп" часу. Для зміни даних можна використовувати консольний діалог без розробки графічних вікон.
18	<b>Гра "вгадування 4-значного числа".</b> Користувач на клієнті вгадує 4-значне ціле, яке зберігається на сервері. З клієнта передаються 4 цифри, на які сервер дає відповідь із двох цифр: кількість правильних цифр та кількість цифр на своїх місцях. Клієнт в ході гри може її завершити, почати нову гру, завершити сеанс. Клієнт може здатися і тоді сервер розкриває загадане число.. Користувач на клієнті може вибирати спосіб задання спроб: в діалозі вводити самому чи автоматична генерація випадкових 4-х значних чисел в заданій кількості.
19	<b>Порівняння файлів .</b> На клієнті та сервері спочатку зберігаються ідентичні текстові файли (невеликого розміру). З часом файли можуть змінюватися (засобами будь-

	якого текстового редактора), але вони будуть квазіідентичними, якщо: 1) символи верхнього і нижнього регістрів є ідентичними та 2) всі "білі" поля (з одного і більше пропусків) еквівалентні одному пропуску. По команді із клієнта виконується процедура порівняння - клієнт передає свої рядки, користувач на клієнті отримує список рядків, в яких порушена квазіідентичність.
20	<b>Вгадування числа за методом Монте-Карло.</b> На сервері загадується ціле число в діапазоні 0..2000000. Користувач на клієнті задає час на автопідбір числа клієнтом за методом Монте-Карло (генеруванням випадкового значення у потрібному діапазоні). Після завершення інтервалу сервер передає кількість виконаних спроб та, якщо не було вгадування, найбільш близьке число. Окремою командою можна отримати від сервера загадане число.
21	<b>Аналог утиліти ping.</b> Клієнт ping-ує сервер (посилає повідомлення і вимірює час приходу відповіді) з такою частотою, щоб досягнути заданого навантаження на сегмент мережі (виходячи із поточного завантаження LAN). Повинна бути можливість варіювати частотою та розміром сокет-повідомлень. Завантаженість мережі можна контролювати за допомогою "Диспетчера задач".
22	<b>Тест "розмір повідомлення/швидкість".</b> Побудувати (програмно) для поточного завантаження LAN залежність часу на відсилку серії із 10/50/100/1000/10000 сокет-повідомлень розміром 10/100/1000/10000 байт. Сервер, після отримання відповідної серії повідомлень, передає клієнту час для кожної серії.
23	<b>Тестування продуктивності мережі.</b> Клієнт та сервер по чергово обмінюються повідомленнями, розмір яких завжди вибирається випадково в діапазоні від 10 до 50000 байт задану кількість разів (серія вимірювання). Результат вимірювання має таке представлення: сумарна кількість повідомлень, витрачений час, середній розмір повідомлення, середній час на повідомлення, min та max розмір повідомлення в серії.
24	<b>Диспетчер розділяємих змінних.</b> Клієнт і сервер спільно використовують 10 змінних, які зберігаються на сервері. Змінні можна читати та задавати їм нове значення. Клієнт серією спроб повинен поміняти всім змінним значення. Доступ до змінних регулює сервер. Випадковим чином диспетчер вибирає дозвіл чи заборону на зміну значення для кожної із змінних. Якщо змінна заблокована, то сервер посилає відмову клієнту, і той через тайм-аут, вибраний випадково в діапазоні [0.1, 0.3] секунди може повторити спробу (тобто сервер чергу запитів на зміни не веде). Проведіть серію тестів для визначення часу для проведення по 1 зміні значення для кожної змінної, по 10 та по 100 змін. Також потрібно зберігати кількість виконаних спроб для 1, 10 та 100 редагування для кожної змінної. (Нове значення для змінних предметної цінності не становить і може бути будь-яким.)
25	<b>Частота символів.</b> Клієнт отримує від сервера допустиму кількість символів для передачі та передає на сервер множину символів по одному символу в сокетному повідомленні (наприклад, посимвольно кілька рядків текстового файлу). Сервер для прийнятих символів формує таблицю частоти символів і після

	завершення передачі клієнтом передає до клієнта кількість рядків у таблиці частот та саму таблицю по рядку в одному повідомленні.
26	<b>Пошук числа методом дихотомії та золотого перерізу.</b> На сервері задається ціле число в діапазоні $[0, 1000000]$ , яке має знайти клієнт. Клієнт, знаючи діапазон можливих значень, для пошуку використовує (одночасно чи по черзі) методи <i>дихотомії</i> та <i>золотого перерізу</i> . На кожну пропозицію числа клієнтом сервер відповідає чи воно менше, дорівнює, чи більше загаданого числа. Після завершення процедур пошуку клієнт видає звіт про кількість ітерацій для кожного із методів та витрачений час.
27	<b>Відновлення об'єкта.</b> На сервері у прямокутній області відомого для клієнта розміру зберігається певний об'єкт (може бути просто множина точок). Клієнт має $N$ спроб для розпізнавання об'єкта (фігури), це буде правильно вгадана множина точок об'єкта в аналогічній прямокутній області. Вибір точки в області проводити за методом Монте-Карло (вибравши початок координат, наприклад, у нижній лівій вершині, і випадково вибираємо для поточної точки значення ширини та висоти). Точка буде вгадана, якщо вона попадає в заданий окіл правильної точки об'єкта. На клієнті потрібно відобразити отриманий об'єкт. На сервері має бути відображений оригінальний об'єкт. Можна область з зображенням будувати у матриці символів у консольному вікні без графічних об'єктів.
28	<b>Фільтр простих чисел.</b> Клієнт передає на сервер цілі числа по одному у повідомленні, отримані: 1) у діалозі з користувачем, або 2) випадково згенеровані 50 значень. Сервер по кожному числу дає відповідь щодо просте воно чи ні. Клієнт прості числа заносить у вихідний текстовий файл. Після завершення клієнтом передачі чисел сервер передає клієнту зведені дані: 1) кількість оброблених чисел, 2) кількість простих, 3) <i>min</i> та <i>max</i> значення.
29	<b>Переформатування текстових рядків.</b> Клієнт передає на сервер текстові рядки, читаючи їх із підготовленого файлу. Сервер в отриманих рядках всі "білі" поля (це більше ніж один пропуск підряд) скорочує до одного пропуску та переформатовує у рядки до 40 символів не розриваючи слова. Переформатований рядок/рядки повертаються до клієнта та зберігаються ним у вихідному файлі.
30	<b>Синхронізація множин чисел.</b> На клієнті та сервері зберігаються різні множини цілих чисел. Після обміну множинами на сервері та клієнті мають бути однакові множини. Першим передає свою множину клієнт. По кожному числу сервер дає підтвердження про правильність/неправильність числа (наприклад у сервера є обмеження на максимальне значення тощо). Неправильні числа клієнт має видалити із своєї множини, а сервер їх не буде заносити до своєї множини. Після клієнта свою множину передає сервер без значень, які вже є у множині клієнта.
31	<b>Сканер портів.</b> Клієнт сканує порти сервера і дає звіт про відкриті порти на ньому. На сервері має бути реалізована можливість відкривати/закривати певну кількість портів.