

Київський національний університет імені Тараса Шевченка
Факультет комп'ютерних наук та кібернетики
Кафедра системного аналізу та теорії прийняття рішень

Звіт
з лабораторної роботи №2
Взаємодія розподілених процесів через механізм сокетів

Виконав
студент групи К-23

Забровський В.Д.



Прийняв

Махно М.Ф

Київ – 2023

Мета

Освоїти механізм роботи сокетів стеку протоколів TCP/IP. Розробити дві програми(клієнта та сервера), розробити протокол обміну даними між ними.

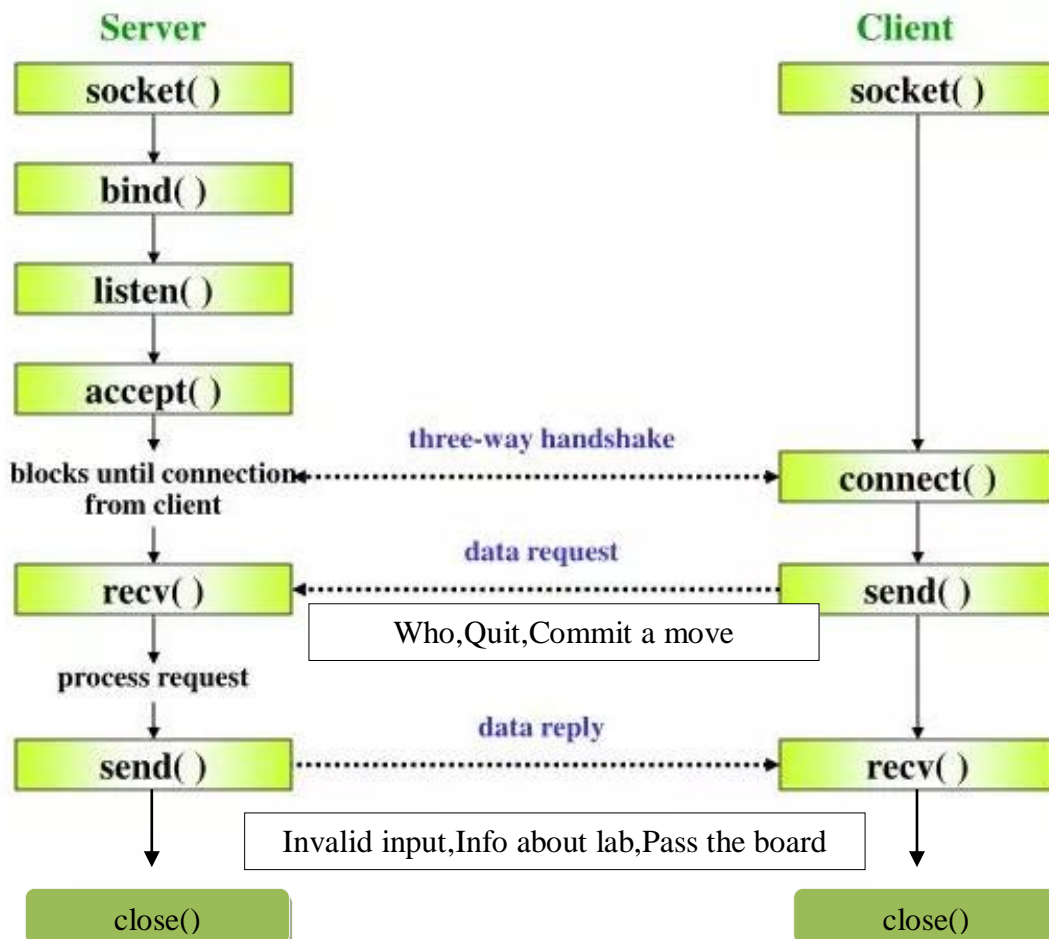
Індивідуальне завдання

Гра "хрестики-нолики"

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.

Опис протоколу

TCP Client-Server Model



Усього клієнту доступні 3 типи команд, які сервер обробляє таким чином:

1)**Who** – сервер дає інформацію про автора та варіант лабораторної роботи.

2)**Quit** – сервер,отримуючи команду, розриває зв'язок із клієнтом.Клієнт відповідно бачить у своєму вікні повідомлення про завершення роботи.

3) **Enter row and column** – сервер обробляє хід користувача: якщо він валідний,то робить свій хід та передає дошку клієнту,очікуючи від нього наступний хід.Інакше сервер відсилає повідомлення про неправильний хід та очікує від гравця наступну команду.

Код сервера:

```
import socket
import numpy as np
import logging
from datetime import datetime

logging.basicConfig(filename='server_log.txt',filemode='w', level=logging.INFO,
format='% (message) s')

def create_board():
    return np.full((5, 5), '-')

def check_make_move(board, row, col):
    if 0 <= row <= 4 and 0 <= col <= 4 and board[row][col] == '-':
        return True
    else:
        return False

def place_random_O(board):
    empty_cells = [(i, j) for i in range(5) for j in range(5) if board[i][j] == '-']
    if empty_cells:
        index = np.random.choice(len(empty_cells))
        row, col = empty_cells[index]

        board[row][col] = 'O'

def server_program():
    host = socket.gethostname()
    port = 1037

    server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    server_socket.bind((host, port))
    server_socket.listen(1)
    print("Waiting for connection from the client...")
    conn, address = server_socket.accept()
    print("Connection from: " + str(address))

    board = create_board()

    while True:
        data = conn.recv(256).decode()

        current_time = datetime.now().strftime('%Y-%m-%d - %H:%M:%S')
```

```

        if data.lower().strip() == "who":
            user_info = "Author: Vladislav Zabrovsky, Variant 12 - Tic-Tac-Toe
game"
            conn.send(user_info.encode())
            print("User asked info about labwork")
            logging.info(f"{current_time} - INFO - User asked info about
labwork")

        elif data.lower().strip() == "quit":
            print("Client quit the game")
            logging.info(f"{current_time} - INFO - Client quit the game")
            break

        else:
            try:
                row, col = map(int, data.split())
                if check_make_move(board, row, col):
                    print("User committed a move")
                    logging.info(f"{current_time} - INFO - User committed a
move")

                    board[row][col] = 'X'

                    place_random_O(board)

                    conn.send(np.array_str(board).encode())
                    print("Server committed a move")
                    logging.info(f"{current_time} - INFO - Server committed a
move")

                else:
                    response = "Invalid move. Cell is already occupied or
doesn't exist."
                    conn.send(response.encode())
                    print("User committed an invalid move")
                    logging.info(f"{current_time} - INFO - User committed an
invalid move")

            except ValueError:
                response = "Invalid input. Please enter row and column (e.g., '2
3')."
                conn.send(response.encode())
                print("User committed an invalid move")
                logging.info(f"{current_time} - INFO - User committed an invalid
move")

    conn.close()

if __name__ == '__main__':
    try:
        server_program()
    except Exception as e:
        print(f'Oooops...., smth went wrong - {e}')

```

Код клієнта:

```
import socket
import logging
from datetime import datetime

logging.basicConfig(filename='client_log.txt', filemode='w', level=logging.INFO,
format='% (message) s')

def client_program():
    host = socket.gethostname()
    port = 1037

    client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    client_socket.connect((host, port))

    while True:
        message = input("Enter row and column (e.g., '2 3'), Who, or Quit:
").strip()

        client_socket.send(message.encode())
        data = client_socket.recv(256).decode()

        current_time = datetime.now().strftime('%Y-%m-%d %H:%M:%S')
        logging.info(f"{current_time} - INFO - Sent: {message}")
        logging.info(f"{current_time} - INFO - Received: {data}")
        print(data)
        if message.lower() == 'quit':
            print('End of the game')
            logging.info(f"{current_time} - INFO - End of the game")
            break

    client_socket.close()

if __name__ == '__main__':
    try:
        client_program()
    except Exception as e:
        print(f'Ooops...., smth went wrong - {e}')
```

Системні журнали сервера та клієнта:

```
[2023-10-27 - 22:33:20] - INFO - User committed a move
[2023-10-27 - 22:33:20] - INFO - Server committed a move
[2023-10-27 - 22:33:23] - INFO - User committed a move
[2023-10-27 - 22:33:23] - INFO - Server committed a move
[2023-10-27 - 22:33:30] - INFO - User committed an invalid move
[2023-10-27 - 22:33:38] - INFO - User committed an invalid move
[2023-10-27 - 22:33:40] - INFO - User asked info about labwork
[2023-10-27 - 22:33:43] - INFO - Client quit the game
```

Журнал сервера

```
[2023-10-27] 22:33:20 - INFO - Sent: 2 3
[2023-10-27] 22:33:20 - INFO - Received: [['-' 'O' '-' '-' '-']]
[['-' '-' '-' '-' '-']]
[['-' '-' '-' 'X' '-']]
[['-' '-' '-' '-' '-']]
[['-' '-' '-' '-' '-']]
[2023-10-27] 22:33:23 - INFO - Sent: 3 2
[2023-10-27] 22:33:23 - INFO - Received: [['-' 'O' '-' '-' '-']]
[['-' '-' '-' '-' '-']]
[['-' 'O' '-' 'X' '-']]
[['-' '-' 'X' '-' '-']]
[['-' '-' '-' '-' '-']]
[2023-10-27] 22:33:30 - INFO - Sent: absfki
[2023-10-27] 22:33:30 - INFO - Received: Invalid input. Please enter row and column (e.g., '2 3').
[2023-10-27] 22:33:38 - INFO - Sent: 45 86
[2023-10-27] 22:33:38 - INFO - Received: Invalid move. Cell is already occupied or doesn't exist.
[2023-10-27] 22:33:40 - INFO - Sent: Who
[2023-10-27] 22:33:40 - INFO - Received: Author: Vladislav Zabrovsky, Variant 12 - Tic-Tac-Toe game
[2023-10-27] 22:33:43 - INFO - Sent: Quit
[2023-10-27] 22:33:43 - INFO - Received:
[2023-10-27] 22:33:43 - INFO - End of the game
```

Журнал клієнта