

ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені ІВАНА ФРАНКА

Факультет прикладної математики та інформатики

Комп'ютерні інформаційні мережі

ЛАБОРАТОРНА РОБОТА №10

Виконав :

Ст Гуменюк С. А.

ПМІ -33

Тема: Програмування сокетів

Завдання: Використовуючи бібліотеку для роботи з сокетами, на будь-якій мові програмування написати клієнт-серверний застосунок, де клієнтська програма відправляє Login і Passw та отримує від серверного додатка вітання з вашим Ім'ям та Прізвищем, наприклад "Вітаю ..."

Реалізувати вибір режиму роботи сервера під час його запуску — послідовна/паралельна обробка запитів.

Хід роботи

Код сервера:

```
1  const net = require('net');
2  const { Worker, isMainThread, parentPort } = require('worker_threads');
3
4  const VALID_CREDENTIALS = { 'user1': 'password1', 'user2': 'password2' };
5
6  function authenticate(credentials) {
7    return VALID_CREDENTIALS[credentials.login] === credentials.password
8      ? `Hi Stanislav! You are in system.\n`
9      : `Error: Wrong login/password combination.\n`;
10 }
11
12 if (isMainThread) {
13   const parallel = process.argv.includes('--parallel');
14
15   net.createServer(socket => {
16     let buffer = '';
17
18     if (!parallel) {
19       socket.on('data', data => {
20         buffer += data.toString();
21         if (buffer.includes('\n')) {
22           socket.end(authenticate(JSON.parse(buffer)));
23         }
24       });
25       return;
26     }
27
28     const worker = new Worker(__filename);
29     socket.on('data', data => {
30       buffer += data.toString();
31       if (buffer.includes('\n')) {
32         worker.postMessage(buffer);
33         buffer = '';
34       }
35     });
36     worker.on('message', response => socket.end(response));
37     socket.on('end', () => worker.terminate());
38
39   }).listen(3000, () => console.log(`Server running on port 3000 in ${parallel ? 'parallel' : 'sequential'} mode`));
40 } else {
41   parentPort.on('message', data => {
42     try {
43       parentPort.postMessage(authenticate(JSON.parse(data)));
44     } catch (err) {
45       parentPort.postMessage(`Error: ${err.message}\n`);
46     }
47   });
48 }
```

Код клієнтського застосунку:

```
1  const net = require('net');
2
3  new net.Socket()
4    .connect(3000, 'localhost', function() {
5      this.write(JSON.stringify({login: 'user1', password: 'password1'}) + '\n');
6    })
7    .on('data', data => console.log('Response:', data.toString()))
8    .on('close', () => console.log('Connection closed'))
9    .on('error', err => console.error('Error:', err.message));
```

Про режим роботи сервера інформується при запуску:

```
sonorma@hellcat > ~/.../CIN/lab10 > ʘ main ● node server.js  
Server running on port 3000 in sequential mode
```

```
sonorma@hellcat > ~/.../CIN/lab10 > ʘ main ● node server.js --parallel  
Server running on port 3000 in parallel mode
```

Результат виконання клієнту при запусченому сервері:

```
sonorma@hellcat > ~/.../CIN/lab10 > ʘ main ● node client.js  
Response: Hi Stanislav! You are in system.  
  
Connection closed
```

Помилка при неможливості підключення до сервера

```
sonorma@hellcat > ~/.../CIN/lab10 > ʘ main ● node client.js  
Error: connect ECONNREFUSED ::1:3000  
Connection closed
```