# ЛАБОРАТОРНА РАБОТА № 4 Дослідження роботи з потоковими сокетами в режимі опиту

## Мета роботи: вивчити роботу з потоковими сокетами в режимі опиту. Створити сервер котрий буде відповідати на запити клієнтів.

* 1. Теоретичні відомості

Socket API був вперше реалізований в операційній системі UNIX. Зараз цей програмний інтерфейс доступний практично в будь-якій операційній системі. Хоча всі реалізації чимось відрізняються один від одного, основний набір функцій у них збігається. Спочатку сокети використовувалися в програмах на C/C++, але в даний час вони є майже в всіх нових мовах програмування (Perl, С#, Java та ін.).

Сокети надають дуже потужний і гнучкий механізм взаємодії між процесами (IPC). Вони можуть використовуватися для організації взаємодії програм на одному комп'ютері, по локальній мережі або через Інтернет, що дозволяє вам створювати розподілені додатки різної складності. Крім того, з їх допомогою можна організувати взаємодію з програмами, що працюють під управлінням інших операційних систем.

Сокети підтримують багато стандартних мережевих протоколів (конкретний їх список залежить від реалізації) і надають уніфікований інтерфейс для роботи з ними. Найбільш часто сокети використовуються для роботи в IP-мережах.

Сокети, незалежно від виду, поділяються на три типи: потокові, сирі і дейтаграмні. Потокові сокети працюють з установкою з'єднання, забезпечуючи надійну ідентифікацію обох сторін і гарантують цілісність і успішність доставки даних, спираючись на протокол TCP. Дейтаграмні сокети працюють без встановлення з'єднання і не забезпечують ні ідентифікації відправника, ні контролю успішності доставки даних, зате вони швидше потокових, спираючись на протокол UDP. Сирі сокети, вони надають можливість ручного формування TCP \ IP-пакетів.

Також існує 2 види сокетів:

- синхронні – затримують управління на час виконання операції;

- асинхронні – повертають управління, але продовжують виконувати роботу в фоні та після закінчення повідомляють про це.

**Хід роботи**

1. Вихідний код серверу що повертає клієнту число та парність числа зображено в лістингу 1. Вихідний код клієнта що надсилає серверу поточний час зображено в лістингу 2.

Лістинг 1 – Файл серверу

import socket  
import threading  
server = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)  
server.bind(("127.0.0.1", 5555))  
server.listen(5)  
users = []  
  
def send\_message(data, user\_self, sec):  
 for user in users:  
 if(sec % 2 == 0):  
 result = " -> (Число є парним)"  
 data = data.decode("utf-8")  
 tx = data + result  
 tx = tx.encode("utf-8")  
 user.send(tx)  
 else:  
 result = " -> (Число є непарним)"  
 data = data.decode("utf-8")  
 tx = data + result  
 tx = tx.encode("utf-8")  
 user.send(tx)  
  
def listen\_user(user):  
 while True:  
 try:  
 data = user.recv(2048)  
 text = data.decode("utf-8")  
 sec = text[17:19]  
 print(sec)  
 sec\_int = int(sec)  
 send\_message(data, user, sec\_int)  
 except:  
 user.close()  
 for u in users:  
 if(u == user):  
 users.remove(u)  
def start():  
 while True:  
 socket\_user, adress\_user = server.accept()  
 print(socket\_user, " is connected")  
 users.append(socket\_user)  
 thread\_for\_user = threading.Thread(target=listen\_user, args= (socket\_user,))  
 thread\_for\_user.start()  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 start()

Лістинг 2 – Файл клієнта

import socket  
import threading  
import datetime  
import time  
client = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)  
client.connect(("127.0.0.1", 5555))  
  
def listen\_server():  
 while True:  
 data = client.recv(2048)  
 print(data.decode("utf-8"))  
  
def send\_data():  
 thread\_for\_listen = threading.Thread(target=listen\_server)  
 thread\_for\_listen.start()  
 while True:  
 now = datetime.datetime.now()  
 client.send(str(now).encode("utf-8"))  
 time.sleep(1)  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 send\_data()

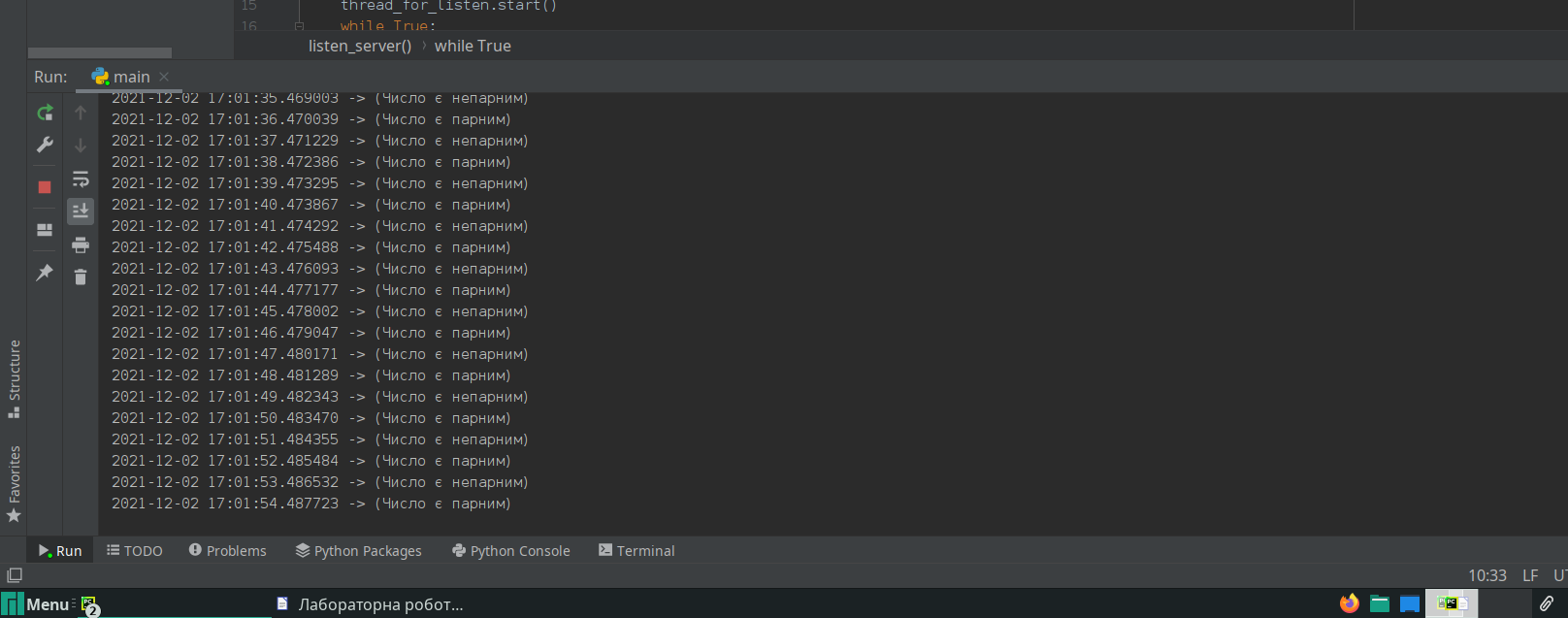
Результат виконання програми зображено на рисунку 1. На рисунку видно повернений час та перевірка парності що надійшла клієнту від сервера.

Рисунок 1 - Результат виконання програми що повертає парність секунд

## **Висновки**: вивчив роботу з потоковими сокетами в режимі опиту. Створив сервер котрий відповідав на запити клієнта.