

Київський національний університет
імені Т.Шевченка

Звіт

до лабораторної роботи
на тему:

***«Взаємодія розподілених процесів через
механізм сокетів»***

***Студента другого курсу
Групи К-25
Факультету комп'ютерних наук
та кібернетики
Федорича Андрія***

Київ-2022

Мета

Метою даної лабораторної роботи є розробка клієнт-серверної програми, що дозволяє користувачу відправляти математичні вирази до програми-сервера, які будуть повертатися у обчисленому вигляді(якщо це можливо), а також при завершенні процесу взаємодії користувач може отримати коротку статистику, щодо своїх запитів.

Основні принципи виконання роботи

Варіант 9

Сервер реалізує калькулятор арифметичних операцій (+, -, *, /, ^, %). З клієнта за одне звернення до сервера передається введений користувачем вираз, або сукупність виразів записаних через “;”. В кінці сеансу сервер повертає до клієнта min, max, average обчислених виразів та кількість виконаних дій.

Джерело:

Код програми та лог файли можна знайти за посиланням:

<https://github.com/StopFuture/ServerCalculator>

Ця ж інформація буде прикріплена, як додаток до звіту.

Реалізація:

- Робота виконана мовою програмування **Python**.
- Комунікація клієнта та сервера відбувається через консоль.
- Для створення та підтримки роботи сокетів використано вбудовану бібліотеку **socket**.
- Використано **блокуючі(синхронні)** сокети.
- Водночас до сервера може бути під'єднаний тільки **один** клієнт.
- Для ведення log файлів клієнта та сервера застосовано модуль **logging**.

Структура пакету обміну

Заголовок	Дані: Команда1; Команда2; ...
1 байт	до 255 байт

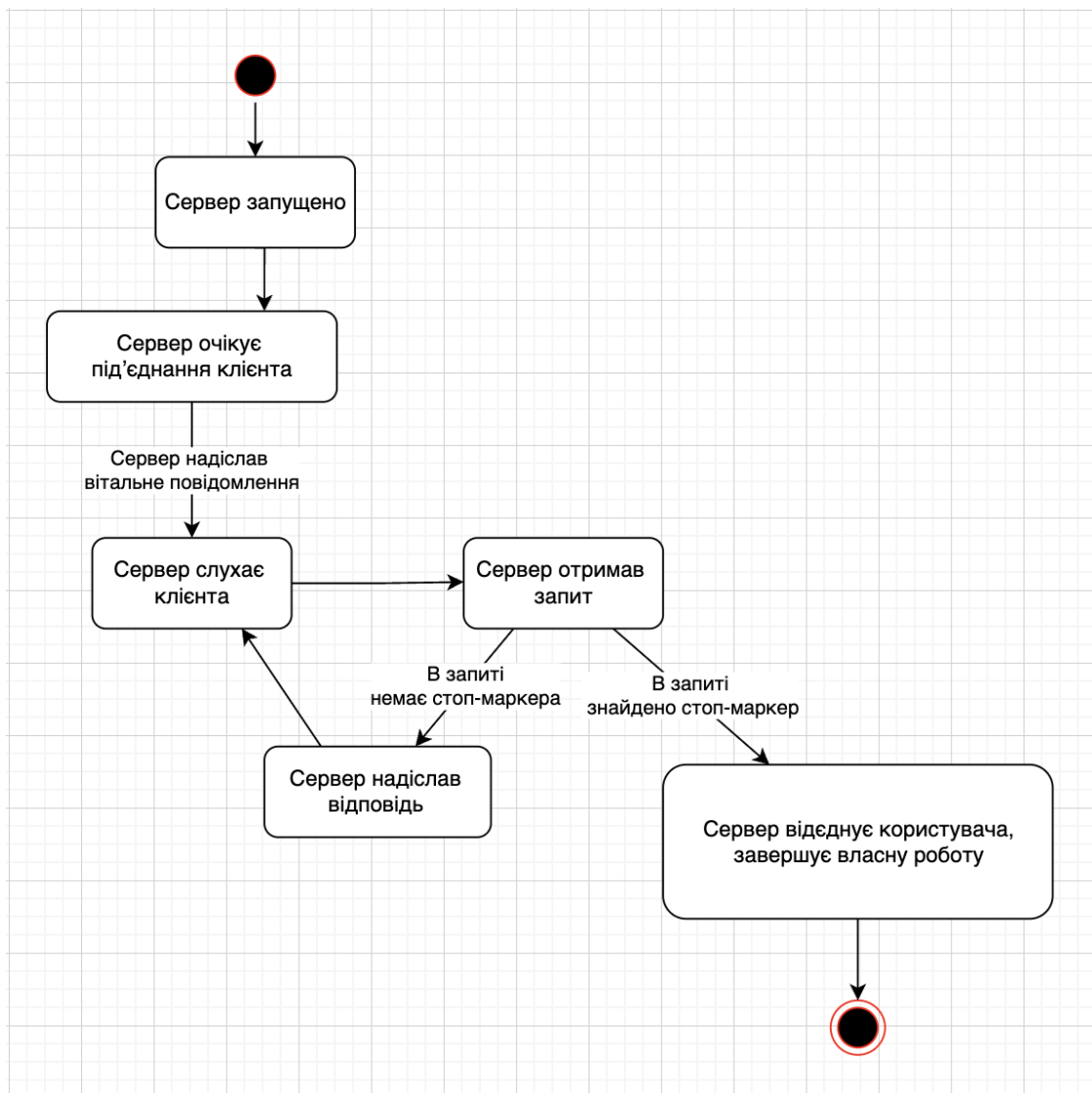
Особливості:

- Заголовок “=” завжди вказаний за клієнта.
- Всі звернення та відповіді передаються у вигляді масиву байтів згідно кодуванню **UTF-8**, що отримуються за рахунок конвертації рядків.
- Всі команди повинні бути розподілені “;”, двокрапка після останньої команди не є обов’язковою.
- Будь-який арифметичний вираз, що складається із дозволених символів(цифри та арифметичні операції) вважається командою.
- Якщо вираз не може бути обчисленим, користувач отримує повідомлення про помилковий ввід.
- Команда “who” повертає значення інформацію про розробника, а саме “Fedorych Andrii, K-25, V9, Calculator”, може бути написана в будь-якому регістрі.

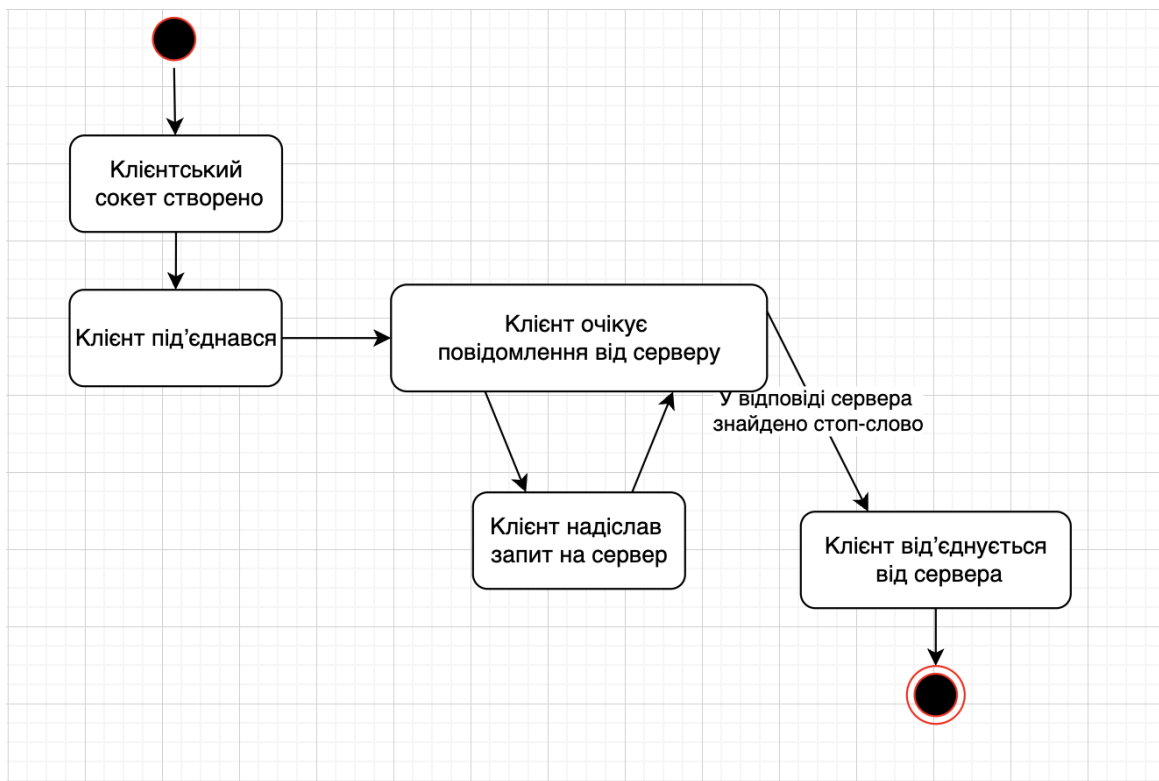
- Команда “stop” повертає результати сесії(min, max, average обчислених виразів та кількість виконаних дій) та завершує сесію.
- Порожній рядок є відповіддю на команду із 0 байт.
- Всі команди, що будуть введені у складному запиті до сервера, але написані після команди “stop” будуть проігноровані.

Діаграми станів

1. Сервер:



2. Клієнт:



Висновок

Отже, сокет це програмний інтерфейс для забезпечення обміну між процесами, він представляє собою абстрактну точку мережевого з'єднання. Кожен сокет, що використовується, має тип і асоційований з ним процес.

Сокети мають стандартне API (Berkeley sockets) та нативно реалізовані в більшості операційних системах. Завдяки такому стандарту сокети можуть слугувати засобами комунікації, між різними ОС та також не залежать від реалізації самої програми.

Додаток

Код програми Сервера:

```
from socket import *
import logging

class Server:
    def __init__(self):
        user = None
        try:
            HOST = '192.168.1.38'
            PORT = 1025 + 9
            self.utf = 'utf-8'
            self.cnt = 0
            self.mx = -float("inf")
            self.mn = float("inf")
            self.sm = 0

            self.server = socket(
                AF_INET, SOCK_STREAM,
            )

            logging.getLogger("Server")
            logging.basicConfig(filename="server_log.log",
                                level=logging.INFO,
                                format='%(asctime)s - %(levelname)s -
%(message)s',
                                filemode='a',
                                )
```

```
logging.info("Server launched")

self.try_connection(HOST, PORT)
print(self.server.getblocking())

user, addr = self.server.accept()

print('Connected:', addr)
logging.info(f'Connected to a client {addr}')
user.send(self.start_msg().encode(self.utf))
logging.info("Sent info")
print('Listening...')
logging.info('Listening to the client')

try:
    while True:
        data = user.recv(256).decode(self.utf)
        logging.info(f'Received message: {data}')
        print(f'Received message: {data}')
        data = data[1:]

        try:
            ans = "=" + str(self.process_msg(data))
            if not ans:
                break
            print(f'Send answer: {ans}')
            if "stop" in ans:
                ans += "\n" + self.final_msg()
            user.send(ans.encode(self.utf))
```

```

        logging.info(
            f'Sent answer: \n{ans}\n ')

        logging.info("Stop marker founded.")
        logging.info("Connection closed.")
        break

    user.send(ans.encode(self.utf))
    logging.info(f'Sent answer: \n{ans}')

except Exception as exp:
    print("Exception: ", exp)
    break

except Exception as exp:
    print("Error", exp)
    logging.info("Error. Connection closed.")
    user.send("stop".encode(self.utf))
finally:
    try:
        user.close()
    except Exception as e:
        print(f'Exception: {e}')
        logging.warning(f'Exception: {e}')
    print("Connection closed.")
    logging.info("Server stopped!\n" + "-" * 20 + "\n")

def process_msg(self, data):

    char_set = set(list(data))

```



```

if len(char_set) > 0:
    ans = ""
    data = data.split(";")
    for i, req in enumerate(data):
        if req.strip().lower() == "who":
            ans += self.who() + ";\n "
        elif req.strip().lower() == "stop":
            ans += "stop;\n "
        return ans
    elif req != "":

        ans += self.calculate(req) + ";\n "

    elif i != len(data) - 1:
        ans += "Empty command;\n "
    return ans
else:
    return "Empty command;\n "

```

```

def calculate(self, data):
    st = set(list(data))
    for char in st:
        if char not in "0123456789*-+/%.^ ":
            return "Wrong Input"
    try:
        ans = str(eval(data))
        self.cnt += 1
        self.sm += float(ans)
        self.mx = max(self.mx, float(ans))

```

```

        self.mn = min(self.mn, float(ans))
    except ZeroDivisionError:
        return "Zero Division Error"
    return ans

def try_connection(self, HOST, PORT):
    try:
        self.server.bind((HOST, PORT))
    except OSError:
        print("Host is used.")
        logging.warning('HOST IS USED.')
        exit()
    self.server.listen(1)

def final_msg(self):
    if self.cnt != 0:
        x = self.sm / self.cnt
        return f'Results of session:\n\tcount = {self.cnt};' \
            f'\n\tmin = {self.mn};\n\tmax = {self.mx};\n\taverage = {x};\n\t'
    else:
        return f'Results of session:\n\tcount = {0};' \
            f'\n\tmin = {None};\n\tmax = {None};\n\taverage = {None};\n\t'

    @staticmethod
    def start_msg():
        return "Write a command or sequence of commands through ';' that will be sent to the server.\n" \
            "For example: '5 + 13; 41/2' and so on.\n" \

```

"The words 'who' and 'stop' can be used as a command, "\
"to get information about the author or stop the session"

```
@staticmethod
```

```
def who():
```

```
    return 'Fedorych Andrii, K-25, V9, Calculator'
```

```
def main():
```

```
    Server()
```

```
if __name__ == "__main__":
```

```
    main()
```

Код програми Клієнта:

```
from socket import *
```

```
import logging
```

```
class Client:
```

```
    def __init__(self):
```

```
        try:
```

```
            HOST = '192.168.1.38'
```

```
            PORT = 1025 + 9
```

```
            logging.getLogger("Client")
```

```
            logging.basicConfig(filename="client_log.log",
```

```
                                level=logging.INFO,
```

```
                                format='%(asctime)s - %(levelname)s -
```

```
%(message)s',
```

```
        filemode='w',  
    )
```

```
logging.info("Client socket created")
```

```
client = socket(  
    AF_INET, SOCK_STREAM,  
)  
# print(client.getblocking())
```

```
client.connect((HOST, PORT))  
logging.info(f'Connected to a server {HOST}')
```

```
data = client.recv(256).decode('utf-8')  
print(data)
```

```
while True:
```

```
    msg = "=" + input('Send request:\n=')  
    client.send(msg.encode('utf-8'))
```

```
logging.info(f'Sent request: {msg}')
```

```
data = client.recv(256).decode('utf-8')  
logging.info(f'Received answer: {data}')
```

```
print('Answer:\n ', data)
```

```
if "stop" in data:
```

```
    logging.info(f'Connection closed')  
    client.close()  
    break
```

```
print("Connection closed.")

except Exception as exp:
    logging.exception(f'Happened exception: {exp}. The server may
be down.')
    print(f'Happened exception: {exp}. The server may be down.')

def main():
    Client()

if __name__ == "__main__":
    main()
```

Лог-файл Сервера:

```
2022-11-11 12:04:45,446 - INFO - Server launched
2022-11-11 12:04:48,033 - INFO - Connected to a client ('192.168.1.38',
58384)
2022-11-11 12:04:48,033 - INFO - Sent info
2022-11-11 12:04:48,033 - INFO - Listening to the client
2022-11-11 12:04:51,576 - INFO - Received message: =12+1;
2022-11-11 12:04:51,576 - INFO - Sent answer:
=13;

2022-11-11 12:04:54,341 - INFO - Received message: =11-1;
2022-11-11 12:04:54,342 - INFO - Sent answer:
=10;

2022-11-11 12:04:57,441 - INFO - Received message: =34 - 1;
2022-11-11 12:04:57,441 - INFO - Sent answer:
=33;
```

2022-11-11 12:04:59,904 - INFO - Received message: =sfs-

2022-11-11 12:04:59,905 - INFO - Sent answer:

=Wrong Input;

2022-11-11 12:05:01,558 - INFO - Received message: =stop

2022-11-11 12:05:01,560 - INFO - Send answer:

=stop;

Results of session:

count = 3;

min = 10.0;

max = 33.0;

average = 18.666666666666668;

2022-11-11 12:05:01,560 - INFO - Stop marker founded.

2022-11-11 12:05:01,560 - INFO - Connection closed.

2022-11-11 12:05:01,560 - INFO - Server stopped!

2022-11-11 12:16:28,065 - INFO - Server launched

2022-11-11 12:16:30,780 - INFO - Connected to a client ('192.168.1.38',
58444)

2022-11-11 12:16:30,780 - INFO - Sent info

2022-11-11 12:16:30,780 - INFO - Listening to the client

2022-11-11 12:16:42,590 - INFO - Received message: =13/2;

2022-11-11 12:16:42,591 - INFO - Sent answer:

=6.5;

2022-11-11 12:16:51,359 - INFO - Received message: =13/5;23+2;

2022-11-11 12:16:51,359 - INFO - Sent answer:

=2.6;

25;

2022-11-11 12:16:54,153 - INFO - Received message: =a+b

2022-11-11 12:16:54,153 - INFO - Sent answer:

=Wrong Input;

2022-11-11 12:17:11,817 - INFO - Received message: =2**3

2022-11-11 12:17:11,819 - INFO - Sent answer:

=8;

2022-11-11 12:17:31,331 - INFO - Received message: =who

2022-11-11 12:17:31,333 - INFO - Sent answer:

=Fedorych Andrii, K-25, V9, Calculator;

2022-11-11 12:18:29,532 - INFO - Received message:

=24+4;who;stop;23+1;

2022-11-11 12:18:29,533 - INFO - Send answer:

=28;

Fedorych Andrii, K-25, V9, Calculator;

stop;

Results of session:

count = 5;

min = 2.6;

max = 28.0;

average = 14.02;

2022-11-11 12:18:29,534 - INFO - Stop marker founded.

2022-11-11 12:18:29,534 - INFO - Connection closed.

2022-11-11 12:18:29,534 - INFO - Server stopped!

2022-11-11 12:20:10,761 - INFO - Server launched

2022-11-11 12:20:14,476 - INFO - Connected to a client ('192.168.1.38',
58460)

2022-11-11 12:20:14,476 - INFO - Sent info

2022-11-11 12:20:14,476 - INFO - Listening to the client

2022-11-11 12:20:18,418 - INFO - Received message: =stop

2022-11-11 12:20:18,418 - INFO - Send answer:

=stop;

Results of session:

count = 0;

min = None;

max = None;

average = None;

2022-11-11 12:20:18,418 - INFO - Stop marker founded.

2022-11-11 12:20:18,418 - INFO - Connection closed.

2022-11-11 12:20:18,419 - INFO - Server stopped!

2022-11-11 12:20:30,071 - INFO - Server launched

2022-11-11 12:20:30,071 - WARNING - HOST IS USED.

2022-11-11 12:20:30,072 - WARNING - Exception: 'NoneType' object has no attribute 'close'

2022-11-11 12:20:30,072 - INFO - Server stopped!

2022-11-11 12:20:41,563 - INFO - Server launched

2022-11-11 12:20:41,563 - WARNING - HOST IS USED.

2022-11-11 12:20:41,563 - WARNING - Exception: 'NoneType' object has no attribute 'close'

2022-11-11 12:20:41,563 - INFO - Server stopped!

2022-11-11 12:23:29,490 - INFO - Server launched

2022-11-11 12:27:28,586 - INFO - Connected to a client ('192.168.1.38', 58488)

2022-11-11 12:27:28,587 - INFO - Sent info

2022-11-11 12:27:28,587 - INFO - Listening to the client

2022-11-11 12:27:35,033 - INFO - Received message: =34(1+1)

2022-11-11 12:27:35,034 - INFO - Sent answer:

=Wrong Input;

2022-11-11 12:27:40,303 - INFO - Received message: =43-1

2022-11-11 12:27:40,303 - INFO - Sent answer:

=42;

2022-11-11 12:27:42,951 - INFO - Received message: =23+4

2022-11-11 12:27:42,951 - INFO - Sent answer:

=27;

2022-11-11 12:27:47,647 - INFO - Received message: = 544+ 1

2022-11-11 12:27:47,647 - INFO - Sent answer:
=545;

2022-11-11 12:27:49,945 - INFO - Received message: =stop
2022-11-11 12:27:49,946 - INFO - Send answer:
=stop;

Results of session:

count = 3;
min = 27.0;
max = 545.0;
average = 204.66666666666666;

2022-11-11 12:27:49,946 - INFO - Stop marker founded.
2022-11-11 12:27:49,946 - INFO - Connection closed.
2022-11-11 12:27:49,946 - INFO - Server stopped!

Лог-файл Клієнта:

2022-11-11 12:27:28,585 - INFO - Client socket created
2022-11-11 12:27:28,585 - INFO - Connected to a server 192.168.1.38
2022-11-11 12:27:35,033 - INFO - Sent request: =34(1+1)
2022-11-11 12:27:35,034 - INFO - Received answer: =Wrong Input;

2022-11-11 12:27:40,302 - INFO - Sent request: =43-1
2022-11-11 12:27:40,303 - INFO - Received answer: =42;

2022-11-11 12:27:42,951 - INFO - Sent request: =23+4
2022-11-11 12:27:42,951 - INFO - Received answer: =27;

2022-11-11 12:27:47,647 - INFO - Sent request: = 544+ 1
2022-11-11 12:27:47,647 - INFO - Received answer: =545;

2022-11-11 12:27:49,945 - INFO - Sent request: =stop
2022-11-11 12:27:49,946 - INFO - Received answer: =stop;

Results of session:

count = 3;
min = 27.0;
max = 545.0;
average = 204.66666666666666;

2022-11-11 12:27:49,946 - INFO - Connection closed