

# Proiect

Realizat de:

- Lăcătuș Paul-Lucian
- Gajdos-Rusu Radu-Gabriel

## Scurta descriere

Programul realizează o conexiune între server și partea de client, utilizând protocolul TCP (Transmission Control Protocol), care este un protocol de transport orientat spre conexiune ce asigură livrarea precisă și în ordine a datelor între două dispozitive într-o rețea. Clientul introduce o expresie algebrică (ca șir de caractere), iar prin intermediul funcției `eval()` serverul returnează rezultatul expresiei.

Partea de sever - Python3 - Ubuntu

Partea de client - php - Windows

## Partea de server - Realizarea conexiunii

```
# Configurare server
HOST = '0.0.0.0' # Ascultă pe toate interfețelele de rețea disponibile
PORT = 55555

# Crearea unui socket
server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server_socket.bind((HOST, PORT))
server_socket.listen(1)

print("Serverul este în așteptare...")
```

- Linia de cod `HOST = '0.0.0.0'` stabilește adresa pe care serverul va asculta.
- Linia `PORT = 55555` specifică numărul portului pe care serverul va aștepta conexiuni.
- `server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)` creează un socket de tip TCP (Transmission Control Protocol) pentru comunicare pe rețea.
- `server_socket.bind((HOST, PORT))` leagă socketul la adresa și portul specificate pentru a putea asculta conexiuni.
- `server_socket.listen(1)` stabilește o singură conexiune în așteptare.

## Partea de client- Realizarea conexiunii

```

$host = "127.0.0.1";
$port = 55555;

// Crearea socket-ului
$client_socket = socket_create(AF_INET, SOCK_STREAM, SOL_TCP);

if (!$client_socket) {
    die("Imposibil de creat socket\n");
}

if (!socket_connect($client_socket, $host, $port)) {
    die("Imposibil de conectat la server\n");
}

echo "Conectat la server\n";

```

- `$host = "127.0.0.1";` specifică adresa IP a serverului (localhost in cazul nostru).
- `$port = 55555;` specifică portul pe care clientul se conectează.
- `$client_socket = socket_create(AF_INET, SOCK_STREAM, SOL_TCP);` : Se creează un socket pentru comunicare TCP.
- `if (!$client_socket) { die("Imposibil de creat socket\n"); }` verifică crearea socketului.
- `if (!socket_connect($client_socket, $host, $port)) { die("Imposibil de conectat la server\n"); }` verifică conexiunea la server.

## Comunicarea dintre cele doua părți

Toata partea de comunicare se afla în interiorul unei bucle while

```

#Server
client_socket.send("Introduceți o expresie aritmetică (de exemplu, 2+2, 5*3, etc.): ".encode('utf-8'))

```

Primul mesaj catre client.

```

#Client
$prompt = socket_read($client_socket, 1024);
echo $prompt;

$expression = readline();
socket_write($client_socket, $expression, strlen($expression));

```

Clientul primește mesajul de la server și trimite înapoi expresia introdusă de la tastatură.

```
#Server
data = client_socket.recv(1024)

expression = data.decode('utf-8')

try:
    result = str(eval(expression))
    response = f"Rezultat: {result}\n"
except Exception as e:
    response = f"Eroare: {str(e)}\n"

client_socket.send(response.encode('utf-8'))
```

Serverul primește expresia și încearcă să o evalueze cu ajutorul funcției predefinite din python de `eval()`

Am folosit try și catch, în caz că expresia nu este compatibilă cu funcția `eval()` atunci serverul trimite spre client eroarea.

În caz că `eval()` a funcționat cu succes serverul returnează rezultatul expresiei.

```
#Client
$response = socket_read($client_socket, 1024);
echo $response;
```

Clientul afișează răspunsul primit de la server, fie eroare sau rezultatul expresiei.

## Exemple de rulare a programului

### caz functional

```
C:\Users\Radu\Desktop>php client.php
Conectat la server
Introduceți o expresie aritmetică (de exemplu, 2+2, 5*3, etc.): 2*2
Rezultat: 4
Introduceți o expresie aritmetică (de exemplu, 2+2, 5*3, etc.): 2+2
Rezultat: 4
Introduceți o expresie aritmetică (de exemplu, 2+2, 5*3, etc.): 4/2
Rezultat: 2.0
Introduceți o expresie aritmetică (de exemplu, 2+2, 5*3, etc.): 4-2
Rezultat: 2
```

### caz de eroare

```
C:\Users\Radu\Desktop>php client.php
Conectat la server
Introduceți o expresie aritmetică (de exemplu, 2+2, 5*3, etc.): 2popo2
Eroare: invalid decimal literal (<string>, line 1)
```

## Cod Server

```
import socket

# Configurare server
HOST = '0.0.0.0' # Ascultă pe toate interfețelele de rețea disponibile
PORT = 55555

# Crearea unui socket
server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server_socket.bind((HOST, PORT))
server_socket.listen(1)

print("Serverul este în așteptare...")

# Acceptarea unei conexiuni
client_socket, client_address = server_socket.accept()
print(f"Conectat de către {client_address}")

while True:
    # Trimite un prompt pentru introducerea expresiei aritmetice la client
    client_socket.send("Introduceți o expresie aritmetică (de exemplu, 2+2, 5*3, etc.): ".encode('utf-8'))

    # Primește date de la client
    data = client_socket.recv(1024)

    # Decodează expresia primită de la client
    expression = data.decode('utf-8')

    try:
        # Evaluează expresia și obține rezultatul
        result = str(eval(expression))
        response = f"Rezultat: {result}\n"
    except Exception as e:
        # În caz de eroare, generează un mesaj de eroare
        response = f"Eroare: {str(e)}\n"

    # Trimite răspunsul la client
    client_socket.send(response.encode('utf-8'))

# Închide socket-ul clientului
client_socket.close()
```

## Cod Client4

```
<?php
$host = "127.0.0.1";
```

```

$port = 55555;

// Crearea socket-ului
$client_socket = socket_create(AF_INET, SOCK_STREAM, SOL_TCP);

if (!$client_socket) {
    die("Imposibil de creat socket\n");
}

if (!socket_connect($client_socket, $host, $port)) {
    die("Imposibil de conectat la server\n");
}

echo "Conectat la server\n";

while (true) {
    // Citirea prompt-ului de la server
    $prompt = socket_read($client_socket, 1024);
    echo $prompt;

    // Citirea expresiei de la utilizator
    $expression = readline();
    socket_write($client_socket, $expression, strlen($expression));

    // Citirea răspunsului de la server și afișarea sa
    $response = socket_read($client_socket, 1024);
    echo $response;
}

// Închiderea socket-ului la final
socket_close($client_socket);
?>

```