

# Partea I. Server-Client în limbajul C

## 1. Ce face proiectul

Proiectul constă într-o aplicație client–server realizată în limbajul C. Aceasta permite trimiterea unui număr întreg de la client către server. După ce primește numărul, serverul calculează primii  $n$  termeni ai șirului Fibonacci, determină suma lor, creează un mesaj cu aceste informații și îl trimite înapoi clientului. Comunicarea se face prin protocolul TCP, utilizând socketuri.

## 2. Program Server

### 2.1. Cod comentat (server\_fib\_sum.c)

Serverul creează un socket TCP, se leagă la o adresă IP și port, apoi așteaptă conexiuni de la clienți. Când primește un număr de la client, calculează șirul Fibonacci și suma acestuia, formează un mesaj text și îl trimite înapoi clientului.

```
#include <stdio.h> // pentru input/output
#include <stdlib.h> // pentru functii ca atoi()
#include <string.h> // pentru functii de lucru cu siruri (strcat, sprintf)
#include <unistd.h> // pentru functiile read, write, close
#include <arpa/inet.h> // pentru lucrul cu adrese IP si sockets

#define PORT 8822 // portul TCP pe care asculta serverul
#define MAX 1024 // dimensiune maxima buffer

// Functie care calculeaza primii n termeni Fibonacci si suma lor
void compute_fibonacci(int n, int* seq, int* sum) {
    seq[0] = 0;
    if (n > 1) seq[1] = 1;
    *sum = seq[0] + (n > 1 ? seq[1] : 0);

    for (int i = 2; i < n; i++) {
        seq[i] = seq[i-1] + seq[i-2]; // fiecare termen e suma celor doua
        // anterioare
        *sum += seq[i]; // se acumuleaza suma totala
    }
}

int main() {

    int sockfd, newsockfd, len, n, sum;
    struct sockaddr_in servaddr, client;
    char buffer[MAX]; // buffer pentru citire
    int fib[100]; // array pentru sirul Fibonacci

    // Create socket TCP
```

```

sockfd = socket(AF_INET, SOCK_STREAM, 0);

// Setare adresa IP si port
servaddr.sin_family = AF_INET;
servaddr.sin_addr.s_addr = inet_addr("37.120.249.45");
servaddr.sin_port = htons(PORT);

// Asociere socket cu adresa IP si port
bind(sockfd, (struct sockaddr*)&servaddr, sizeof(servaddr));

// Pune socketul in modul de ascultare
listen(sockfd, 1);
printf("\n[SERVER] Listening on 37.120.249.45:%d...\n", PORT);

// Acceptare conexiune de la un client
len = sizeof(client);
newsockfd = accept(sockfd, (struct sockaddr*)&client, &len);

// Primire numar n de la client
read(newsockfd, buffer, sizeof(buffer));
n = atoi(buffer); // transforma sirul de caractere in numar intreg

// Calcul Fibonacci si suma
compute_fibonacci(n, fib, &sum);

// Creare mesaj de raspuns
char response[MAX] = "The Fibonacci sequence with ";
char num[10];
sprintf(num, "%d", n);
strcat(response, num);
strcat(response, " element/s is: ");

for (int i = 0; i < n; i++) {
    char temp[20];
    sprintf(temp, "%d ", fib[i]); // adauga fiecare termen in sir
    strcat(response, temp);
}

// Adaugare suma
char temp[50];
sprintf(temp, "| SUM = %d", sum);
strcat(response, temp);

response[MAX - 1] = '\0'; // asigurare ca mesajul e terminat

// Trimitere mesaj inapoi catre client
write(newsockfd, response, strlen(response));

```

```

    printf("[SERVER] Sent to c_fib_sum - by Divoiu Mihaela gr1088:\n%s\n",
response);

    // Inchidere conexiune
    close(newsockfd);
    close(sockfd);
    return 0;
}

```

## 2.2. Print screen cu rularea serverului

```

gr5_22@sys.ase.ro:22 - Bitvise xterm - gr5_22@sys:~/proiect
[gr5_22@sys proiect]$ ./s_fib_sum

[SERVER] Listening on 37.120.249.45:8822...
[SERVER] Sent to c_fib_sum - by Divoiu Mihaela gr1088:
The Fibonacci sequence with 5 element/s is: 0 1 1 2 3 | SUM = 7
[gr5_22@sys proiect]$

```

## 2.3 Portul si protocolul

Serverul folosește portul 8822 și protocolul TCP. Acest lucru se observă din tipul socketului (SOCK\_STREAM) și utilizarea funcțiilor bind, listen și accept, specifice TCP.

## 2.4 Procesul

Serverul creează un socket, îl asociază la adresa IP 37.120.249.45 și portul 8822, apoi așteaptă conexiuni de la clienți. La primirea unei conexiuni, citește un număr, efectuează calculul necesar, trimite rezultatul, apoi închide conexiunea. Activitatea serverului poate fi urmărită folosind comenzile `netstat -a -n -p | grep -w 8822` și `ps -p -u gr5_22 | grep s_fib_sum`.

```

gr5_22@sys.ase.ro:22 - Bitvise xterm - gr5_22@sys:~/proiect
[gr5_22@sys proiect]$ netstat -a -n -p | grep -w 8822
(Not all processes could be identified, non-owned process info
will not be shown, you would have to be root to see it all.)
tcp        0      0 37.120.249.45:8822  0.0.0.0:*           LISTEN      3418380/./s_fib_sum
[gr5_22@sys proiect]$ ps -p -u gr5_22 | grep s_fib_sum
sshd(3418130)--bash(3418284)---s_fib_sum(3418380)
[gr5_22@sys proiect]$ netstat -a -n -p | grep -w 8822
(Not all processes could be identified, non-owned process info
will not be shown, you would have to be root to see it all.)
tcp        0      0 37.120.249.45:8822  0.0.0.0:*           LISTEN      3418380/./s_fib_sum
tcp        0      0 37.120.249.45:8822  37.120.249.45:48354 ESTABLISHED 3418380/./s_fib_sum
tcp        0      0 37.120.249.45:48354 37.120.249.45:8822  ESTABLISHED 3418432/./c_fib_sum
[gr5_22@sys proiect]$ netstat -a -n -p | grep -w 8822
(Not all processes could be identified, non-owned process info
will not be shown, you would have to be root to see it all.)
tcp        0      0 37.120.249.45:8822  37.120.249.45:48354 TIME_WAIT   -
[gr5_22@sys proiect]$

```

### 3. Program Client

#### 3.1 Cod comentat (client\_fib\_sum.c)

Clientul cere utilizatorului să introducă un număr, îl trimite serverului și primește înapoi un mesaj care conține șirul Fibonacci și suma acestuia. Mesajul este apoi afișat pe ecran.

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <arpa/inet.h>

#define PORT 8822
#define MAX 1024

int main() {

    int sockfd;
    struct sockaddr_in servaddr;
    char buffer[MAX]; // pentru raspunsul serverului
    char input[10]; // pentru valoarea introdusa de utilizator

    // Creare socket TCP
    sockfd = socket(AF_INET, SOCK_STREAM, 0);

    // Configurare adresa server
    servaddr.sin_family = AF_INET;
    servaddr.sin_port = htons(PORT);
    servaddr.sin_addr.s_addr = inet_addr("37.120.249.45");

    // Conectare socket la server
    connect(sockfd, (struct sockaddr*)&servaddr, sizeof(servaddr));

    // Solicita utilizatorului unui numar
    printf("Enter a number (n) to get Fibonacci + sum: ");
    fgets(input, sizeof(input), stdin);

    // Trimitere numar la server
    write(sockfd, input, strlen(input));

    // Primire raspuns de la server
    int bytes = read(sockfd, buffer, sizeof(buffer) - 1);
    if (bytes >= 0) {
        buffer[bytes] = '\0'; // adaugare terminator null
    }
}
```

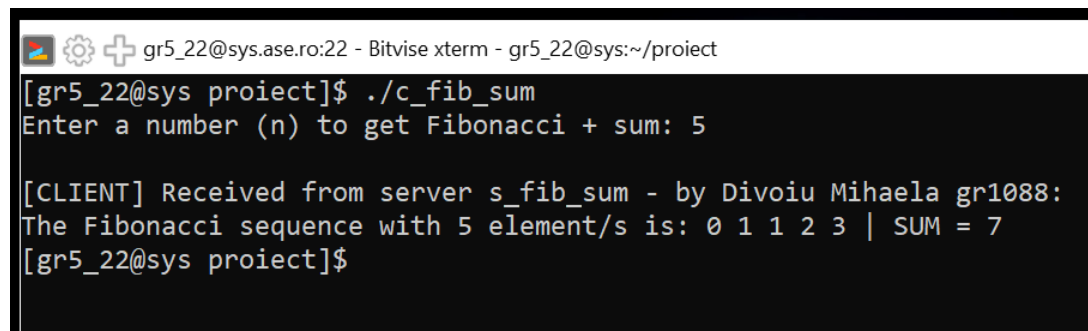
```

    // Afisare rezultat
    printf("\n[CLIENT] Received from server s_fib_sum - by Divoiu Mihaela
gr1088:\n%s\n", buffer);

    // Inchidere conexiune
    close(sockfd);
    return 0;
}

```

### 3.2 Print screen cu rularea clientului



```

gr5_22@sys.ase.ro:22 - Bitvise xterm - gr5_22@sys:~/proiect
[gr5_22@sys proiect]$ ./c_fib_sum
Enter a number (n) to get Fibonacci + sum: 5

[CLIENT] Received from server s_fib_sum - by Divoiu Mihaela gr1088:
The Fibonacci sequence with 5 element/s is: 0 1 1 2 3 | SUM = 7
[gr5_22@sys proiect]$

```

## Partea II. Server-Client în limbajul Python

### 1. Ce face proiectul

A doua parte a proiectului este implementată în Python și funcționează similar cu cea din limbajul C. Diferența este că șirul Fibonacci generat de server este inversat înainte de a fi trimis către client.

### 2. Program Server

#### 2.1. Cod comentat (server\_fib\_reverse.py)

Serverul în Python primește un număr de la client, generează șirul Fibonacci, îl inversează și trimite rezultatul înapoi.

```

import socket # modul pentru comunicare prin socketuri

# Functie care genereaza primii n termeni din sirul Fibonacci
def fibonacci(n):
    if n <= 0:
        return []
    elif n == 1:
        return [0]

```

```

seq = [0, 1]
for _ in range(2, n):
    seq.append(seq[-1] + seq[-2]) # fiecare termen e suma celor doua
anterioare
return seq

# Setari retea
HOST = '37.120.249.45'
PORT = 8822

# Se creeaza socketul TCP
with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as server_socket:
    server_socket.bind((HOST, PORT)) # asociaza socketul cu IP-ul si portul
    server_socket.listen(1) # serverul asteapta o conexiune
    print(f"\n[SERVER] Listening on {HOST}:{PORT}")

    conn, addr = server_socket.accept() # accepta conexiunea de la client
    with conn:
        print(f"[SERVER] Connected by {addr}")
        data = conn.recv(1024).decode() # primeste datele (n) trimise de client

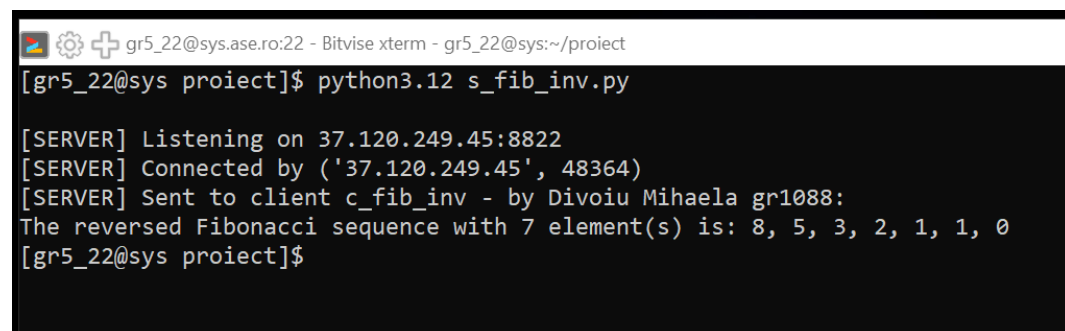
        if data.isdigit():
            n = int(data)
            fib_seq = fibonacci(n) # calculeaza sirul Fibonacci
            reversed_seq = list(reversed(fib_seq)) # inverseaza lista
            response = ', '.join(map(str, reversed_seq)) # formateaza ca text

            response = f"The reversed Fibonacci sequence with {n} element(s) is: "
            response += ', '.join(map(str, reversed_seq))
        else:
            response = "Invalid input. Please send a number."

        conn.sendall(response.encode()) # trimite rezultatul clientului
        print(f"[SERVER] Sent to client c_fib_inv - by Divoiu Mihaela
gr1088:\n{response}")

```

## 2.2 Print screen cu rularea serverului



```

gr5_22@sys.ase.ro:22 - Bitwise xterm - gr5_22@sys:~/proiect
[gr5_22@sys proiect]$ python3.12 s_fib_inv.py

[SERVER] Listening on 37.120.249.45:8822
[SERVER] Connected by ('37.120.249.45', 48364)
[SERVER] Sent to client c_fib_inv - by Divoiu Mihaela gr1088:
The reversed Fibonacci sequence with 7 element(s) is: 8, 5, 3, 2, 1, 1, 0
[gr5_22@sys proiect]$

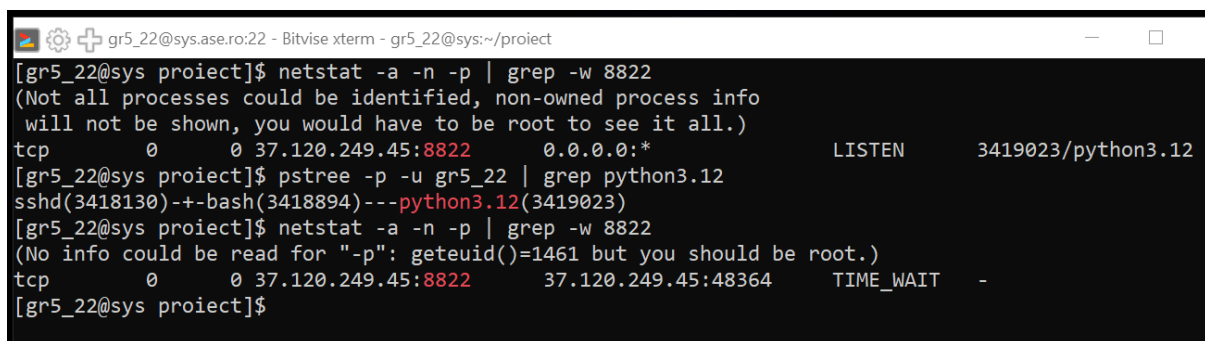
```

## 2.3 Portul cu identificare protocolului

Se utilizează același port (8822) și protocolul TCP. Acest lucru este confirmat de folosirea `socket.SOCK_STREAM` și a funcțiilor `bind`, `listen`, `accept`.

## 2.4 Procesul

Serverul inițiază un socket, îl leagă la adresa IP 37.120.249.45 și portul 8822, apoi rămâne în așteptarea unei conexiuni din partea unui client. Odată ce conexiunea este acceptată, serverul primește un număr, calculează șirul Fibonacci corespunzător, îl inversează și formează un mesaj text cu rezultatul. Acest mesaj este trimis clientului, iar conexiunea este închisă imediat după trimitere. Funcționarea serverului poate fi verificată folosind comenzile `netstat -a -n -p | grep -w 8822` și `ps -p -u gr5_22 | grep python3.12`.



```
gr5_22@sys.ase.ro:22 - Bitvise xterm - gr5_22@sys:~/proiect
[gr5_22@sys proiect]$ netstat -a -n -p | grep -w 8822
(Not all processes could be identified, non-owned process info
will not be shown, you would have to be root to see it all.)
tcp        0      0 37.120.249.45:8822  0.0.0.0:*        LISTEN      3419023/python3.12
[gr5_22@sys proiect]$ ps -p -u gr5_22 | grep python3.12
sshd(3418130)-+-bash(3418894)---python3.12(3419023)
[gr5_22@sys proiect]$ netstat -a -n -p | grep -w 8822
(No info could be read for "-p": geteuid()=1461 but you should be root.)
tcp        0      0 37.120.249.45:8822  37.120.249.45:48364  TIME_WAIT   -
[gr5_22@sys proiect]$
```

## 3. Program Client

### 3.1 Cod comentat (client\_fib\_reverse.py)

Clientul trimite un număr serverului și primește înapoi șirul Fibonacci inversat, pe care îl afișează.

```
import socket # modul pentru socket TCP/IP

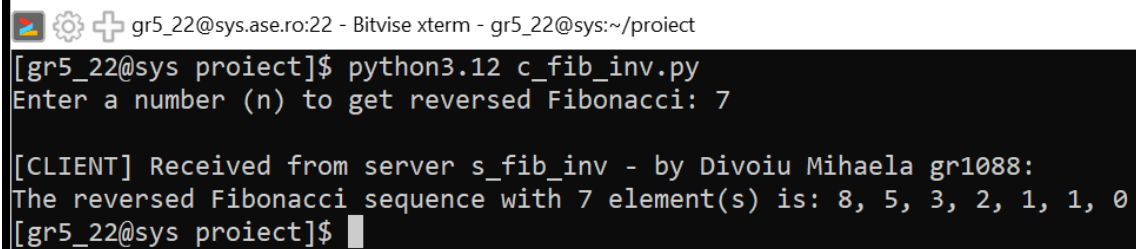
# Setari retea
HOST = '37.120.249.45'
PORT = 8822

# Solicita utilizatorului un numar
n = input("Enter a number (n) to get reversed Fibonacci: ")

# Creare socket TCP si conectare la server
with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
    s.connect((HOST, PORT)) # realizeaza conexiunea
    s.sendall(n.encode()) # trimite n ca sir de caractere
    data = s.recv(1024).decode() # primește raspunsul de la server
```

```
# Afisare rezultat
print("\n[CLIENT] Received from server s_fib_inv - by Divoiu Mihaela gr1088:")
print(data)
```

### 3.2 Print screen cu rulara clientului



The screenshot shows a terminal window titled "gr5\_22@sys.ase.ro:22 - Bitvise xterm - gr5\_22@sys:~/proiect". The user has executed the command `python3.12 c_fib_inv.py`. The program prompts the user to "Enter a number (n) to get reversed Fibonacci: 7". The output of the program is displayed as follows:

```
[CLIENT] Received from server s_fib_inv - by Divoiu Mihaela gr1088:
The reversed Fibonacci sequence with 7 element(s) is: 8, 5, 3, 2, 1, 1, 0
[gr5_22@sys proiect]$
```



## Bibliografie

1. Beej's Guide to Network Programming – Using Internet Sockets, <https://beej.us/guide/bgnet/>
2. Python socket – Documentație oficială, <https://docs.python.org/3/library/socket.html>
3. Real Python – Socket Programming in Python, <https://realpython.com/python-sockets/>
4. MathWorld – Fibonacci Numbers, <https://mathworld.wolfram.com/FibonacciNumber.html>
5. Materiale de seminar – Rețele de Calculatoare, <https://carment.ase.ro/rc/sem/socket.html>