

# *Mini server web*

Grupa C-113D

Std. Sg. Negoescu Elena - Camelia

Std. Sg. Rusu Petru - Călin



# Cuprins

## Capitolul 1: Introducere

- 1.1 Scopul proiectului
- 1.2 Lista definițiilor
- 1.3 Structura DCS

## Capitolul 2: Descrierea generală a produsului software

- 2.1 Descrierea produsului software
- 2.2 Detalierea platformei HW/SW

## Capitolul 3: Detalierea cerințelor specifice

- 3.1 Cerințe funcționale:
- 3.2 Cerințe non-funcționale:

## Capitolul 4: Bibliografie



## Capitolul 1: Introducere

### 1.1 Scopul proiectului

Scopul acestui proiect este dezvoltarea unui mini server web care permite utilizatorilor să acceseze un forum anonim, unde pot încărca și descărca write-up-uri de CTF, precum și să adauge comentarii și descrieri. Acest proiect urmărește să ofere o platformă simplă și eficientă, care să încurajeze colaborarea și schimbul de informații între pasionații de CTF, într-un mod anonim.

### 1.2 Lista definițiilor

ABREVIERE	TERMEN
DCS	Document cu cerințe software
HTTP	Hypertext Transfer Protocol
SW	Software
CGI	Common Gateway Interface
CTF	Capture the Flag
DNS	Domain Name Server
WWW	World Wide Web
HW	Hardware

### 1.3 Structura DCS

În documentul cu cerințe software se va prezenta în capitol 2 descrierea generală a mini serverului web ca mai apoi, în capitolul 3, să fie detaliate toate cerințele funcționale. De asemenea, tot în capitolul 2, se vor prezenta elementele de hardware și software pentru care produsul este dezvoltat.



## **Capitolul 2: Descrierea generală a produsului software**

### **2.1 Descrierea produsului software**

Scopul acestui proiect este de a dezvolta un mini server web în limbajul C, cu suport pentru gestionarea conexiunilor multiple utilizând tehnici de thread pooling. Serverul va permite utilizatorilor să acceseze un forum anonim dedicat comunității CTF (Capture The Flag), unde aceștia vor putea încărca și descărca write-up-uri (descrieri ale rezolvărilor problemelor) și adăuga comentarii și descrieri. Implementarea va include funcționalități de bază pentru procesarea cererilor HTTP (GET, POST, PUT), tratarea eficientă a conexiunilor prin crearea de procese noi sau thread-uri, suport pentru CGI, gestionarea corectă a headerelor HTTP și a codurilor de stare (inclusiv pagini predefinite pentru erori, cum ar fi 404, 403, 300), precum și compresia datelor transmise, în format pdf.

### **2.2 Detalierea platformei HW/SW**

Cerințe minime pentru ca aplicația să poată fi rulată:

- Sistem de operare: Windows 7 sau mai recent sau orice distribuție de Linux
- Spațiu de stocare: 3 GB disponibil
- Conexiune la internet stabilă

## **Capitolul 3: Detalierea cerințelor specifice**

### **3.1 Cerințe funcționale:**

#### **1. Multithreading pentru gestionarea conexiunilor simultane:**

- Serverul trebuie să gestioneze conexiuni multiple folosind thread-uri, fiecare cerere nouă fiind tratată de un thread dedicat.
- Serverul va putea primi cereri HTTP de la clienți și le va procesa în mod paralel, fără a bloca alte cereri în curs.

#### **2. Crearea și configurarea socket-ului serverului:**

- Serverul trebuie să creeze un socket folosind funcția socket din biblioteca sys/socket, să asocieze socket-ul cu adresa IP a serverului (localhost) și portul 8080 utilizând bind, și să asculte conexiunile client folosind listen.



**3. Procesarea cererilor HTTP:**

- Serverul va accepta conexiuni de la clienți prin accept și va procesa cereri HTTP de tip GET, POST și PUT.

**4. Generarea răspunsurilor HTTP:**

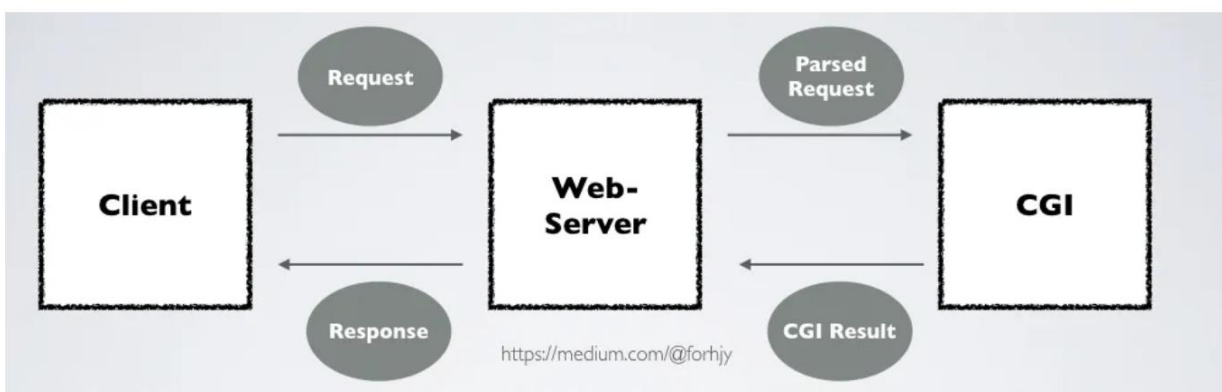
- Serverul va genera răspunsuri conform protocolului HTTP, incluzând linia de status (versiune HTTP, cod de stare și text de status), antetul răspunsului (Content-Type, Content-Length) și corpul răspunsului (fișierul solicitat sau codul HTML).
- Răspunsurile vor fi trimise înapoi clientului prin socket folosind write

**5. Gestionarea codurilor de eroare HTTP:**

- Serverul va returna coduri de eroare corespunzătoare, cum ar fi 404 pentru fișierele lipsă, și va afișa pagini HTML predefinite pentru aceste erori.

**6. Suport pentru CGI:**

- CGI (Common Gateway Interface) permite serverului web să ofere pagini web dinamice clienților. Atunci când un client trimite o cerere (de obicei prin metodele GET sau POST) către server, serverul web creează un proces pentru a rula datele solicitate. Ulterior, serverul trimite rezultatul înapoi către client.



CGI Workflow

## 3.2 Cerințe non-funcționale:

**1. Performanță:**



- Serverul trebuie să fie capabil să gestioneze simultan un număr mare de conexiuni cu o latență minimă, prin utilizarea eficientă a multithreading-ului.
- 2. **Scalabilitate:**
  - Soluția trebuie să fie scalabilă, astfel încât să poată gestiona creșteri în numărul de utilizatori și cereri fără o degradare semnificativă a performanței.
- 3. **Ușurință în utilizare:**
  - Interfața și modul de utilizare al forumului trebuie să fie simple și intuitive, permițând utilizatorilor să încarce și descarce fișiere cu ușurință și să adauge comentarii fără complicații.
- 4. **Portabilitate:**
  - Codul serverului trebuie să fie portabil și să poată fi rulat pe diverse sisteme de operare, fără modificări semnificative.
- 5. **Compatibilitate:**
  - Serverul trebuie să fie compatibil cu majoritatea browserelor moderne și să respecte standardele HTTP 1.1 pentru a asigura interoperabilitatea.



## Capitolul 4: Bibliografie

- <https://forhgy.medium.com/42-webserv-cgi-programming-66d63c3b22db>
- <https://www.garshol.priv.no/download/text/http-tut.html>

