

Mini server web

Îndrumător:

Slt. Ing. Vaman Adina

Realizat de:

Std. Sg. Negoescu Elena - Camelia

Std. Sg. Rusu Petru – Călin



Cuprins

Capitolul 1: Introducere	3
1.1 Introducere	3
1.2 Lista definițiilor	3
1.3 Obiectivele proiectului	3
Capitolul 2: Detalierea cerințelor specifice	4
2.1 Cerințe funcționale:	4
2.1.1 Gestionarea cererilor HTTP:	4
2.1.2 Concurență și Performanță	4
2.1.3 Timeout pentru conexiuni inactive	4
2.1.4 Servirea fișierelor statice și dinamice	4
2.1.5 Suport pentru conexiuni sigure (SSL/TLS)	4
2.1.6 Compresia și optimizarea răspunsurilor	4
2.2 Structura Proiectului:	5
2.3 Cerințe și Configurare	5
2.3.1 Dependențe	5
2.3.2 Compilare și rulare	5
Capitolul 3: Concluzie	6
Capitolul 4: Bibliografie	6



Capitolul 1: Introducere

1.1 Introducere

Acest document prezintă implementarea unui mini server web în C++, cu accent pe conceptele fundamentale ale sistemelor de operare. Server-ul este conceput pentru a gestiona cereri simultane, utilizând tehnici avansate de programare în rețea și concurență, fiind un exemplu practic pentru înțelegerea arhitecturii serverelor web.

1.2 Lista definițiilor

ABREVIERE	TERMEN
DCS	Document cu cerințe software
HTTP	Hypertext Transfer Protocol
SW	Software
CGI	Common Gateway Interface
SSID	Session Identifier
DNS	Domain Name Server
WWW	World Wide Web
HW	Hardware

1.3 Obiectivele proiectului

- Dezvoltarea unui server web capabil să gestioneze multiple conexiuni simultan.
- Implementarea de mecanisme avansate pentru tratarea cererilor HTTP, inclusiv metodele GET, POST și PUT.
- Integrarea unui sistem modular, care să permită extinderea funcționalităților fără a afecta structura generală a proiectului.
- Asigurarea performanței prin utilizarea unui model de concurență eficient.
- Oferirea de funcționalități suplimentare, cum ar fi logarea, timeout-ul conexiunilor inactive și suport pentru fișiere statice.



Capitolul 2: Detalierea cerințelor specifice

2.1 Cerințe funcționale:

2.1.1 Gestionarea cererilor HTTP:

Serverul este capabil să proceseze cereri HTTP, incluzând metodele:

- **GET**: Servirea fișierelor statice, precum paginile HTML, CSS și JavaScript.
- **POST**: Manipularea datelor trimise de clienți, inclusiv încărcări de fișiere sau autentificare.
- **PUT**: Actualizarea resurselor existente pe server.
- **DELETE**: Ștergerea fișierelor sau datelor de pe server.

2.1.2 Concurență și Performanță

Serverul utilizează un **thread pool**, care oferă următoarele beneficii:

- ◆ Reduce timpul de creare și distrugere a firelor de execuție.
- ◆ Optimizează utilizarea resurselor hardware, distribuită uniform între fire.
- ◆ Permite procesarea cererilor simultan, utilizând socket-uri non-blocante.

2.1.3 Timeout pentru conexiuni inactive

Conexiunile inactive sunt gestionate printr-o listă de timeout (TimeList):

- Nodurile din listă conțin informații despre conexiuni și timpul de expirare.
- Conexiunile inactive sunt închise automat după depășirea timpului de așteptare.

2.1.4 Servirea fișierelor statice și dinamice

Serverul este capabil să ofere:

- Fișiere statice, inclusiv imagini și fișiere multimedia.
- Fișiere dinamice prin integrarea cu CGI, care permite rularea scripturilor precum PHP.

2.1.5 Suport pentru conexiuni sigure (SSL/TLS)

Prin utilizarea bibliotecii **OpenSSL**, serverul permite conexiuni HTTPS, oferind criptare pentru toate datele transmise între client și server.

2.1.6 Compresia și optimizarea răspunsurilor

Serverul suportă compresia datelor pentru a reduce latența și dimensiunea transferurilor. Răspunsurile sunt procesate eficient pentru a minimiza utilizarea lățimii de bandă.



2.2 Structura Proiectului:

Modulele principale:

- ✓ ☐ **main.cpp**: Punctul de intrare al aplicației, care inițializează serverul și începe ascultarea cererilor pe un port specificat.
- ✓ **WebServer.cpp și WebServer.h**: Gestionează logica principală a serverului, inclusiv inițializarea componentelor și monitorizarea conexiunilor.
- ✓ **HttpConnection/**: Manipulează cererile HTTP și construiește răspunsurile corespunzătoare.
- ✓ **ThreadPool/**: Implementează un mecanism eficient de gestionare a firelor, reducând costurile asociate creării acestora.
- ✓ **TimeList/**: Asigură monitorizarea conexiunilor inactive și închiderea lor după depășirea unui timeout.
- ✓ **Config/**: Gestionează parametrii de configurare a serverului, inclusiv portul, numărul de fire și opțiunile SSL.

2.3 Cerințe și Configurare

2.3.1 Dependențe

Pentru rularea proiectului, sunt necesare următoarele librării și pachete:

- **Pthreads** pentru gestionarea firelor de execuție.
- **OpenSSL** pentru suport HTTPS.
- **nlohmann-json** pentru procesarea datelor JSON.
- **php-cgi** pentru suport CGI.

Comenzi pentru instalare:

```
sudo apt update  
sudo apt install libssl-dev zlib1g-dev libcurl4-openssl-dev nlohmann-json3-dev php-cgi
```

2.3.2 Compilare și rulare

1. Clonarea proiectului:

```
git clone https://github.com/elenacamelia465/Mini-server-Web.git  
cd Mini-server-Web
```

2. Compilarea proiectului:

```
make
```

3. Rularea serverului:

```
./server [-p port] [-n numWorker] [-s] [-t timeout]
```

Parametrii acceptați:

- ☐ -p: Portul serverului (implicit: 8086).



- ☐ -n: Numărul de fire în thread pool (implicit: 5).
- ☐ -s: Activează conexiunile HTTPS.
- ☐ -t: Timeout pentru conexiuni inactive (implicit: 20 secunde).

Capitolul 3: Concluzie

Proiectul **Mini-server-Web** reprezintă o soluție completă pentru înțelegerea și implementarea unui server web minimalist în C++. Cu un design modular și funcționalități avansate, acesta poate fi extins pentru a servi aplicații mai complexe. Prin integrarea unor concepte avansate precum gestionarea concurenței, procesarea evenimentelor și securitatea datelor, proiectul oferă o bază solidă pentru dezvoltarea aplicațiilor web performante.

Capitolul 4: Bibliografie

<https://forhgy.medium.com/42-webserv-cgi-programming-66d63c3b22db>

<https://www.garshol.priv.no/download/text/http-tut.html>

<https://forhgy.medium.com/42-webserv-cgi-programming-66d63c3b22db>

<https://www.linkedin.com/pulse/damn-threads-web-server-too-nikhil-srivastava/>

https://github.com/Dungyichao/http_server

<https://www.garshol.priv.no/download/text/http-tut.html>

