**APLICAȚIE**

**MINI SERVER WEB**

****

Îndrumător:

Lt. Dan Avram

Realizat de :

Sd. Sg. Buzatu Constantin-Mihai

Sd. Sg. Banu Ion

Grupa C113A

# Cerinte funcționale

* Crearea unui server web
* Primirea și gestionarea cererilor HTTP de la client:
  + Serverul web trebuie să poată primi cereri HTTP de la client (browserul web) și să le gestioneaze corespunzător.
* Stocarea paginilor web statice:
  + Furnizarea paginilor web statice, ce nu necesită prelucrare la fiecare accesare.
* Implementarea metodelor folosite pentru HTTP request și response:
  + Trebuie să fie capabil să gestioneze metode precum GET pentru obținerea datelor, POST pentru trimiterea lor, PUT pentru actualizare și DELETE pentru ștergere.
* Implementarea unei modalități de a folosi scripturi prin intermediul CGI.
  + Asigurarea unei comunicări eficiente între serverul web și scripturi pentru prelucrarea corectă a cererilor primite de la client, dar și transmiterea acestora fără erori sau pierderi de informații.
* Pe lângă clasica abordare de creare a unui server bazat pe multithreading, se va implementa și o variantă bazată pe multiprocessing.
  + Gestionarea cererilor și răspunsurilor HTTP, gestionarea mai multor procese independente pentru a trata cererile, fiecare proces trebuie să poată rula in paralel.

# Ce este mai exact un server web?

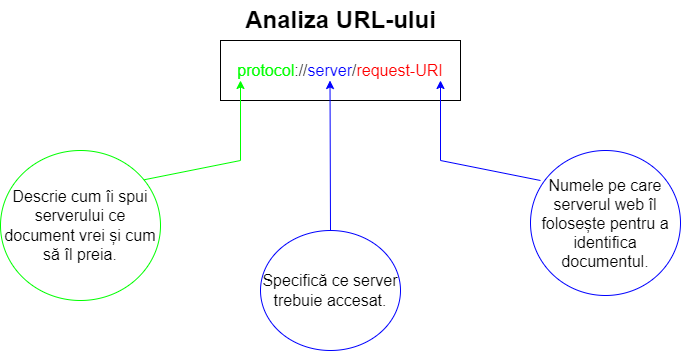
Este un program care primește ”cerințe” de la un browser și le răspunde acestora.

O cerință se referă la dorința unui client de a accesa un anumit document căruia îi cunoaște [URL](#_URL)-ul (adresa). Astfel serverul trimite un răspuns în care se află documentul (detalii [aici](#_Cum_funcționează_mai)).

# Cum funcționează mai exact comunicarea server-client? HTTP!

Un server web primește o [cerere](#_Requestul) HTTP de la un client. Serverul procesează cererea și decide ce [răspuns](#_Răspunsul) să trimită, generând un cod de stare HTTP. Serverul construiește antetele HTTP pentru răspuns, adăugând informații precum data, tipul de conținut, dar și alte informații relevante. Serverul atașează corpul răspunsului, care poate conține datele solicitate sau un mesaj de eroare. Răspunsul este trimis înapoi către client, care îl interpretează și afișează corespunzător.

# URL - structură



# Cererea

Există diferite tipuri de request-uri, dar cel mai comun este GET. Forma unui astfel de request este: GET document\_path HTTP/version.

Header-urile: Acestea conțin informații suplimentare despre request. Spre exemplu:

* Host specifică adresa serverului.
* User-Agent indică browserul și sistemul de operare utilizat.
* Accept specifică tipurile de conținut acceptate de client
* Accept-Language, Accept-Encoding specifică preferințele de limbă și de compresie a conținutului.

Exemplu:

**GET /pagina.html HTTP/1.1**

Host: www.example.com

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:70.0) Gecko/20100101 Firefox/70.0

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8

Accept-Language: en-US,en;q=0.5

Accept-Encoding: gzip, deflate

# Răspunsul

Când servărul primește o cerere HTTP, acesta localizează documentul cerut și îl trimite. Forma unui astfel de răspuns este: HTTP/version code text

Version = versiunea de HTTP folosită

Code = un număr format din 3 cifre, un status code

Text = un mesaj care descrie dacă totul este OK

După prima linie urmează antetul, care conține informații despre document, precum length-ul sau tipul, urmat de un spațiu și de conținutul documentului.

Exemplu:

HTTP/1.0 200 OK

Server: Netscape-Communications/1.1

Date: Tuesday, 25-Nov-97 01:22:04 GMT

Last-modified: Thursday, 20-Nov-97 10:44:53 GMT

Content-length: 6372

Content-type: text/html

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">

<HTML>

...conținutul documentului...

# CGI

Este o modalitate pentru server de a interacționa cu alte programe externe.

Se poate folosi prin crearea unui director CGI pe server și setarea acestui director ca fiind unul cu scripturi, care se vor executa când se va cere. Acestea reprezinta simple programe executabile, scrise în orice limbaj de programare.

Înainte ca programul să pornească, serverul setează un număr de variabile de environment care conțin informații primite prin cerere.

Răspunsul se va face prin standard output.

# Multiprocessing

To be continued...