**Universitatea Tehnică a Moldovei**

**Facultatea Calculatoare Informatică și Microelectronică**

**Departamentul** **Ingineria Software şi Automatică**

**Raport**

**Lucrare de laborator nr.1**

Disciplina: Programarea în rețea

Tema: Socket | HTTP | Multithreading

A îndeplinit: Gaina Nicoleta ,TI-171

A verificat: Buldumac Oleg

Chișinău 2020

Scopul lucrării:

1. Să se creeze o cerere(request) GET HTTP către web-serverul website-ului unite.md utilizînd socket.

2. Cu ajutorul expresiilor regulate să se extragă toate imaginile din raspunsul primit din punctul 1.

3. Utilizînd Socket, HTTP și firele de execuții să se descarce toate imaginile găsite din puncul 2 într-un folder

Intrebări de control:

1. Ce este un socket?

Socket-ul e un tip de comunicare între un server și un client, în care datele sunt transmise real time(live).

1. Scopul protocolului HTTP?

Scopul protocolului HTTP este de transmiterea și obtinerea paginii web de la distanță.

1. Ce este un fir de execuție?

Un fir de execuție este cea mai mică unitate de procesare ce poate fi programată spre execuție de către sistemul de operare.

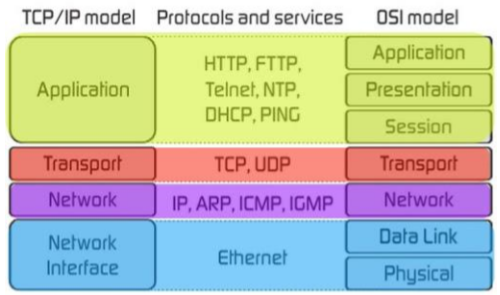
1. Diferența dintre TCP și UDP?

TCP așteaptă confirmare ca pachetele au fost primite, UDP nu. Respectiv, TCP urmarește pachetele, așa ca datele nu riscă de a fi pierdute, UDP nu le urmărește. UDP ne oferă însa rapiditate. De regula este folosit în chat-uri, video real time, ș.a.

1. Ordinea în care octeții sunt secvențializați în rețea?

Cea mai frecvent utilizată este big-endian(plasează octeții de la cel mai semnificativ la mai puțin semnificativ).

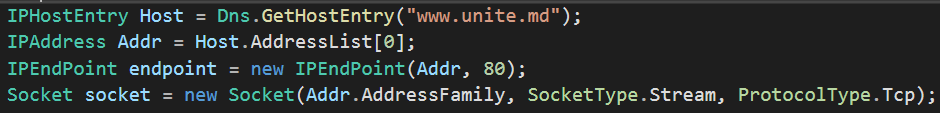
1. Cele patru nivele ale stivei TCP/IP și șapte nivele ale modelului OSI?



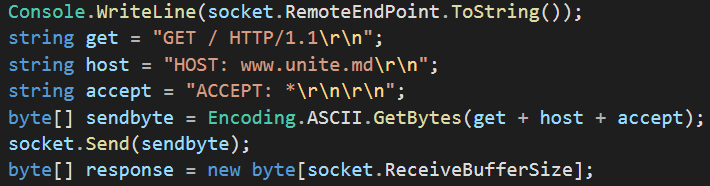
1. Din ce cauză nu puteți transmite cereri HTTP website-ului utm.md utilizînd portul 80 ?

Deoarece utm.md folosește protocolul HTTPS care respectiv se alfă pe alt port(443).

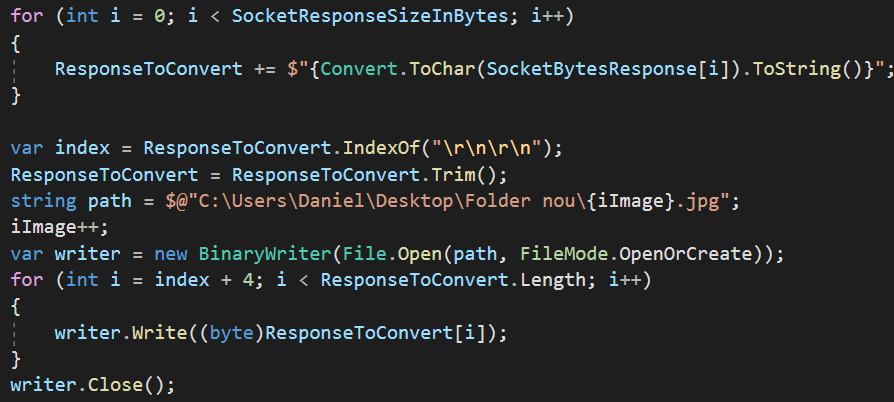
1. Cum se creează o conexiune Socket ?



1. Cum se transmit date intr-un Socket ?



1. Cum se citesc datele dintr-un Socket ?



1. Care sunt metodele HTTP?

* GET
* POST
* HEAD
* PUT
* DELETE
* CONNECT
* OPTION
* TRACE

1. Codurile de stare HTTP(200, 301, 302, 401, 404, 405, 500)

* 200 - OK
* 301 – Moved permanently
* 302 – Found(Previously ”Moved temporarily”)
* 401 – Unauthorized
* 404 – Not Found
* 405 – Method Not Allowed
* 500 – Internal Server Error

1. Ce este HTTP securizat?

S-HTTP (Secure HTTP) este o extensie la Hypertext Transfer Protocol (HTTP) care permite schimbul sigur de fișiere pe World Wide Web. Fiecare fișier S-HTTP este fie criptat, conține un certificat digital, fie ambele. Pentru un document dat, S-HTTP este o alternativă la un alt protocol de securitate binecunoscut, Secure Sockets Layer (SSL). O diferență majoră este că SHTTP permite clientului să trimită un certificat pentru autentificarea utilizatorului, în timp ce, folosind SSL, doar serverul poate fi autentificat. S-HTTP este mai probabil să fie utilizat în situațiile în care serverul reprezintă o bancă și necesită autentificare de la utilizator care este mai sigur decât un utilizator și o parolă.

1. Diferența dintre HTTP și HTTPS

**HTTP:**

1. HTTP înseamnă Hypertext Transfer Protocol

2. HTTP folosește TCP (Transmission Control Protocol), în general peste portul 80, pentru a trimite și primi pachete de date pe web.

**HTTPS:**

1. Secure (denumit și HTTP peste TLS sau HTTP peste SSL).
2. HTTPS folosește de asemenea TCP (Transmission Control Protocol) pentru a trimite și primi pachete de date, dar face acest lucru prin portul 443, într-o conexiune criptată de Transport Layer Security (TLS).
3. Diferența dintre Socket și WebSocket?

Chiar dacă realizează (în general) lucruri similare, da, sunt cu adevărat diferite. WebSockets rulează de obicei din browserele care se conectează la Application Server printr-un protocol similar cu HTTP care trece TCP / IP. Deci sunt în principal pentru aplicații web care necesită o conexiune permanentă la serverul său. Pe de altă parte, prize simple sunt mai puternice și generice. Se execută peste TCP / IP, dar nu sunt restricționate la browsere sau protocolul HTTP. Ele ar putea fi folosite pentru a implementa orice fel de comunicare. În multe aplicații web, websocket-urile sunt utilizate pentru a transmite mesaje către un client pentru actualizări în timp real. Una dintre caracteristicile mai interesante și adesea trecute cu vederea este faptul că majoritatea bibliotecilor websocket acceptă, de asemenea, răspunsuri directe la mesajele websocket de la un client ( recunoașterea mesajelor în coada- vorbirii ).

1. Diferența dintre un process și un fir de execuție?

**Proces:**

* O instanță de execuție a unui program se numește proces.
* Unele sisteme de operare folosesc termenul „sarcină” pentru a face referire la un program care este executat.
* Un proces este întotdeauna stocat în memoria principală denumit și memorie primară sau memorie de acces aleatoriu.
* Prin urmare, un proces este denumit o entitate activă. Dispare dacă mașina este repornită.
* Mai multe procese pot fi asociate cu același program.
* Pe un sistem multiprocesor, mai multe procese pot fi executate în paralel.
* Pe un sistem uni-procesor, deși nu se realizează un paralelism adevărat, se aplică un algoritm de planificare a procesului, iar procesorul este programat să execute fiecare proces, odată obținând o iluzie de concordanță.
* Exemplu: Executarea mai multor instanțe ale programului „Calculator”. Fiecare dintre cazuri este denumită un proces.

**Fir:**

* Un thread este un subset al procesului.
* Este denumit „proces ușor”, deoarece este similar cu un proces real, dar se execută în contextul unui proces și împărtășește aceleași resurse alocate procesului de către kernel.
* De obicei, un proces are un singur fir de control - un set de instrucțiuni ale mașinii care se execută simultan.
* Un proces poate fi format și din mai multe fire de execuție care execută simultan instrucțiuni.
* Mai multe fire de control pot exploata adevăratul paralelism posibil pe sistemele multiprocesoare.
* Pe un sistem uni-procesor, se aplică un algoritm de planificare a thread-ului, iar procesorul este programat să ruleze fiecare thread la rând.

1. Ce este un Deadlock și când el apare?

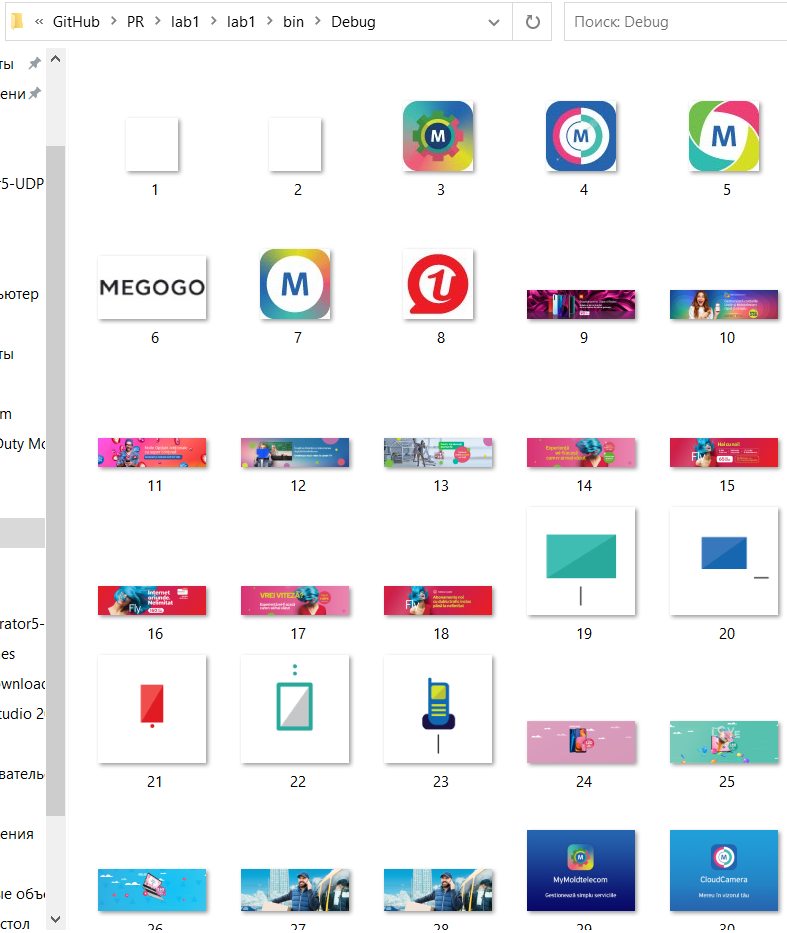
Deadlock este procesul de așteptare a incheierii procesului ce se află în execuție.

Apare atunci când avem mai multe procese care rulează unul după altul.

1. Pentru ce este nevoie de metoda join() in programarea multithreading?

Metoda join() permite unui thread să se poziționeze la sfârșitul altui thread.

Rezultate obținute:



Concluzie:

In urma efectuării acestei lucrări de laborator s-au acumulat competențe de lucru cu socket-urile. De asemenea au fost căpătate cunoștințe în ceea ce privește protocoalele HTTP, HTTPS, metodele acestor protocoale, metode de transmitere a pachetelor și altele.

În lucrarea dată s-a realizat conectarea la site-ul “unite.md” prin intermediul socket-ului, după care din codul HTML a fost extrase toate denumirile de imagini(\*.jpg;\*.png;\*.gif) intr-un array de string-uri prin intermediul expresiilor regulate, iar in final, toate aceste imagini au fost .