Ministerul Educaţiei, Culturii şi Cercetării Republicii Moldova

Universitatea Tehnică a Moldovei

Facultatea Calculatoare Informatică și Microelectronica

Departamentul Ingineria Software şi Automatică

**Referat**

**Disciplina: Programarea în rețea**

**Laboratorul 1**

Tema: Socket | HTTP | Multithreading

A elaborat:

Pavlenco Ecaterina, st. gr. TI-171

A verificat:

Buldumac Oleg

Chișinău 2020

Sarcina lucrării:

* Să se creeze o cerere(request) GET HTTP către web-serverul website-ului utm.md utilizînd socket.
* Cu ajutorul expresiilor regulate să se extragă toate imaginile din raspunsul primit din punctul 1.
* Utilizînd Socket, HTTP și firele de execuții să se descarce toate imaginile găsite din puncul 2 într-un folder.

Mersul lucării:

În cadrul acestui laborator s-a propus utilizarea soketurilor ,

Pentru puntul 1 unde trebuia să se creeze o cerere request GET HTTP către web-serverul website-ului utm.md utilizînd socket. Site-ul utm.md este utilizat protocolul https (protocolul securizat) , de aceea am utilizat conexiunea TCP prin clasa TcpClient , și clasa SslStreams deja pentru conexiunea client-server pentru portul securizat . După care s-a implimentat cererea- request-ul , în request introducem adresa , metoda și protocolul de comunicare a site-ului.

însăși requestul :



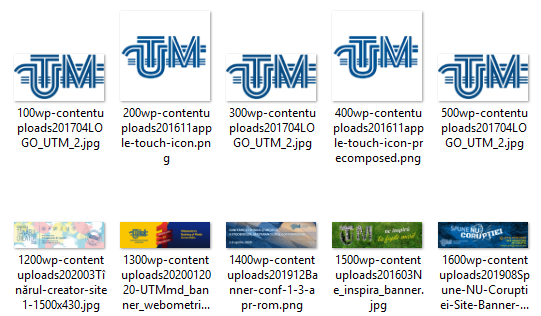
La punctul 2 trebuie de extras prin expresiile regulate toate imaginile din răspunsul primit la puncutl 1 , adică din html-ul afișat. Aici deja s-a folosit însăși clasa Regex , unde s-a făcut alegerea de imagini din cele obținute doar a imaginilor jpg|png|gif.



Punctul 3 presupune descărcarea imaginilor găsite prin fire de execuție. Pentru controlul acestora și restricția la 4 fire de execuție am utilizat semafoare, care presupune o simplă variabilă dar se utilizează pentru a controla accesul la o resursă comună prin mai multe procese. În cazul nostru am restricționat clasa Semathore la date de intrare 4 și numărul maxim de intrări la fel 4 : 

Pentru inițializarea firelor utilizăm deja clasa Thread 

care este inclusă într-un ciclu for, după cum observăm este legată de clasa Save image în care se descarcă imaginile.Pentru însăși descărcarea imaginilor noi am extras bytes a imaginilor , din lista imaginilor găsite anterior se făceau cereri la linkuri . După care însăși bytes le-am transformat în imagini respective care au fost descărcate.



Link-ul către github:

<https://github.com/katy198/PR-laboratorul1>

Video:



**Întrebări:**

• Ce este un Socket ?

Socket este un proces de comunicare între 2 sisteme(exemplu client-server).

• Scopul protocolului HTTP

Cel mai important scop este transmiterea datelor la nivel de text .

• Ce este un fir de execuție ?

Fir de execuție este o secvență de instrucțiuni care pot fi executate și gestionate independent de alte procese.

• Diferența dintre TCP si UDP

Una din cea mai bine cunoscută diferență este faptul că comunicare prin TCP necesită o conexiuni bine stabilită înainte de a transmite date, și trebuie să închidă conexiunea după transmiterea datelor pe când la comunicarea prin UDP nu se stabilește conexiune strictă , datele se transmit la o adresă, și nu se verifică daca au fost primite. Sau mai ușor spus UDP este utilizată la multicast.

• Ordinea în care octeții sunt secvențializați în rețea

Ordinea în care octeții sunt secvențializați într-un protocol de rețea octet orientat se numește comandă de byte de rețea .

• Cele patru nivele ale stivei TCP/IP și șapte nivele ale modelului OSI

Stiva de protocoale ale modelului TCP/IP : nivelul aplicației , nivelul transport, nivelul internet , nivelul legătură

Stiva de protocoale ale modelului OSI: nivelul aplicație, nivelul prezentare, nivelul sesiune , nivelul transport , nivelul pachete , nivelul cadre , nivelul biți .

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

Portul standart pentru HTTP este 80 , site-ul utm.md utilizează protocolul HTTPS care preupune aplicarea portului 443 (port securizat).

• Cum se creează o conexiune Socket ?

Se introduce adresa, host-ul și portul în conexiunii creată , TCP sau UDP.

• Cum se transmit date intr-un Socket ?

Prin request , în request introducem adresa , metoda și protocolul de comunicare.

• Cum se citesc datele dintr-un Socket ?

Prin intermediul clasei TcpClient sau UdpClient în dependență de conxiune , și se folosește metoda read() .

• Care sunt metodele HTTP?

Metodele HTTP sunt :

GET

POST

PUT

DELETE

OPTIONS

HEAD.

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

200- Raspuns cu succes primit

301 – Redirecție permanentă, (spre exemplu de la http la https)

301 – URL redirecție , (spre exemplu de la http la https)

401- Este eroare de client, clientul nu este autenticat pentru acces

404 – Adresa introdusă de client nu a fost găsită la server

405 – Metoda care transmitem este cunoscută de server , dar nu este permisă

500 – O eroare la prelucrarea procesului la nivel de server

• Ce este HTTP securizat ?

HTTPS este o extensie a protocolului HTTP, utilizat pentru securizarea infomației.

• Diferența dintre HTTP și HTTPS

HTTP este utilizat pentru comunicarea cu clarificări necuprimate, ceea ce înseamnă că datele transferate pot fi interceptate și citite în câmpie de către un om. Câmpurile de utilizator / parolă pot fi, de exemplu, capturate și citite.HTTPS se referă la comunicarea criptată SSL / TLS. Trebuie decriptat pentru a fi citit. În mod normal / ideal, doar punctele finale sunt capabile să cripteze / să decripteze datele.

• Diferența dintre Socket și WebSocket

WebSocket este diferit de protocolul HTTP prin care comunică Socket, WebSocket este făcut pentru comunicarea porturilor în afară de celea 80 și 443 care foloseste HTTP.

• Diferența dintre un proces și un fir de execuție

Noțiunea de proces înseamnă că orice program este în execuție. Pe când firul de execuție este segmentul unui proces .Îseamnă că un proces poate avea mai multe fire și aceste mai multe fire sunt conținute în cadrul unui proces.

• Ce este un Deadlock și când el apare ?

Deadlock este atuncti cînd un fir de execuție așteaptă un alt fir, inclusiv el însuși. Această poate fi datorită faptului cînd 2 fire utilizează o variabilă statică

• Pentru ce este nevoie de metoda join() în programarea multithreading?

Metoda join se utilizează când acesta va pune firul curent în așteptare până când firul pe care este numit este mort.