Retele de calculatoare – Informatica anul 3 (2019-2020)

Note de Laborator Retele de calculatoare	**Specializare: Informatica anul 3** **Contact:** **retelecdsd@gmail.com** **http://www.cdsd.ro**

Comunicatii de Date si Sisteme Distribuite
http://www.cdsd.ro

Laborator 12

1. Obiective:

- Aplicatii SSL (Secure Socket Layer)/ TLS (Transport Layer Security) **– Java/ Python (Anexa 3, pag13 - The Programming Process)**
- Comunicare Client-Server SSH
- Recapitulare VLSM

2. Consideratii teoretice (Partea practica- pag.2; Tema pag. 11)

SSL – Secure Socket Layer/ TLS – Transport Layer
https://wiki.mozilla.org/Security/Server_Side_TLS ; https://www.trumarkonline.org/docs/default-source/pdfs/all-browsers.pdf?sfvrsn=2; https://docs.microsoft.com/en-us/windows-server/security/tls/tls-schannel-ssp-changes-in-windows-10-and-windows-server ; https://datatracker.ietf.org/wg/tls/charter/ este un protocol de securitate care oferă **comunicare sigura prin Internet**. Protocolul permite aplicaţiilor client / server să comunice astfel incat sa **se împiedice capturarea, modificarea sau falsificarea mesajelor.**
Versiuni: SSL 1.0, 2.0 and 3.0; TLS 1.0 (SSL 3.1); TLS 1.1 (SSL 3.2); TLS 1.2 (SSL 3.3)

Protocolul SSL oferă securitatea conexiunii, având **trei proprietăţi**:

- **Conexiunea este privată -** Criptarea se utilizează după un înţelegerea iniţială pentru definirea cheii secrete. Criptografia simetrică se utilizează pentru criptarea simetrica a datelor (AES, 3DES, RC6 etc.)
- **Identitatea părţilor se autentifică utilizând criptografie asimetrică** (RSA, Curbe eliptice, DSS etc.)
- **Conexiunea este de încredere -** Transportul mesajului include verificarea acestuia cu un MAC (Message Authentication Code) parametrizat. Pentru calcularea MAC-ului se folosesc funcţiile de dispersie SHA, MD5, MAC-uri etc.

Scopurile protocolului SSL, în ordinea priorităţilor sunt:
1. **Securitatea criptografică**: SSL ar trebui să se utilizeze pentru conexiune sigură între două părţi.
2. **Inter-operabilitate**: Programatori independenţi ar trebui să fie capabili de a dezvolta aplicaţii SLL care să funcţioneze cu succes fără ca aceştia să aibă cunoştinţă de codul scris de altcineva.
3. **Extensibilitatea**: SSL încearcă să furnizeze un cadru în care să se integreze metode noi de criptare (simetrică sau asimetrică), acest lucru contribuind la evitarea creării unui protocol nou (riscând implicit să se introducă noi slăbiciuni) şi evitarea scrierii unei biblioteci de securitate noi.
4. **Eficienţa relativă**: Operaţiile criptografice tind să fie puternic procesor intensive, în particular criptografia cu chei publice. Din acest motiv, SSL a înglobat un **mecanism de caching al sesiunii** pentru a reduce numărul de conexiuni ce trebuie stabilite în totalitate.

3.7.2 Certificate

Pentru a crea un certificat putem sa utilizam programul *keytool.exe* care face parte din Java SDK.

https://docs.oracle.com/javase/8/docs/technotes/tools/unix/keytool.html (https://www.ibm.com/support/knowledgecenter/en/SSYKE2_7.0.0/com.ibm.java.security.component.70.doc/security-component/keytoolDocs/keytool_overview.html , http://www.entrust.net/knowledge-base/technote.cfm?tn=8425)
Programul poate fi gasit in directorul `$JAVA_HOME/bin`

Comanda pentru a genera un certficat este

```
keytool -genkey -keystore mySrvKeystore -keyalg RSA
```

Reference: https://docs.oracle.com/middleware/12212/wls/SECMG/keytool-summary-appx.htm

Programul va solicita informatii despre proprietarul certificatului precum si o parola. Introduceti parola 123456. Puteti sa folositi orice parola (CEEA CE ESTE SI INDICAT: FOLOSIREA UNEI PAROLE CORECT ALESE....
https://www.sans.org/security-resources/policies/general/pdf/password-protection-policy ; https://www.sans.org/security-resources/policies/general/pdf/password-construction-guidelines) dar in acest caz trebuie modificat si codul programelor/comenzilor. Dupa generare, certificatul va fi salvat in directorul de lucru sub numele de **`mySrvKeystore`**.

Pure-Python Java Keystore (JKS) library
https://pypi.python.org/pypi/pyjks
PyJKS enables Python projects to load and manipulate Java KeyStore (JKS) data without a JVM dependency. PyJKS supports JKS, JCEKS, BKS and UBER (BouncyCastle) keystores
KeyStore Explorer is an open source GUI replacement for the Java command-line utilities keytool and jarsigner. KeyStore Explorer presents their functionality, and more, via an intuitive graphical user interface. **http://keystore-explorer.org/**

3. Partea practica

3.1. Aplicatia A1 : Client-Server SSL (Java)

Indicatii

Pasul 1: Certificat
Pentru a crea un certificat putem sa utilizam programul *keytool.exe* care face parte din Java SDK.
Programul poate fi gasit in directorul `$JAVA_HOME/bin`

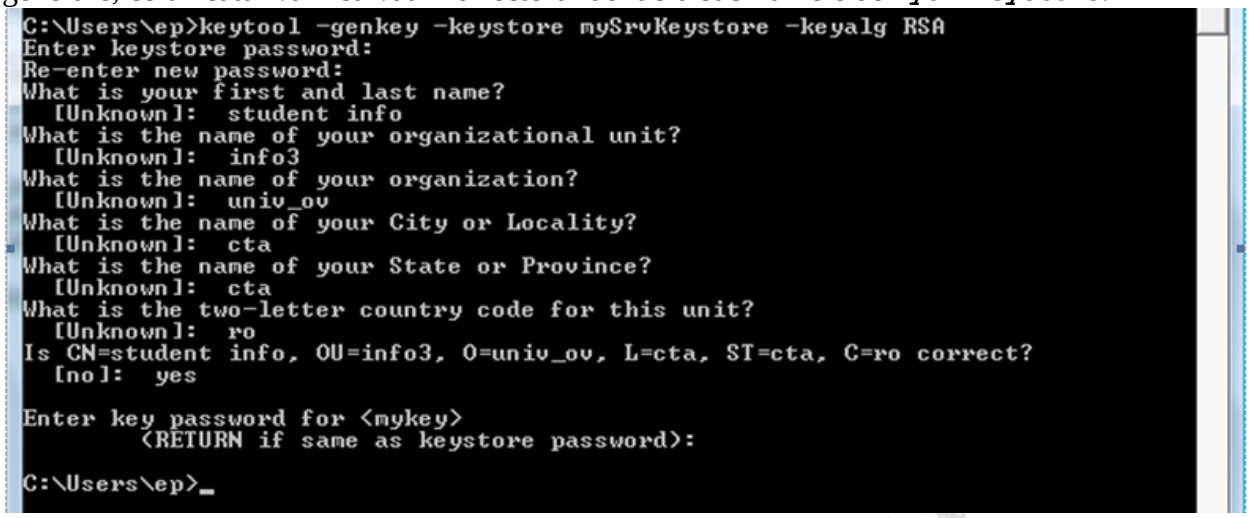
Comanda pentru a genera un certficat este:

```
keytool -genkey -keystore mySrvKeystore -keyalg RSA
```

Reference: https://docs.oracle.com/middleware/12212/wls/SECMG/keytool-summary-appx.htm

Atentie
Informatii de interes (care trebuies si aplicate !!!!!) cu privire la alegerea corecta a unei parole:
https://www.sans.org/security-resources/policies/general/pdf/password-protection-policy ; https://www.sans.org/security-resources/policies/general/pdf/password-construction-guidelines

Programul va solicita informatii despre proprietarul certificatului precum si o parola. Introduceti parola 123456. Puteti sa folositi orice parola dar in acest caz trebuie modificat si codul programelor. Dupa generare, certificatul va fi salvat in directorul de lucru sub numele de **mySrvKeystore**.



Pasul 2: SSLServer

```
//genereaza contextul SSL pentru Server: verifica si initializeaza
//KeyManagerul, incarca KeyStore-ul adecvat si intoarce un obiect de tip
//ServerSocketFactory.
```

```
//package Server;
import java.io.*;
import java.security.KeyStore;
import java.util.*;
import javax.net.ServerSocketFactory;
import javax.net.ssl.*;

public class SSLServer
{
    private static int port=4000;
    private static SSLServerSocketFactory sf;
    private static SSLServerSocket ss;

    public static void StabilireConexiune(int nrPort)
    {
        try {
            sf = (SSLServerSocketFactory) SSLServer.getServerSocketFactory();
            ss = (SSLServerSocket) sf.createServerSocket(nrPort);
            System.out.println("Server conectat si gata de a accepta noi conexiuni la adresa "
                    + ss.getLocalPort());
            String[] enable={"TLS_DH_anon_WITH_AES_128_CBC_SHA"};
            ss.setEnabledCipherSuites(enable);
            String[] cipherSuites = ss.getEnabledCipherSuites();
            System.out.println("CipherSuites: ");
            for(int i=0;i<cipherSuites.length;i++)
```

```
            {
                System.out.println(cipherSuites[i]);
            }
        }
        catch (Exception e)
        {e.printStackTrace();
    }
}
    private static SSLSocket clientSocket;

    public static void ConectareClient()
    {
        try
        {
            clientSocket = (SSLSocket) ss.accept();
            System.out.println("Client conectat cu succes");

            InputStream input = clientSocket.getInputStream();
            InputStreamReader inputreader = new InputStreamReader(input);
            BufferedReader br = new BufferedReader(inputreader);
            String string = null;

            while ((string = br.readLine()) != null) {
                System.out.println(string);
                System.out.flush();
            }
        }
         catch (Exception e) {e.printStackTrace();
     }
         finally
         {
             try{
                     clientSocket.close();
         }
         catch (IOException ex) {
             ex.printStackTrace();
         }
     }
}
```

```
private static ServerSocketFactory getServerSocketFactory()
{
    SSLServerSocketFactory ssf = null;
    try
    {
        SSLContext ctx;
        KeyManagerFactory kmf;
        KeyStore ks;

        char[] passphrase="123456".toCharArray();

        ctx = SSLContext.getInstance("TLS");
        kmf = KeyManagerFactory.getInstance("SunX509");
        ks = KeyStore.getInstance("JKS");

        ks.load(new FileInputStream("mySrvKeystore"), passphrase);
        kmf.init(ks,passphrase);
        ctx.init(kmf.getKeyManagers(), null, null);
        ssf = ctx.getServerSocketFactory();

        return ssf;
    }
    catch (Exception e)
    {
        e.printStackTrace();
    }
    return null;
}
public static void main (String[] args)
{
    if(args.length != 0) port = Integer.parseInt(args[0]);
    StabilireConexiune(port);

    while (true)
    {
        ConectareClient();
}
}
}
```

Pasul 3. SSLClient

```
//analog ca in cazul serverului,cu diferenta ca intoarce un obiect tip
//SocketFactory.
```

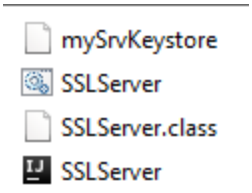
```
//package Client;
import java.io.*;
import java.security.KeyStore;
import java.util.logging.*;
import javax.net.ServerSocketFactory;
import javax.net.SocketFactory;
import javax.net.ssl.*;
public class SSLClient
{
    public static void main (String[] args)
    {
        conectare("127.0.0.1",4000);
    }

    private static SSLSocket socket;

    public static void conectare(String host, int port)
    {
        try
        {
            SSLSocketFactory factory = (SSLSocketFactory)
                    SSLClient.getSocketFactory();
            socket = (SSLSocket) factory.createSocket(host,port);
            BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

            String[] enable={"TLS_DH_anon_WITH_AES_128_CBC_SHA"};
            socket.setEnabledCipherSuites(enable);
            String[] cipherSuites = socket.getEnabledCipherSuites();
            System.out.println("CipherSuites: ");
            for(int i=0;i<cipherSuites.length;i++)
              {
                  System.out.println(cipherSuites[i]);
              }
    socket.addHandshakeCompletedListener(new HandshakeCompletedListener() {
    public void handshakeCompleted(HandshakeCompletedEvent event)
    {
            System.out.println("Handshake reusit");
    }
    });
```

```
        socket.startHandshake();
        PrintWriter out=new PrintWriter (new BufferedWriter(new
                OutputStreamWriter(socket.getOutputStream())));
        System.out.println("Dati mesajul catre server..");
        String string = br.readLine();
        out.println("Mesaj catre server..." + string);
        out.println();
        out.flush();
        }

        catch (IOException ex) {ex.printStackTrace(); }
       finally
       {
           try
           {
               socket.close();
           }
           catch (IOException ex)
           {
               Logger.getLogger(SSLClient.class.getName()).log(Level.SEVERE, null, ex);

           }
    }
}
    private static SocketFactory getSocketFactory()
    {
        SSLSocketFactory ssf = null;
        try
        {

        SSLContext ctx;
        KeyManagerFactory kmf;
        KeyStore ks;

        char[] passphrase="123456".toCharArray();

        ctx = SSLContext.getInstance("TLS");
        kmf = KeyManagerFactory.getInstance("SunX509");
        ks = KeyStore.getInstance("JKS");

        ks.load(new FileInputStream("mySrvKeystore"), passphrase);
        kmf.init(ks,passphrase);
        ctx.init(kmf.getKeyManagers(), null, null);
        ssf = ctx.getSocketFactory();

        return ssf;

    }
```

```
    catch (Exception e)
    {
        e.printStackTrace();
    }
    return null;
}
}
```

Pasul 4. Rularea aplicatiei Client-Server SSL

Certificatul creat la pasul 1 trebui copiat in directorul care contine fisierul bytecode pentru Server / respectiv Client.



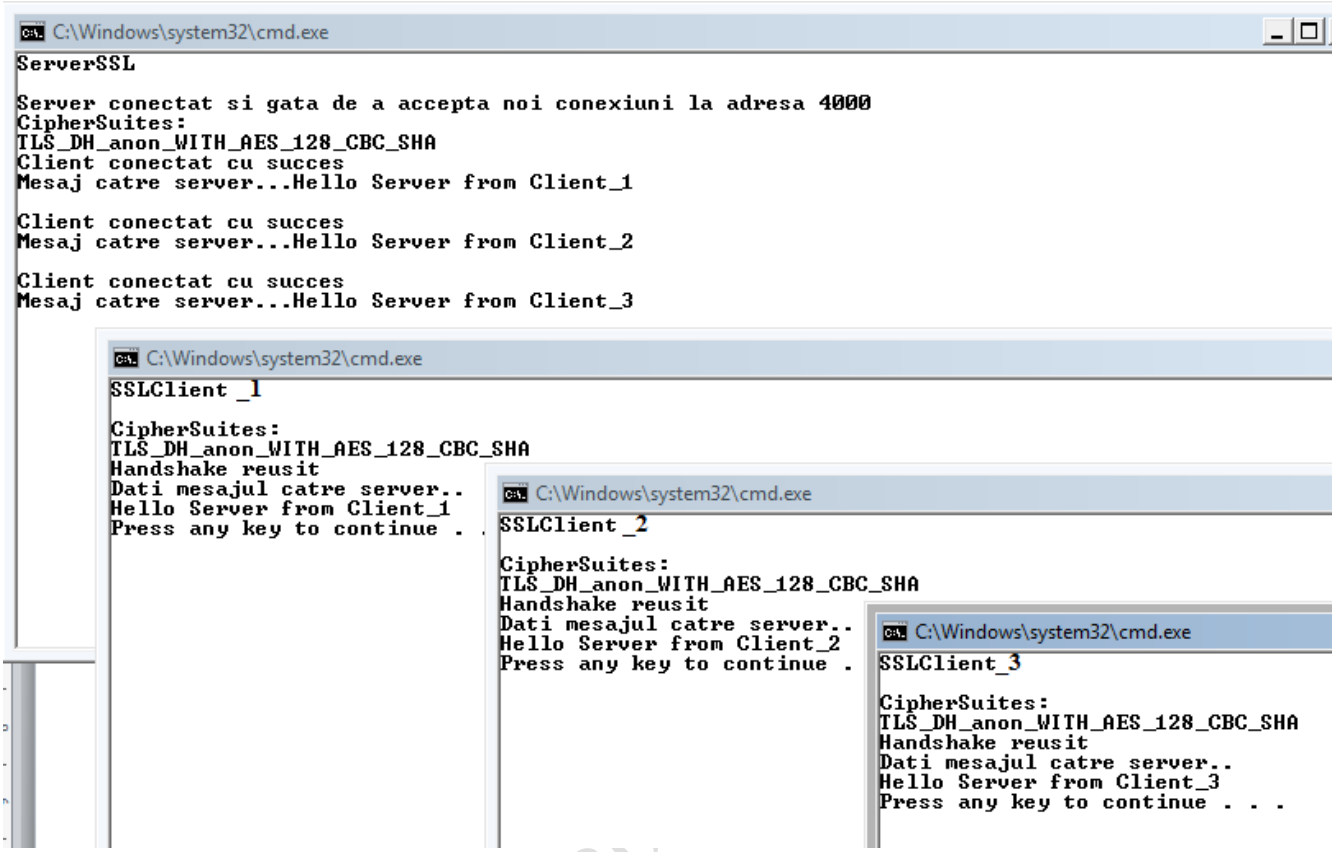
SSLServer.bat

```
1  @echo OFF
2  @echo ServerSSL
3  @echo:
4  javac SSLServer.java
5  java SSLServer
6  pause
```

SSLClient.bat Close All

```
1  @echo OFF
2  @echo SSLClient
3  @echo:
4  javac SSLClient.java
5  java SSLClient
6  pause
```

Exemplu Output program:



3.2. Aplicatia A2: (TEMA !!!!!)

Realizati o aplicatie **client-server SSL**, Solutie **Java**, cu minim 2 clienti care:

a. genereaza certificatul propriu
b. distribuie certficatul serverului
c. folosesc sesiune SSL/TLS pentru comunicarea fiecaruia dintre ei cu serverul

Indicatii: **SSL_mp4** (student anul 3 Info, FMI, UOC, 2017-2018)

3.3. Aplicatia A3: Exercitiu (TEMA !!!!!)

Realizati o aplicatie **client-server SSL**, Solutie **Python**, cu minim 2 clienti care:

a. genereaza certificatul propriu
b. distribuie certificatul serverului
c. folosesc sesiune SSL/TLS pentru comunicarea fiecaruia dintre ei cu serverul

Indicatii

- SSL/TLS client certificate verification with Python v3.4+ SSLContext
- How To Install OpenSSL on Windows
- Anexa 4, pag.14

Challenge: Interfata grafica

Recomandare: Qt Designer , cu Designer din Anaconda prompt).
http://pythonforengineers.com/your-first-gui-app-with-python-and-pyqt/,
https://www.codementor.io/deepaksingh04/design-simple-dialog-using-pyqt5-designer-tool-ajskrd09n, https://wiki.python.org/moin/PyQt/Tutorials

Recapitulare Python

- Python_intro (Lab_02, Lab_03)
- Programare_Python (Lab_02, Lab_03)
- Byte-of-python (Lab_02, Lab_03)
- Python socket network programming_1 (Lab_08)
- Python socket network programming_2 (Lab_08)
- Python Files and os.path (Lab_09)
- Programarea socket-urilor de retea in Python (Lab_09: BasicsOfSockets.pdf)
- Multithreading in Python (Lab_10)
- Lab 11 …

Obs: Anexa 3, pag13 - The Programming Process

3.4. Comunicare Client-Server SSH

3.4.1. Instalare si rulare Server SSH (Bitvise SSH Server, https://www.putty.org/)
3.4.2. Instalare si rulare Client SSH (Bitvise SSH Client, https://www.putty.org/ sau/si Putty)
3.4.3. Testarea conectivitatii Client Server SSH pe aceeasi masina si pe masini diferite ➔ sshNumePrenume.png
3.4.4. Capturi Wireshark (exemplu) ➔sshNumePrenume.cap

3.5. Recapitulare VLSM

Indicatii:

a. Note de curs, Note de laborator - http://www.cdsd.ro
b. **VLSM**

4. Tema:

- **Toate punctele din sectiunea 3 "partea practica"** se vor relua de catre cursanti, folosind etapele de lucru indicate. Rezultatele experimentale:
  - **L12_nume+prenume_Java (folder) - contine subfolderele 3.1, 3.2, fiecare subfolder cu fisierele .java, .bat, .png (Snipping Tool), insotite de un *readme.txt* pentru particularitati de rulare, conform prezentarilor facute.**
  - **L12_nume+prenume_Python (folder) - contine subfolderul 3.3 cu fisierele .py, .png (Snipping Tool) si .doc(readme, observatii)**
  - **L12_nume_prenume_SSH (folder) - contine subfolderele 3.4.1 - 3.4.4 fiecare cu fisierele .png, .cap si .doc corespunzatoare.**

se vor arhiva cu numele **L12_nume+prenume_info3.rar si** se va trimite prin e-mail la adresa **retelecdsd@gmail.com** precizandu-se la **subject: L12_nume+prenume_info3**, pana pe data de **6 ianuarie 2020 e.n., ora 8.00 a.m. (Atentie, gmail nu "prea vrea" .rar in .rar http://www.makeuseof.com/tag/4-ways-email-attachments-file-extension-blocked).**

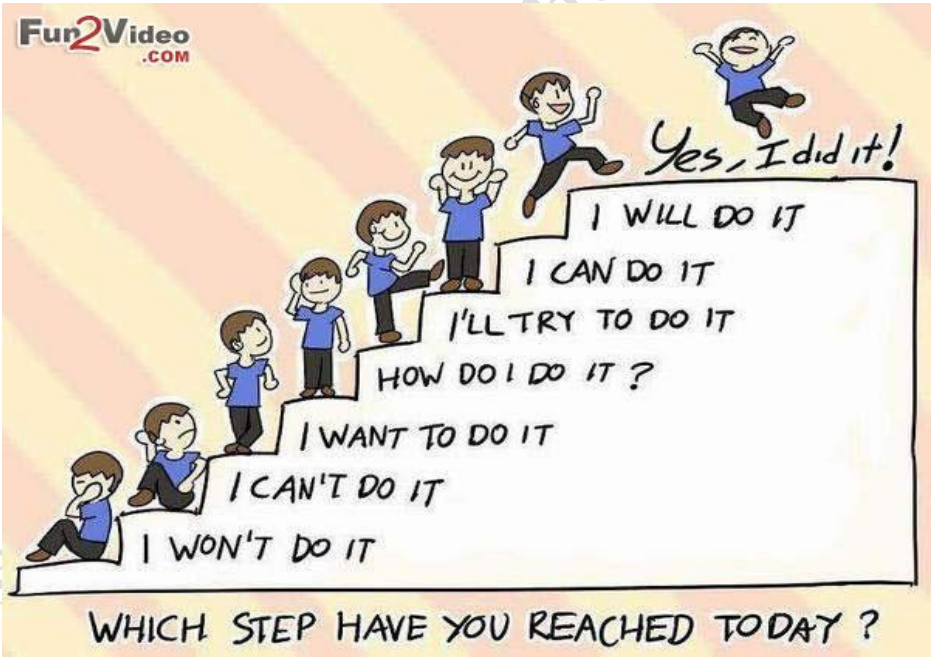
VARIANTA pentru trimiterea arhivei: http://www.wetransfer.com

Cursantii sunt incurajati sa analizeze si sa comenteze rezultatele obtinute, studiind si materialele indicate in bibliografie si anexe. **(+ Recapitulare Laboratoarele 1+2+3+4+5+6+7+8+9+10+11)** (Pentru Modeler, varianta "programare" C++: **OMNeT++ Network Simulation Framework** http://www.omnetpp.org/;

Obs:

Punctaj maxim (Data trimiterii temei)			
<= 6.01. 2020	**8.01. 2020**	**10.01.2020**	**12.01.2020**
100 pct	**80 pct**	**60 pct**	**50 pct**

Obs: Participarea (activa!) la Curs si Laborator permite, prin cunostintele acumulate, obtinerea unor rezultate bune si f. bune, asa cum ni le dorim cu totii.



Sursa: http://www.funnfun.in/wp-content/uploads/2013/06/steps-of-success-encouraging-quote.jpg

How to send an e-mail

http://lifehacker.com/5803366/how-to-send-an-email-with-an-attachment-for-beginners
https://support.google.com/mail/answer/6584?hl=en "As a security measure to prevent potential viruses, Gmail doesn't allow you to send or receive executable files (such as files ending in `.exe`)."
https://support.google.com/mail/answer/2480713?hl=en
http://fastupload.ro/free.php
http://www.computerica.ro/siteuri-transfer-fisiere-mari-upload/

Bibligrafie:

Lab_01, Lab_02, Lab_03, Lab_04, Lab_05, Lab_06, Lab_07, Lab_08, Lab_09, Lab_10, Lab_11, TL_01
http://www.cdsd.ro/cursuri

Python (Lab1, Lab2)
Using Python on Windows - https://docs.python.org/3/using/windows.html
The Hitchhiker's Guide to Python - http://docs.python-guide.org/en/latest/intro/learning/
A Byte of Python - https://www.gitbook.com/book/swaroopch/byte-of-python/details
GUI Programming in Python - https://wiki.python.org/moin/GuiProgramming
https://winpython.github.io/ ; https://www.python.org/
https://social.technet.microsoft.com/wiki/contents/articles/910.windows-7-enabling-telnet-client.aspx
http://www.telnet.org/htm/places.htm
rainmaker.wunderground.com : weather via telnet!

https://docs.python.org/3/library/socket.html
18.1. socket — Low-level networking interface

Java Sockets
http://download.oracle.com/javase/tutorial/networking/sockets/
http://www.oracle.com/technetwork/java/socket-140484.html

Pyton Sockets
http://docs.python.org/howto/sockets.html

C++ Sockets
http://www.linuxhowtos.org/C_C++/socket.htm
http://cs.baylor.edu/~donahoo/practical/CSockets/winsock.html

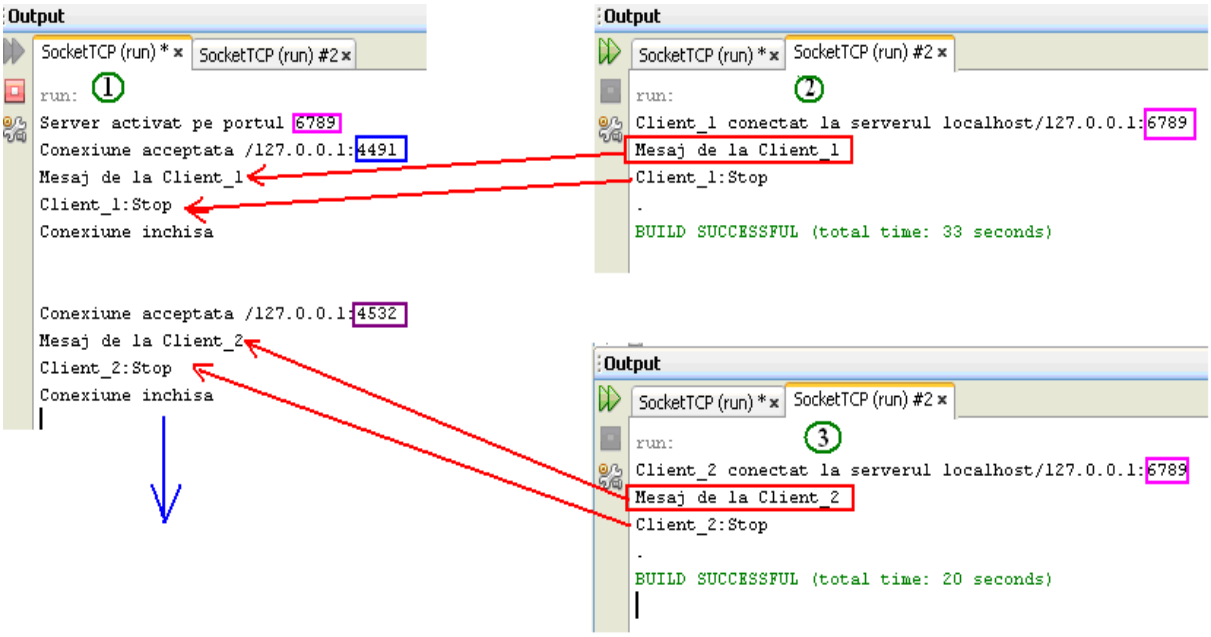
PHP Sockets
http://www.php.net/manual/en/book.sockets.php

Perl Socket
http://www.devshed.com/c/a/Perl/Socket-Programming-in-PERL/

Ruby Sockets
http://en.wikibooks.org/wiki/Ruby_Programming/Reference/Objects/Socket
https://www6.software.ibm.com/developerworks/education/l-rubysocks/l-rubysocks-a4.pdf
http://www.tutorialspoint.com/ruby/ruby_socket_programming.htm

etc....

Anexa 1 - Exemplu comentariu



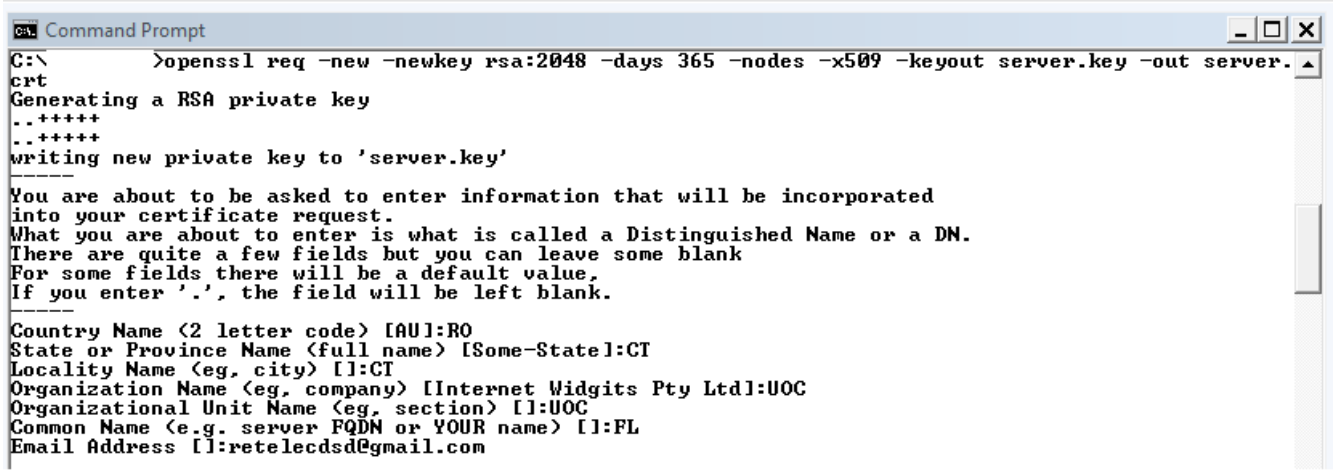
Anexa 2

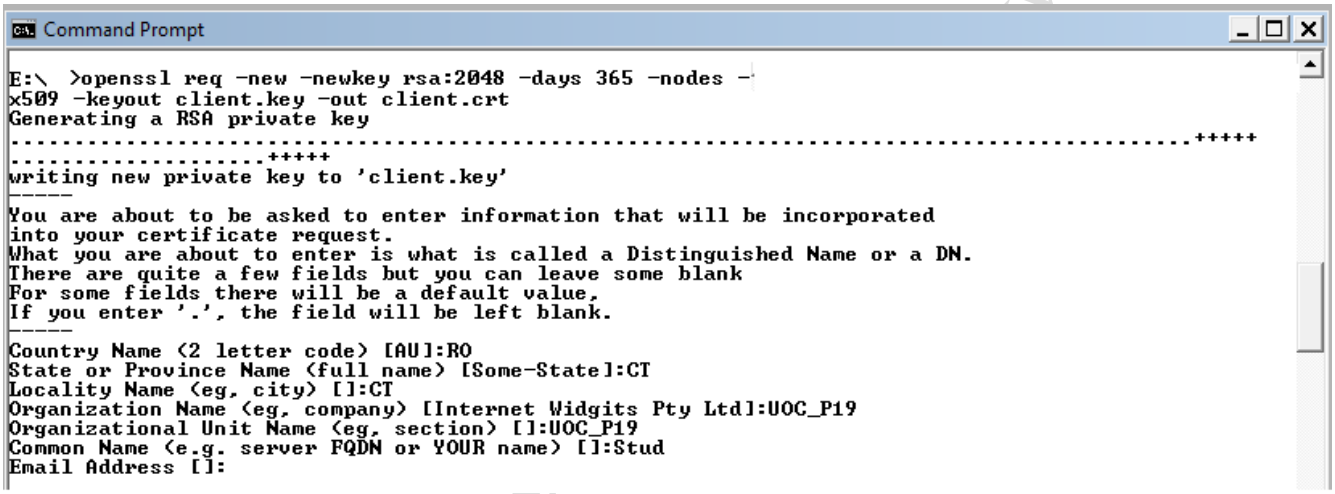
- **Create client-server application for web service in Java**
  https://www.codejava.net/java-ee/web-services/create-client-server-application-for-web-service-in-java
  https://docs.oracle.com/javaee/5/tutorial/doc/bnayn.html
- **Client-Side Web Programming**
  https://wiki.python.org/moin/WebClientProgramming
- **Simple HTTP server and client in Python**
  https://www.junian.net/2014/07/simple-http-server-and-client-in-python.html

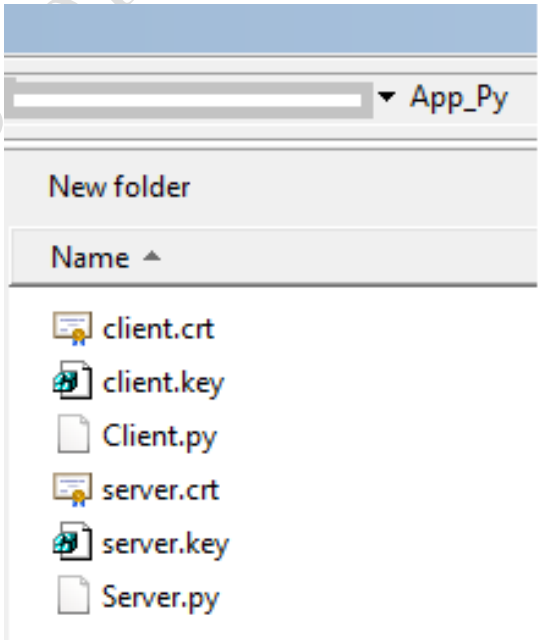
Anexa 3: The Programming Process

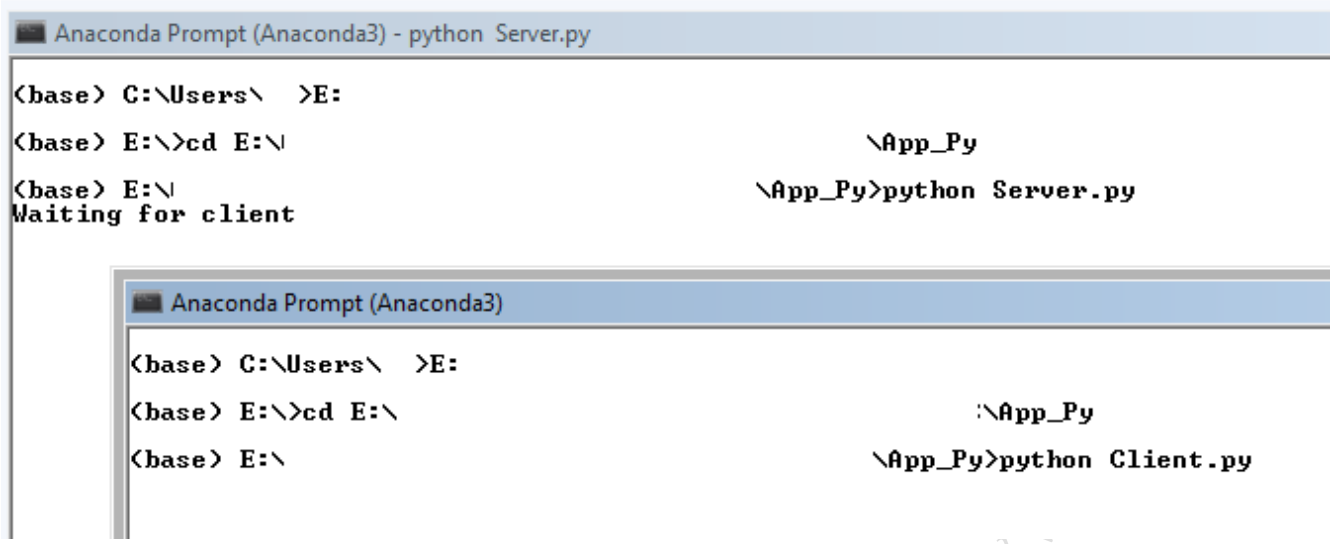
1. Identify the Problem - **What** Are You Trying To Do?
   - Requirements
   - Specification
2. Design a Solution - **How** Is It Going To Be Done?
3. Write the Program - **Teaching** the Computer
   - Code
   - Compile
   - Debug
4. Check the Solution - **Testing** it Understands You

Anexa 4:









Obs:

Thread-SSL-not-working-with-Python-3-7

https://python-forum.io/Thread-SSL-not-working-with-Python-3-7