

**Departamento de Informática**  
**Faculdade de Ciências e Tecnologia**  
**UNIVERSIDADE NOVA DE LISBOA**

Licenciatura em Engenharia Informática  
PROVA DE TESTE PRÁTICO – Redes de Computadores  
1º Semestre, 2003/2004

---

**NOTAS:**

Leia com atenção cada questão antes de responder. A interpretação do enunciado de cada pergunta é um factor de avaliação do teste.

Pode utilizar elementos pessoais de consulta. A duração do teste é 2h00.

O enunciado contém 6 páginas que devem ser entregues com a resposta ao teste.

---

**NOME:** \_\_\_\_\_ **Nº Aluno:** \_\_\_\_\_

**1)** Considere os programas apresentados nos anexos A.1 e A.2 usados para implementar um sistema cliente/servidor que permite verificar a validade de um par nome de utilizador/senha.

a. Indique qual dos programas é o cliente e qual é o servidor:

Classe A (anexo A.1): \_\_\_\_\_ Classe B (anexo A.2): \_\_\_\_\_

b. Assinale e corrija os erros existentes nos programas apresentados de forma a que o cliente comunique correctamente com o servidor (caso exista mais do que uma solução possível, escolha uma – a leitura e escrita de valores booleanos usando as classes `DataInputStream/OutputStream` estão correctas).

Indique se as afirmações seguintes são verdadeiras ou falsas, justificando as afirmações falsas:

c. O cliente pode começar a executar antes do servidor.

Verdadeiro                      Falso                      porque...

d. Dado o programa apresentado, é impossível saber qual o porto local do socket no cliente.

Verdadeiro                      Falso                      porque...

e. O cliente pode sempre contactar o servidor, independentemente da máquina em que o servidor executa.

Verdadeiro

Falso

porque...

- 2) Considerando o serviço descrito na pergunta anterior, apresente o código necessário à criação de um servidor concorrente.
- 3) Suponha que se pretende criar um proxy TCP para o serviço descrito nas perguntas anteriores. Esse proxy deve contactar sempre o mesmo servidor, cujo nome é o primeiro parâmetro do programa --- por exemplo, ao iniciar o programa com a seguinte linha de comando "java P forest.bears.org", o proxy criado deve contactar sempre o servidor "forest.bears.org".
- Complete o anexo B para implementar o proxy pretendido.
  - Descreva brevemente as modificações ao programa cliente apresentado anteriormente de forma a que este utilize o proxy em vez do servidor.

- 4) No âmbito do primeiro trabalho prático referente ao desenvolvimento de um cliente, servidor e proxy de HTTP, indique se as afirmações seguintes são verdadeiras ou falsas, justificando as afirmações falsas:
- Removendo as instruções de espera do servidor iterativo (`Thread.sleep(...)`), o servidor iterativo é necessariamente tão ou mais rápido que o servidor concorrente quando executam numa máquina com apenas um processador.

Verdadeiro

Falso

porque...

- O proxy desenvolvido pode receber um pedido HTTP com um URL relativo (por exemplo, /`index.html`).

Verdadeiro

Falso

porque...

- 5) Para as seguintes perguntas, indique a resposta que julga ser a mais correcta. Tenha atenção que respostas erradas podem retirar pontuação.

**a. No modelo cliente/servidor para aplicações baseadas nos protocolos da pilha TCP/IP é necessário que:**

- Os clientes conheçam o endereço IP e o process id (PID) do servidor.
- Os servidores conheçam previamente o endereço e porto do cliente
- Os clientes conheçam previamente o endereço e o porto do servidor
- O endereço IP do cliente deve corresponder a localhost, embora o porto possa ser qualquer

Resposta correcta: \_\_\_\_\_

**b. Na pilha TCP/IP, o protocolo HTTP**

- É um protocolo de aplicação suportado em TCP

2. É um protocolo de transporte diferente do TCP e que está vocacionado para envio e recepção de mensagens entre clientes e servidores WEB
3. É um protocolo de aplicação suportado em UDP
4. É um protocolo de transporte inspirado no protocolo TCP mas que pressupõe que todas as mensagens são enviadas para o porto 80

Resposta correcta: \_\_\_\_\_

**c. A resposta a um pedido de uma imagem (por HTTP 1.0) enviado por um browser WEB a um servidor ou a um proxy:**

1. Implica que o browser leia do seu socket uma única string, após o que, o servidor fecha a conexão.
2. Implica que o browser leia do seu socket uma sequência de strings até receber uma string vazia.
3. Implica que o browser leia do seu socket uma sequência de "byte arrays" terminando a recepção com o fecho da conexão pelo servidor ou pelo proxy.
4. Implica que o browser leia do seu socket uma sequência de strings até uma string vazia e depois uma sequência de "byte arrays" até que se dê o fecho da conexão pelo servidor ou pelo proxy.

Resposta correcta: \_\_\_\_\_

- 6)** Os programas dos anexos A.1 e A.2 foram corrigidos e modificados de forma a medirem o tempo necessário para verificar 100 vezes seguidas a validade de um par utilizador / senha passado em parâmetro, usando sempre o mesmo socket. Sabe-se também que os pacotes do cliente até ao servidor atravessam 10 routers, ligados por canais de muito alta velocidade, isto é, de cerca de 1 G bps. O servidor está situado no Alaska e o cliente num laboratório em Portugal. Sabe-se ainda que o método validLogin da classe MySystem do program do anexo A.2 executa sempre em memória em tempo constante e que o tempo de execução do servidor é constante.

Os tempos medidos tinham o valor médio de 800 mili segundos, o tempo mínimo de 100 mili segundos e o tempo máximo de 1600 milisegundos.

Quais dos seguintes tempos que caracterizam os canais e os routers da rede explicam uma tão grande variação do tempo necessário para fazer a verificação do par utilizador / senha ? Justifique a sua resposta.

- a. Tempo de propagação dos canais.
- b. Tempo de transmissão dos pacotes.
- c. Tempo de procesamento dos routers.
- d. Tempo em fila de espera dos pacotes nos routers.

Resposta correcta: \_\_\_\_\_, porque

## **ANEXO A.1**

```
import java.io.*;
import java.net.*;

class A {
    public static void main( String[] args) throws Exception {
        Socket sock = new Socket( "polar.bears.org", 8700);

        DataOutputStream os = new DataOutputStream( sock.getOutputStream());
        DataInputStream is = new DataInputStream( sock.getInputStream());

        os.writeBytes( args[0] + "\n" );
        os.writeBytes( args[1] + "\n" );

        sock.close();

        boolean b = is.readBoolean();

        System.out.println( b);
    }
}
```

## **ANEXO A.2**

```
import java.io.*;
import java.net.*;

public class B {
    public static void main( String argv[]) throws Exception {
        ServerSocket s_one = new ServerSocket( 8500);

        while( true) {
            Socket s_two = s_one.accept();

            DataInputStream is = new DataInputStream( s_two.getInputStream());
            DataOutputStream os = new DataOutputStream( s_two.getOutputStream());

            String s = is.readLine();
            String s2 = is.readLine();

            os.writeBoolean( MySystem.validLogin( s, s2));

            s_one.close();
        }
    }
}
```

```
class MySystem {
    /**
     * Devolve "true" se "pwd" for a password do utilizador "user".
     * Caso contrário devolve "false".
     */
    public static boolean validLogin( String user, String pwd) {
        ....
    }
}
```

### **ANEXO A.3**

## ANEXO B

```
import java.io.*;
import java.net.*;

public class P {
    public static void main( String argv[]) throws Exception {
        ServerSocket proxy = new ServerSocket( 8585);

        while( true) {
            Socket con = proxy.accept();

            DataInputStream is = new DataInputStream( con.getInputStream());
            DataOutputStream os = new DataOutputStream( con.getOutputStream());
            String s1 = is.readLine();
            String s2 = is.readLine();
            boolean reply = true;

            os.writeBoolean( reply);
            con.close();
        } // fim do ciclo WHILE
    } // fim do método main
}
```