

REDES DE COMPUTADORES – UNIOESTE – 2008

Prática – SMTP – Envio de e-mails

Agente usuário de correio: versão simplificada

Este laboratório está dividido em duas partes. Na primeira, você deverá usar o telnet para enviar e-mail manualmente através de um servidor de correio SMTP. Na segunda, você terá de escrever o programa Java que realiza a mesma ação.

Parte 1: Enviando e-mail com telnet

Tente enviar um e-mail para você mesmo. Isso significa que você precisa saber o nome do hospedeiro do servidor de correio do seu próprio domínio. Para encontrar essa informação, você pode consultar o DNS para buscar o registro MX que mantém informações sobre seu domínio de correio. Por exemplo, bob@someschool.edu possui o domínio de correio *someschool.edu*. O comando a seguir consulta o DNS para encontrar os servidores de correio responsáveis pela entrega de correio neste domínio:

```
nslookup -type=MX someschool.edu
```

Para a resposta a este comando, pode haver vários servidores de correio que entregam e-mail para as caixas de correio no domínio *someschool.edu*. Suponha que o nome de um deles é *mx1.someschool.edu*. Nesse caso, o seguinte comando estabelecerá a conexão TCP a este servidor de correio. (Note que a porta número 25 é especificada na linha de comando.)

```
telnet mx1.someschool.edu 25
```

Neste ponto, o programa telnet permitirá a você entrar com os comando SMTP e exibirá as respostas do servidor de correio. Por exemplo, a seqüência de comandos a seguir envia um e-mail da Alice para o Bob.

```
HELO alice
MAIL FROM: <alice@crepes.fr>
RCPT TO: <bob@someschool.edu>
DATA
From: teste@teste.com
To: alo@ha.com
Subject: teste
Ola, que tal?
.
QUIT
```

O protocolo SMTP foi originalmente projetado para permitir às pessoas interagirem manualmente com os servidores de correio em modo de conversação. Por essa razão, se você digitar um comando com sintaxe incorreta, ou com argumentos inaceitáveis, o servidor retornará uma mensagem reportando isso e permitirá que você tente novamente.

Para completar esta parte do laboratório, você deve enviar uma mensagem de e-mail para você mesmo e verificar se ela chegou.

Parte 2: Enviando e-mail com Java

A Java fornece uma API para interagir com o sistema de correio da Internet, que é chamado JavaMail. No entanto, não usaremos esta API, pois ela esconde os detalhes do SMTP e da programação de sockets. Em vez disso, você escreverá um programa Java que estabelece uma conexão TCP com um servidor de correio através da interface de socket e enviará uma mensagem de e-mail.

Você pode colocar todo seu código dentro do método principal de uma classe chamada *EmailSender*. Execute seu programa com o simples comando a seguir:

```
java EmailSender
```

Isso significa que você incluirá em seu código os detalhes da mensagem de e-mail que você está tentando enviar.

Aqui está um esqueleto do código que você precisará para escrever:

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

public class EmailSender
{
    public static void main(String[] args) throws Exception
    {
        // Estabelecer uma conexão TCP com o servidor de correio.

        // Criar um BufferedReader para ler a linha atual.
        InputStream is = socket.getInputStream();
        InputStreamReader isr = new InputStreamReader(is);
        BufferedReader br = new BufferedReader(isr);
        // Ler os cumprimentos do servidor.
        String response = br.readLine();
        System.out.println(response);
        if (!response.startsWith("220")) {
            throw new Exception("Resposta 220 nao recebida do servidor.");
        }

        // Pegar uma referência para o trecho de saída do socket.
        OutputStream os = socket.getOutputStream();

        // Enviar o comando HELO e pegar a resposta do servidor.
        String command = "Helo localhost\r\n";
        System.out.println(command);
        os.write(command.getBytes("US-ASCII"));
        response = br.readLine();
        System.out.println(response);
        if (!response.startsWith("250")) {
            throw new Exception("Resposta 250 nao recebida do servidor.");
        }

        // Enviar o comando MAIL FROM.
        // Enviar o comando RECP TO.
        // Enviar o comando DATA.
        // Enviar o dados da mensagem.
        // Terminar com uma linha de um único período.
        // Enviar o comando QUIT.
    }
}
```