

Problema 1: Lava duto

Cleybson C. Leite

¹Universidade Estadual de Feira de Santana (UEFS)
Av. Transnordestina – Feira de Santana – BA – Brazil

cleybsonc@gmail.com

Abstract. *This report talks about the software created to solve the problem of file transfer request by the federal police intelligence sector, describing the technologies used in each application, the server and client, so that the problem was resolved.*

Resumo. *Este relatório fala sobre os software criados para resolver o problema de transferência de arquivos pedido pelo setor de inteligência da polícia federal, descrevendo as tecnologias utilizadas em cada aplicação, no servidor e no cliente, para que o problema fosse resolvido.*

1. Introdução

Após a segunda guerra mundial, o desenvolvimento da eletrônica teve um grande crescimento, principalmente na área da ciência de computação, que conseguia interceptar e decodificar as comunicações e mensagens em códigos cifrados do eixo nazi-fascista, fazendo com que eles se preparassem e pre-vessem os passos dos inimigos.

Mesmo com o fim dos conflitos, o desenvolvimento de novas tecnologias não teve fim, e com isso foi crescendo e se desenvolvendo em vários países, varias inovações. Até que na década de 60, foi criado um terminal interativo, em que o usuário acessava remotamente o computador da rede telefônica, em que nessa época era o mais se utilizava para a comunicação entre pessoas.

Com o desenvolvimento das tecnologias e o surgimento de microcomputadores, tornou-se algo mais popular e eficaz para a compra e utilização, porém ainda eram muito caros e isso fez com que nem todos tivessem acesso. E somente em 1969, surgiu a necessidade de conectar computadores entre si. E a ARPANET, foi a grande precursora da internet, conectando 4 computadores a uma rede. E com o tempo foi sentindo-se a necessidade de conectar mais computadores, para que eles pudessem compartilhar informações entre si, de forma rápida e pratica.

Com essa ideia de compartilhar informações de maneira pratica, rápida e segura, que o setor de inteligencia da policia federal, sentiu a necessidade de um programa que compartilhasse arquivos confidenciais para os seus usuários.

2. Fundamentação teórica

A aplicação desenvolvida deve ser feita de maneira que o usuário possa ter acesso aos arquivos confidenciais disponibilizados, para isso, foi criado 2 aplicações: o servidor, esta aplicação levanta um servidor e faz comunicação direta com os usuários através da outra aplicação, cliente; A aplicação cliente é responsável por fazer todas as requisições e fazer com quem o servidor se comunique com o usuário.

Para que a comunicação fosse feita entre os computadores, foi utilizado o *TCP* (Transmission Control Protocol - Protocolo de Controle de Transmissão), que é um conjunto de protocolos (linguagem utilizada para a comunicação entre dois computadores), que se dividem em 4 camadas: aplicação, transporte, rede e interface. A **camada de aplicação**, de acordo com [Tanenbaum 2003], esta camada contém todos os protocolos de nível mais alto, nele é possível enviar e receber informações de outros programas através da rede. A **camada de transporte**, que é responsável por receber dados pela camada de aplicação, verificar se os dados estão corretos e dividir em pacotes. A **camada de rede** recebe os pacotes feitos pelo protocolo de transporte e são anexados em um ip.

Para a comunicação entre os computadores, nesta aplicação foi utilizado o socket, que é a base da comunicação em redes TCP. Que serve para conectar elementos diferente e fazer com que eles se comuniquem, como por exemplo, cliente e servidor.

Outra tecnologia adicionada junto ao socket, foi o threads, que é uma estrutura que executa simultaneamente com o processo do processo que gerou a thread, que nesse caso seria o servidor, ou seja, ele executa ao mesmo tempo, enquanto o servidor estiver sendo executado.

3. Metodologia

A linguagem utilizada para o desenvolvimento da aplicação foi java, e como dito na seção anterior, foram criadas 2 aplicações, o servidor e o cliente.

3.1. Servidor

Esta classe inicialmente abre uma porta, que funciona como um processo do computador, e a criação de um socket pois, como dito anteriormente, servidor e cliente são elementos diferentes, e necessita usar o socket para que eles se comuniquem. Outra tecnologia necessária foi o thread, pois como o software iria lidar com vários cliente, e se não utilizasse esse método, o servidor só iria suportar um cliente por vez, e assim, não atenderia os requisitos pedidos no problema.

Então o servidor ficou responsável somente por esperar os cliente, já que esta função pode ser feita de maneira que os usuários entrassem no servidor um de cada vez. E ao entrar no servidor, ele seria direcionado para uma porta nova, através da thread que é criada ao receber cada cliente.

3.1.1. Thread

Essa estrutura permite que instruções de uma aplicação sejam executadas ao mesmo tempo de outros processos, isso permite então que o servidor seja responsável somente por aceitar o cliente, e uma classe que implemente o runnable seja executada ao mesmo tempo, fazendo com que todos os procedimentos do cliente sejam feitos nessa classe. Então a parte de login/logout/cadastrar usuário, adicionar/baixar/remover arquivos, sair do programa, sejam responsáveis por ela.

3.1.2. Usuários

Esta classe foi implementada no servidor, pois seria mais fácil, visto que todos os dados do sistema estão salvo no servidor, como a lista de usuários cadastrados. Além disso, esta classe é responsável por 2 dados muito importantes: dizer se o usuário está online ou offline, visto que, uma conta só pode estar logar em um local por vez; Outra informação importante é se o usuário é especial ou não, isso interfere no momento de adicionar/remover arquivo, que são funções que só podem ser feitas caso o usuário seja especial.

3.1.3. Serializable

Foi implementado o serializable, para que não se perdesse os dados de que já está cadastrado no sistema, então toda vez que um usuário novo é cadastrado, programa serializa a lista de usuários e salva, e toda vez que o programa é inicializado, esta lista é lida, assim, sempre tem uma lista atualizada dos usuários.

3.2. Cliente

Esta foi outra aplicação, feita somente para a interação do usuário com o servidor, então ela não guarda nenhuma informação tão relevante, até mesmo por meio de segurança, visto que o usuário vai ter total acesso a essa aplicação. Então basicamente essa aplicação segue um ciclo, recebe uma informação e retorna outra, e caso seja um download, adicionar arquivo, que é algo mais específico, ele foge desse ciclo até concluir a ação, depois retorna ao ciclo novamente.

3.3. Transferência de arquivo

Para que a transferência ocorra, no momento do download, o cliente escolhe o arquivo, entre os que estão na pasta "programa lava duto", e com isso ele pode navegar entre as pastas áudio, música, vídeo, etc. A ideia de criar estas pastas, foi por questão de segurança, pois, limitaria o acesso do usuário, e assim impediria que ele navegasse em todo sistema e fizesse algo que arriscasse a segurança do servidor. Logo após o usuário escolher qual pasta quer salvar o arquivo, navegando entre as pastas do seu sistema operacional.

No momento de adicionar um arquivo no servidor, o usuário tem que ser especial. Para que a transferência ocorra, o usuário tem que estar na pasta do servidor em que ele deseja que salve o arquivo ao fazer o upload, e digitar o comando "add". Logo após o programa mostra as pastas do sistema operacional do usuário, e o cliente navega entre as pastas digitando o respectivo nome da pasta ou arquivo, se for pasta o programa entra na mesma, se for arquivo, o programa faz o upload.

Para ocorrer essas transferências, o arquivo a fazer o download ou o upload, o arquivo é serializado em um array de byte, e somente depois ele é enviado, e convertido novamente em um arquivo.

4. Resultados e discussões

Para a navegação entre as pastas é feita utilizando uma pilha, em que toda pasta que o usuário entrar, o nome da mesma é adiciona na pilha e ao voltar o ultimo é retirado. Assim nunca perdemos o endereço da pasta atual.

Após o usuário logar no sistema, a pasta "programa lava duto" é lida e exibida para o usuário, então para que o programa funcione, essa pasta tem que estar no local certo.

Outra questão foi a respeito de deletar um arquivo enquanto o download está sendo feito, e durante os testes, percebeu-se que isso não era possível, pois o arquivo estava em uso e o sistema operacional não permite que ele seja deletado. E caso o usuário seja desconectado durante a transferência do arquivo, a transferência é interrompida. O mesmo acontece quando o servidor fica offline, a transferência é interrompida e o usuário é desconectado e encerra o software cliente, isso acontece em qualquer momento que o servidor ficar offline.

5. Conclusão

Todos os objetivos foram alcançados, tivemos problemas na transferência entre dois computadores, mas foi resolvido após colocar para o programa antes de enviar o arquivo, ele aguarda-se um tempo, e somente depois começa a enviar. Uma melhoria é que o programa mostra a porcentagem da transferência, para o usuário ter uma noção do tempo que falta para a transferência.

Referências

Tanenbaum, A. S. (2003). *Redes de Computadores*. Editora Campus (Elsevier), 4th edition.