

Bacharelado em Ciência da Computação - UFU
Disciplina - Arquitetura de Redes TCP/IP - 2015/2
Relatório do Trabalho 1

Membros do grupo: Bruno Well Dantas Moraes - 11111BCC008
Frederico Franco Calhau - 11421BCC031
Guilherme Martini
Luiz Guilherme de Souza Pelegrini - 11111BCC024

Resumo do projeto

Foi desenvolvido um programa em linguagem C com o objetivo de colocar em prática os conceitos vistos em aula sobre a camada de aplicação: interface com a camada de transporte e protocolo de aplicação.

1. Introdução

Este projeto consiste no desenvolvimento de um programa distribuído com o objetivo de criar um protocolo na camada de aplicação e implementar uma versão deste programa utilizando o protocolo TCP e outra utilizando o UDP.

Neste relatório são apresentados: uma descrição da estrutura do aplicativo, um esquema que ilustra o protocolo utilizado, os resultados obtidos e a conclusão do grupo.

2. Aplicação desenvolvida

Foi desenvolvida uma aplicação distribuída de agenda telefônica na linguagem C com as funcionalidades: armazenar ou alterar um registro, remover um registro, acessar um registro, finalizar a aplicação.

O programa opera sobre um protocolo de aplicação que comporta as funcionalidades acima, que será descrito na seção 3.

A aplicação foi estruturada de forma que o cliente recebe comandos do usuário via teclado e os repassa ao servidor sem processamento adicional, enquanto o servidor realiza a tarefa de processar as requisições de acordo com o protocolo. Para isso, no lado do servidor, foi utilizada uma lista dinâmica para armazenamento dos registros, que por sua vez, foram implementados por uma estrutura de dados que contém duas strings: nome e telefone.

Foram definidas três funções para a comunicação com a camada de transporte:

- conectar: renomeada como “inicializar” na versão que utiliza UDP. A função chama as funções do sistema responsáveis pela alocação de socket, associação do socket ao endereço e porta do host e, no caso do TCP, a conexão entre o par de hosts.

- enviarmsg: invoca a função send no TCP ou sendto no UDP, recebendo como parâmetro a string a ser enviada ao outro host.

- recebermsg: invoca a função recv no TCP ou recvfrom no UDP, armazenando no buffer a mensagem recebida.

3. Aspectos de Implementação

O protocolo de aplicação implementado conta com um caractere “/” para separação de campos na mensagem e um conjunto de palavras que descrevem as funcionalidades do sistema.

O cliente e o servidor podem ser vistos como autômatos em relação ao protocolo:

Cliente:

Estado Inicial:

Envia “STORE/nome/telefone/” → estado Armazenar

Envia “REMOVE/nome/” → estado Remover

Envia “GET/nome/” → estado Acessar

Envia “END/” → estado Encerrar

Estado Armazenar:

Recebe “OK/” → estado Inicial

Recebe “UPDATE?/” → estado Atualizar

Estado Atualizar:

Envia “YES/” → estado Inicial

Envia “NO/” → estado Inicial

Estado Remover:

Recebe “OK/” → estado Inicial

Recebe “NOT FOUND/” → estado Inicial

Estado Acessar:

Recebe “telefone/” → estado Inicial

Recebe “NOT FOUND/” → estado Inicial

Estado Encerrar:

Recebe “END/”, encerra aplicação

Servidor:

Estado Inicial:

Recebe “STORE/nome/telefone/” → estado Armazenar

Recebe “REMOVE/nome/” → estado Remover

Recebe “GET/nome/” → estado Acessar

Recebe “END/” → estado Encerrar

Estado Armazenar:

Chave não consta na lista, envia “OK/” → estado Inicial

Chave consta na lista, envia “UPDATE?/” → estado Atualizar

Estado Atualizar:

Recebe “YES/”, sobrescreve o telefone do registro → estado Inicial

Recebe “NO/” → estado Inicial

Estado Remover:

Chave consta na lista, envia “OK/” → estado Inicial

Chave não consta na lista, envia “NOT FOUND/” → estado Inicial

Estado Acessar:

Chave consta na lista, envia “telefone/” → estado Inicial

Chave não consta na lista, envia “NOT FOUND/” → estado Inicial

Estado Encerrar:

Envia “END/”, encerra aplicação

4. Resultados e Análise

As funcionalidades de ambas as versões do aplicativo foram testadas em LAN via linha de comando e todos os testes tiveram êxito.

5. Conclusão

O propósito de construir um programa que utiliza um protocolo na camada de aplicação e faz interface com a camada de transporte foi alcançado após algumas dificuldades técnicas. Este trabalho constituiu um aprendizado prático sobre o mecanismo de troca de mensagens numa aplicação distribuída.