

Trabalho Conjunto entre as Disciplinas de Sistemas Operacionais e Fundamentos de Redes de Computadores

Professores: Cristina Nunes, Miguel Xavier

Serviço de bate-papo baseado do protocolo IRC

Objetivo

O objetivo geral do trabalho é desenvolver uma aplicação usando *sockets* UDP e/ou TCP para prover um serviço de bate-papo simplificado inspirado no protocolo IRC (*Internet Relay Chat*). Os objetivos específicos incluem:

- compreender de maneira prática o mecanismo de comunicação por *sockets*;
- desenvolver uma aplicação distribuída usando o modelo cliente/servidor;
- implementar comunicação entre os processos usando *sockets* UDP e/ou TCP.

Descrição

O serviço de bate-papo deverá ser implementado utilizando o modelo cliente/servidor. Um programa servidor irá gerenciar canais de bate-papo e permitir que usuários se comuniquem através desses canais, de forma similar ao serviço IRC (*Internet Relay Chat*). Assim, toda a comunicação entre usuários deverá passar pelo servidor. Os usuários se conectam ao servidor através de um programa cliente. Esse programa deve ler e interpretar os comandos digitados pelo usuário e enviar uma mensagem correspondente ao servidor.

A lista completa de comandos disponíveis para o usuário e mensagens enviadas ao servidor está descrita no ANEXO I. O programa cliente também deve receber mensagens enviadas pelo servidor e mostrá-las ao usuário.

Sugestão: para facilitar o desenvolvimento, inicialmente a lista de canais pode ser criada estaticamente no servidor (criar pelo menos 3 canais de bate-papo para testes).

Resultados e Entrega

Grupos: Individual ou em dupla.

Entrega: Upload no Moodle de um arquivo zipado com o nome do(s) aluno(s), contendo:

1. Relatório descrevendo a implementação
2. Código da implementação

Prazo final para entrega: 07/06 até as 19:30 (antes do início da aula)

Data da apresentação: 07/06

ANEXO 1

Descrição do Protocolo Cliente/Servidor do Serviço de Bate-papo

O serviço de bate-papo implementado utilizará uma versão simplificada do protocolo IRC (*Internet Relay Chat*), a qual é descrita na tabela a seguir.

Comando digitado pelo usuário	Mensagem enviada ao servidor	Descrição
/nick <nickname>	NICK <nickname>	Solicita a alteração do apelido do usuário.
/create <channel>	CREATE <channel>	Solicita a criação de um novo canal no servidor. O usuário que criar o canal será o administrador do mesmo. Apenas o administrador pode realizar operações do tipo REMOVE e KICK. O apelido do administrador do canal deve sempre ser impresso com um * na frente.
/remove <channel>	REMOVE <channel>	Solicita a remoção de um canal. Todos os usuários que fazem parte do canal devem ser informados. Apenas o administrador do canal pode realizar essa operação.
/list	LIST	Solicita a lista de canais disponíveis no servidor.
/join <channel>	JOIN <channel>	Solicita a participação em um canal. A partir desse momento, o usuário deve passar a receber todas as mensagens enviadas ao canal.
/part	PART <channel>	Solicita a saída do canal atual. A partir desse momento, o usuário não deve receber mensagens enviadas ao canal.
/names	NAMES <channel>	Solicita a lista de usuários que fazem parte do canal atual. Necessário ter executado um comando /join anteriormente.
/kick <channel> <nickname>	KICK <channel> <nickname>	Solicita a remoção de um usuário de um canal. Somente o administrador do canal pode realizar essa operação.
/msg <nickname> <message>	PRIVMSG <channel> <nickname> <message>	Envia uma mensagem privada para um usuário. Somente o usuário especificado pelo <nickname> deve receber a mensagem.
<message>	MSG <channel> <message>	Envia uma mensagem o canal atual. Necessário ter executado um comando /join anteriormente.
/quit	QUIT	Desconecta completamente do serviço de chat. Se o usuário estiver participando de algum canal, ele deve ser removido deste canal antes de fechar o programa.

Observações:

- Todos os comandos reservados digitados pelo usuário devem conter o caractere / no início da palavra (ex: /join);
- Comandos sem o caractere / indicam uma mensagem para o canal;
- Nomes de canais e apelidos não devem conter espaços;
- Todos os nomes de canais devem ter o caractere # no início da palavra (ex: #linux);
- Cada mensagem recebida de um usuário deve aparecer com o nome do remente entre < e > na frente (ex: <maria> Olá pessoal.);

Alguns exemplos de saída do programa:

/nick jose

Você alterou o nickname para jose.

/list

Lista de canais no servidor:

#pucrs

#facin

#portoalegre

#linux

/join #linux

Você entrou no canal #linux.

/names

Lista de usuários no canal #linux:

*jose

maria

pedro

/msg maria Olá Maria, tudo bem?

<maria> Olá José, tudo ótimo.

Olá tudo mundo.

<pedro> Olá José!

<maria> Olá de novo.

/part

Você saiu do canal #linux.

Sugestão:

Quem quiser mais exemplos, pode utilizar instalar um cliente IRC e conectar em um servidor real (existem vários disponíveis na Internet). Também é possível utilizar IRC on-line no servidor FreeNode, através do seguinte endereço: <http://webchat.freenode.net/>

No entanto, é importante lembrar que a versão implementada nesse trabalho é uma simplificação didática do serviço real.