

Protocolos utilizados no ep2 de MAC448

André Pereira Mesquita & Renato Scaroni

26 de novembro de 2013

Criando conexão

O cliente inicia enviando para o servidor qual o tipo de conexão vai fazer (tcp ou udp), junto com o nome do usuario e a porta a ser usada para conexão p2p. O servidor lança então uma thread separada para tratar as mensagens enviadas pelo usuário. Do lado do cliente, são criadas duas threads, uma que escuta inputs de usuário e os envia para o servidor e outra que escuta mensagens vindas do servidor.

Tratando as mensagens

Após a conexão ser estabelecida, o tratamento das mensagens é feito da seguinte forma:

- O servidor recebe a mensagem do cliente e manda a mesma para um método parser no próprio servidor que vai cuidar de escolher que mensagem deve ser enviada de volta para o cliente. O próprio parser utiliza o socket guardado na lista de conexões para enviar a mensagem.
- O cliente recebe a mensagem de resposta do servidor e decide entre mostrar na tela para o usuário a mensagem ou tomar alguma ação interna. A decisão do cliente ao receber a mensagem vai depender se a mensagem começa com uma de três palavras reservadas: ?Alive?, <connect> ou <release>

Comunicando o servidor

Quando um programa cliente capta da entrada padrão que deve se conectar com um outro cliente ele envia uma mensagem ao servidor através da thread que escuta a entrada do usuário, dizendo que quer se conectar com o usuário passado como parametro ao comando connect, bem como a porta a que o mesmo deve responder ao pedido de conexão. Em seguida cria uma thread para esperar a conexão do usuário desejado e marca a thread que recebe inputs de usuário para não mais aceitá-los. O servidor vai associar a cada cliente uma string com o nome do cliente ao qual ele está conectado, sinalizando que eles já estão em conexão com algum cliente. Importante notar que a thread que rece mensagem do servidor ainda estará ativa para responder ao heartbeat que o servidor enviará a cada 65 segundos, caso não receba nada do cliente.

Resposta do servidor

Ao receber a mensagem iniciada com a palavra connect de um cliente, o servidor também vai receber o nome do usuário que vai receber a conexão, bem como a porta em que o mesmo deve se conectar no cliente que pediu a conexão. Em seguida o servidor monta uma mensagem contendo o nome do usuário que solicitou a conexão, seu endereço na rede e a porta em que estará escutando a resposta, bem como uma palavra reservada pedindo para o cliente que deve receber o pedido de conexão se conectar com o primeiro.

Iniciando conexão

O cliente que receberá o pedido de conexão, receberá a mensagem montada pelo servidor, descrita no item anterior, e criará dois sockets, um para receber e outro para enviar as mensagens. Esses sockets se conectarão com o outro cliente na porta que ele sinalizou no servidor e marca a quem está conectado, guardando o nome do usuário do cliente que pediu a conexão. Por fim, lança duas threads: uma escuta entrada de usuário e envia as mensagens para outro cliente e a outra para esperar mensagens do outro cliente.

Fim do protocolo de conexão

O cliente que enviou ao servidor o pedido de conexão estará escutando pelos dois sockets criados pelo outro cliente. assim que receber ambos salva a referencia de cada de forma que possa enviar e receber mensagem. Após receber os sockets, são lançadas mais duas threads: uma escuta entrada de usuário e envia as mensagens para outro cliente e a outra para esperar mensagens do outro cliente.

iniciando o protocolo de encerramento de conexão

O cliente que deseja encerrar a conexão recebe o input do usuário (consistindo apenas do comando <close>. Captando isso, a thread que envia mensagem para outro cliente fecha o socket, envia a mensagem para o outro cliente dizendo que quer terminar a conexão, envia uma mensagem para o servidor dizendo que quer terminar a conversa, fecha o socket e encerra a thread. O cliente do outro lado recebe a mensagem, encerra a thread e fecha o socket também.

Servidor

O servidor recebe a mensagem e envia para ambos os clientes uma mensagem para que a thread que escuta input de usuário e envia mensagens ao servidor seja liberada. Os clientes recebem essa mensagem e liberam essas threads, permitindo que possam acessar o servidor novamente.

The End