

UNIVERSIDADE FEDERAL DA FRONTEIRA SUL CAMPUS CHAPECÓ CURSO DE CIÊNCIA DA COMPUTAÇÃO

DEIVIDI LUIZ PINTO DA SILVA LUIS HENRIQUE WASEM DE OLIVEIRA

RELATÓRIO - CHAT UTILIZANDO PROTOCOLO MQTT

DESCRIÇÃO DO PROJETO

O projeto foi desenvolvido com o protocolo MQTT, onde este possibilitou a comunicação entre diversos usuários que utilizam a aplicação para que simule a conversa entre os mesmos, semelhante a um "bate-papo", conhecido também como *chat*. Para o desenvolvimento do código, foi utilizado a linguagem de programação "C". As funções implementadas são:

- Conexão com novos usuários
- Listagem de usuários
- Chats criados
 - Conversas de chats particulares (one to one)
 - Conversa de chats em grupo (N para N)
- Bloqueio de usuários
- Desbloqueio de usuários
- Criação de grupo
- Administração dos grupos
 - Adicionar usuários ao grupo
 - Listar usuários do grupo
 - Excluir usuários do grupo
 - Deletar grupo
 - Sair do grupo
- Desconexão

Cada uma das funcionalidades serão detalhadamente explicadas no decorrer do relatório. Para o funcionamento geral do projeto, foi necessário trabalhar com bibliotecas que possibilitem algumas manipulações, listagem abaixo de algumas bibliotecas utilizadas:

- stdio.h (entrada e saída de dados digitados)
- stdlib.h (realizar conversões entre tipos)
- string.h (trabalhar com strings)
- time.h (tempo da aplicação)
- stdbool.h (trabalhar com variáveis verdadeiro e falso)
- MQTTAsync.h (arquivo assíncrono para utilizações do funcionamento do MQTT)
- pthread.h (utilizar thread para o timer)

Para armazenamento dos usuários ativos, bem como cada chat criado, está sendo utilizado vetores de char, para que seja manipulado toda vez que haja alguma alteração no decorrer da aplicação. Quando um novo terminal executa o chat, é necessário a informação de um *username* (nome de usuário), usado como identificação da pessoa que está utilizando a aplicação, este por sua vez, tem um tópico de controle, denominado "control_username", onde este é assinado pelo usuário. Este tópico é concatenado e criado ao informar seu *username*, assim toda

vez que outro usuário desejar criar uma conexão ou comunicar-se diretamente, é necessário a publicação no tópico de controle do usuário que tem interesse.

Quando inicializado um novo chat, o usuário pode operar nele a todo momento, até que o mesmo seja finalizado, pode-se ficar offline para simular uma desconexão, sem finalizar esta execução.

Arquitetura do sistema/aplicativo

A comunicação entre usuários é inicializada após um deles, publicar no tópico de controle do usuário alvo, assim o mesmo pode aceitar uma nova conexão, ou rejeitá-la. Esta funcionalidade é chamada sempre que deseja criar uma conexão particular (*one to one*) ou para adicioná-lo a um grupo existente, de acordo com a resposta do usuário a aplicação segue o roteiro escolhido. Cada usuário possui um tópico de controle, denominado "control_username", este por sua vez, é inicializado sempre que um novo *chat* for inicializado, e desta forma é possível contatar novos usuários. Quando o convite de conexão é aceito, um novo tópico é criado, por exemplo, "X_Y_TIMESTAMP", onde X representa o usuário que aceitou a conexão, Y representa o usuário que enviou o convite, e TIMESTAMP representa o instante de tempo da aplicação em que houve a criação do tópico. Desta maneira, sempre que uma nova conversa for instanciada entre os mesmos usuários, o TIMESTAMP irá ser alterado.

Para a comunicação entre usuários em determinado grupo, após a confirmação do convite do usuário, ele é inscrito no tópico que contém o nome do grupo, por exemplo, "NAME_GROUP", desta maneira é possível diferenciar e saber quando há conversas particulares ou em grupos. Para todo e qualquer usuário que aceita o convite, é inscrito no tópico do grupo, além disso, o mesmo possui seu status de grupo definido como *USER*, já o dono do grupo, seu *status* é definido como *ADMIN*, diferenciando assim na listagem de usuários contidos dentro do grupo. Este *status* possibilita a diferenciação de opções de cada usuário, visto que, o administrador do grupo possui permissões que usuários comuns não possuem.

Outro tópico criado na inicialização da aplicação, assinado individualmente por cada usuário, denomina-se, *USER_STATUS*, este por sua vez, possibilita a divulgação de seu *status* durante a execução da aplicação, como *status* de *ONLINE* e *OFFLINE*, desta maneira é possível manter atualizado, visto que, todos recebem a publicação de uma nova mensagem quando necessário. Estas alterações alteram a informação da lista de usuários em que eles estão salvos, podendo ser visualizado sempre que listar os usuários.

Abaixo serão explicados o funcionamento geral de cada função/funcionalidade implementada, com seu nome normal e (seu nome no código):

Listar usuários (listUsers()):

Esta função irá realizar a listagem de todos os usuários que inicializam a aplicação, com seu respectivo *status* de conexão e seu *status* de bloqueio.

Listar usuários do grupo (listUsersGroup()):

Esta função irá realizar a listagem de todos os usuários que estão contidos no grupo, toda vez que o usuário é incluído em um grupo, seu nome, nome do grupo, e *status* de *level* do grupo são salvos em uma

lista, desta maneira é percorrida toda a lista e realizando a comparação do nome do grupo que está sendo listado com o nome do grupo que foi salvo na lista. Assim é possível saber quais usuários estão contidos no grupo. Outra verificação é realizada quando este usuário foi encontrado, percorre novamente a lista de usuários para pegar seu status de conexão, listando todas as informações em tela.

• Divisão da *string* recebida (splitString()):

Esta função irá realizar a divisão de toda a *string* recebida, baseada no caractere informado, com ela é possível também definir qual posição da lista deseja obter-se o retorno, assim pode-se passar uma mensagem específica para esta função e obter como retorno somente o texto de interesse.

Exemplo: Sim X_Y_TIMESTAMP

Chamando a função *splitString(exemplo, " ", "1")*, desta maneira estamos informando para a função a mensagem exemplo, que contém o texto acima, e com isso queremos que realize a separação baseada pelo caractere espaço, buscando como interesse a posição um da lista, assim como retorno, iremos ter *X_Y_TIMESTAMP*. Desta maneira conseguimos extrair o tópico que foi criado, bem como a resposta Sim do usuário, basta passar como parâmetro a posição zero. Esta função realiza a divisão do caractere informado a partir de qualquer mensagem, basta informar os parâmetros desejados e terá como retorno o texto de interesse.

Enviar a mensagem de conexão (sendMessage()):

Esta função é responsável pela primeira conexão com o usuário, se o usuário alvo deseja estabelecer uma nova conexão com quem instanciou o chat. Esta função também realiza a conexão deste usuário com um grupo específico. Baseado no parâmetro informado ao chamá-la, como "", vazio, indica que é uma conexão entre usuários particulares (*one to one*), caso não vier o parâmetro vazio, é realizada a conexão para o grupo.

Deletar um usuário (deleteUser()):

Esta função irá realizar a busca do usuário que se tem como interesse em remover do grupo, comparando primeiramente o nome do grupo na lista em que foi salvo, após encontrá-lo, irá realizar uma nova busca pelo nome do usuário que se deseja remover, assim é criado uma nova mensagem informando que o usuário foi removido e quem o removeu, esta mensagem é enviada para o tópico do grupo, assim, todos que possuem inscrição neste tópico irão receber a mensagem, para poder removê-lo da lista.

Sair do grupo (exitGroup()):

Esta função irá remover do grupo o usuário que resolver sair, assim é criado uma mensagem e enviado para o tópico do grupo, informando a todos que o usuário em questão está saindo do grupo. Toda a lista em que estava o grupo, com demais informações dos outros usuários, são limpas.

Deletar o grupo (deleteGroup()):

Esta função irá deletar o grupo selecionado, bem como, remover todos os usuários que compõem o grupo. Uma mensagem de remoção de usuário é enviada para o tópico do grupo e todos que estão inscritos recebem, limpando a lista de componentes do grupo e a lista que contém o chat.

Administrar os grupos (adminGroup()):

Esta função permite que o usuário possa administrar os grupos que ele está, como grupos que é dono e apenas usuário comum. Primeiramente é realizada uma separação dos grupos que o mesmo está inserido, pois todos estão salvos na lista de conversas. A divisão facilita para trabalhar com as opções seguintes. Todos os grupos que o usuário está é listado para poder escolher as opções, se o usuário escolher um grupo que é administrador, terá à disposição mais opções, como adicionar usuários, excluir grupo e deletar o grupo. Para usuários normais, é possível apenas a listagem de usuários e sair do grupo.

Criar grupo (createGrouo()):

Esta função permite a criação de um novo grupo caso o mesmo não exista, caso já exista, ele não permite a criação. Quando um novo grupo é criado, o criador se denomina dono e ADMIN.

Chat (chat()):

Esta função permite a conversa entre os usuários ou no grupo. Toda vez que uma nova mensagem é criada, realiza-se uma concatenação com caracteres que determinam se são mensagens para uma pessoa particular ou para o grupo. G>>> determina que esta mensagem é para um grupo. C>>> determina que esta mensagem é apenas para um usuário. O chat representa a ação de clicar em determinada conversa no aplicativo, como se o usuário estivesse realmente naquela conversa, caso ele não esteja nesta opção, ele recebe como se fosse apenas notificações na tela.

Aceitar contato (acceptContact()):

Esta função permite aceitar ou recusar novos contatos, toda vez que uma nova conexão é iniciada, o usuário alvo pode aceitar, ou recusar a conexão. Quando uma conversa particular é iniciada, é realizada a

criação de um novo tópico, *X_Y_TIMESTAMP*, para que ambos possam conversar em um novo tópico criado, ambos possuem inscrição para receber as mensagens. Quando uma nova conversa em grupo é criada, o usuário alvo recebe o nome do grupo para poder inscrever-se no tópico do grupo, para receber as mensagens. A resposta de ambos os modos é enviada para o tópico de controle de quem realizou o convite para este chat.

Conexão perdida (connlost()):

Esta função é padrão do MQTT, onde ela informa que houve uma desconexão e qual foi a causa, quando houver. A partir deste momento ele tenta realizar uma nova conexão.

Recebimentos de mensagems (msgarrvd()):

Esta função contém o tratamento para cada tipo de mensagem que o usuário da aplicação venha a receber. Cada uma delas serão descritas abaixo:

- → ESTÁ ONLINE: quando receber esta mensagem vai procurar o nome do usuário na lista para poder definir o seu novo *status*. O usuário que recebeu esta mensagem vai disparar outra informando que também está *online*.
- → ESTOU ONLINE TAMBÉM: quando receber esta mensagem vai realizar a mesma busca acima, porém caso não encontrar vai adicionar este novo usuário na lista, com seu *status* de *online*.
- → ESTÁ OFFLINE: quando receber esta mensagem vai procurar o nome do usuário na lista para poder definir o seu novo *status*, como *offline*.
- → DESEJA ENVIAR MENSAGEM: quando receber esta mensagem vai verificar se o usuário não está bloqueado, se tiver não deixa executar. Quando não estiver bloqueado chama a função para adicionar o contato, caso sua resposta for sim, vai adicioná-lo na lista se não encontrá-lo e alterar o status da conexão se necessário.
- → DESEJA ADICIONÁ-LO AO GRUPO: semelhante ao funcionamento acima, porém adiciona na lista de grupos.
- → C>>>: quando receber esta mensagem indica que é conversa particular, imprime na tela com o nome de quem enviou e a mensagem.
- → G>>>: quando receber esta mensagem indica que é conversa em grupo, imprimindo o nome do grupo, o nome de quem enviou e a mensagem. Caso o usuário que mandou a mensagem não estiver na lista de usuários que compõem o grupo, é adicionado e alterado o status para online.
- → CONEXÃO RECUSADA: quando receber esta mensagem indica que não aceitou o convite de um novo chat, ou para compor o grupo, assim

é definido como *Denied* na lista de conversas. Neste caso, o usuário não foi inscrito no tópico.

- → ACEITO CONEXÃO: quando receber esta mensagem indica que o usuário aceitou o convite de um novo chat, irá adicionar na lista como *Accept*, assim, ele foi inscrito no tópico.
- → ENTROU NO GRUPO: quando receber esta mensagem vai imprimir o nome de quem entrou no grupo. Vai criar uma nova mensagem informando que também está no grupo, para que ambos possam ser adicionados em suas respectivas listas com as informações de cada usuário.
- → ENTROU NO GRUPO: quando receber esta mensagem vai adicionar o usuário em sua respectiva lista com as informações dele.
- → SAIU DO GRUPO: quando receber esta mensagem vai imprimir o nome de quem saiu do grupo. Vai remover o usuário da lista.
- → REMOVEU: quando receber esta mensagem indica que alguém foi removido do grupo. Se for o usuário que está com a aplicação aberta, irá mostrar que você foi removido do grupo pelo administrador e o nome dele. Caso você for o administrador do grupo, vai mostrar ao usuário que foi removido e o nome do grupo. Em ambos os casos é realizada a remoção da lista do usuário que foi removido.
- → ESTE GRUPO JÁ EXISTE?: quando receber esta mensagem o administrador vai responder que já existe e assim não é possível definir-se como administrador. Apenas entrar no grupo.
- → O GRUPO EXISTE!: quando receber esta mensagem o usuário que tentou criar um grupo, vai se denominar USER, não sendo possível ser administrador, pois o grupo já existia.

Bloquear usuário (setBlock()):

Esta função irá bloquear um usuário, basta informar o nome de usuário. Assim ele não pode iniciar conexões contigo e você não pode iniciar uma conversa com ele. Como você bloqueou o usuário, o tópico que você estava inscrito, você se desinscreve, assim não recebe mais mensagens do usuário.

Desbloquear usuário (setUnlock()):

Esta função irá desbloquear um usuário. Oposto da funcionalidade acima. Inscrevendo novamente você ao tópico de conversa entre ambos.

Definir online (setOnline()):

Esta função irá publicar no tópico de status informando que está online.

Definir offline (setOffline()):

Esta função irá publicar no tópico status informando que está offline.

Falha ao desconectar (onDisconnectFailure()):

Esta função é padrão do *MQTT*, onde ela informa que houve uma falha ao desconectar e qual foi a causa.

Desconectou com sucesso (onDisconnect()):

Esta função é padrão do *MQTT*, onde ela informa que desconectou com sucesso.

Inscreveu-se com sucesso (onSubcribe()):

Esta função é padrão do *MQTT*, onde ela informa que você se inscreveu com sucesso em um tópico.

Falha ao inscrever-se (onSubscribeFailure()):

Esta função é padrão do *MQTT*, onde ela informa que houve uma falha ao inscrever-se em um tópico.

Conectou (onConnect()):

Esta função é padrão do *MQTT*, onde ela informa que conectou com sucesso, inscreve o usuário nos tópicos de *status* e de controle. Chama a função de definir *online*.

Contador de tempo (timer()):

Esta função é iniciada por uma thread, para executar o timer/contador para atualizar o status dos usuários que finalizarem o programa sem desconectar. Assim é possível manter o *status online* e *offline* atualizado.

Imprimir opções do menu (printMenu()):

Esta função irá imprimir sempre que chamada as opções do menu.

Menu da aplicação (menu()):

Esta função irá ficar em execução a todo momento, até que o usuário desconecte-se ou feche a aplicação. Primeiramente ela solicita o nome de usuário que será usado durante o chat. Assim ela irá realizar a criação do cliente e irá executar o menu com todas as opções descritas acima. Dentro desta função é criado a thread para o timer acima.

Main (main()):

Esta é a função raiz do código, onde ela irá limpar todas as listas e chamar o menu com todas ações que possam ser realizadas.

Descrição dos principais aspectos da implementação

A aplicação executada é via terminal, e toda em modo texto, com opções claras e objetivas para fácil interação do usuário. Segue demonstração do menu abaixo:

```
Welcome deividi

Welcome deividi

1 - Conectar com novo usuário
2 - Listar usuários online
3 - Chats criados
4 - Bloquear usuário
5 - Desbloquear usuário
6 - Criar grupo
7 - Administrar grupos
8 - Desconectar
```

```
Welcome luis

Welcome luis

1 - Conectar com novo usuário
3 - Chats criados
5 - Desbloquear usuário
7 - Administrar grupos

2 - Listar usuários online
4 - Bloquear usuário
6 - Criar grupo
8 - Desconectar
```

A linguagem desta aplicação, conforme informada anteriormente, foi utilizada a linguagem de programação "C". Para implementações gráficas encontramos algumas limitações, desta maneira, utilizamos do modo texto para que o foco do trabalho fosse obtido.

Instruções para compilação e utilização do aplicativo

Para compilar o aplicativo, basta digitar no terminal, após estar no mesmo diretório em que o arquivo está presente.

Digite no terminal: gcc chat.c -o chat -lpaho-mqtt3as -lpthread

Para executar: ./chat

Lembrando que para a execução se tornar possível, é necessário a instalação do *PAHO MQTT* em ambiente linux.

Segue abaixo as instruções de instalação do PAHO MQTT:

- 1 sudo apt-get update
- 2 sudo apt-get upgrade
- 3 sudo apt-get install libssl-dev
- 4 sudo apt-get install doxygen
- 5 git clone https://github.com/eclipse/paho.mqtt.c.git
- 6 cd paho.mqtt.c
- 7 make
- 8 make html
- 9 sudo make install

Segue site utilizado para instalar:

https://www.embarcados.com.br/paho-mqtt-em-c-no-linux-embarcado/