

Trabalho - GEX635 - Tópicos especiais em computação XIII

Matheus Dias Negrão

1711100026

ARQUITETURA

Para o desenvolvimento do trabalho foram utilizadas as funções assíncronas da biblioteca Paho-MQTT, utilizando as funções de callback e tratamento de exceções definidas na biblioteca. Todos os usuários assim que criados tem seus tópicos de Controle e de Cliente definidos, onde eles são assinantes e recebem solicitações de sessões de chat e confirmação ou não de chats respectivamente, apenas os próprios usuários podem escutar nesses tópicos.

Para realizar a comunicação um-a-um o usuário deve inicialmente solicitar uma sessão através da opção no menu, passando o ID do destinatário para envio da solicitação. O payload de solicitação conta apenas o ID do remetente, uma vez que todas as mensagens recebidas no tópico de controle são para solicitações de conexão.

A aplicação do destinatário por sua vez recebe essa solicitação e armazena em uma estrutura de controle de chats temporários. Para aceitar a solicitação, o destinatário acessa a opção no menu e por lá pode visualizar todas as solicitações enviadas a ele. Escolhendo uma e dizendo se quer aceitar ou não.

Em caso de aceite, a aplicação irá definir o tópico da sessão de chat com os ID's correspondentes e um número aleatório. O destinatário assina o tópico e responde o remetente com o tópico da sessão e a tag de aceite no payload. Em caso de negação da solicitação, o destinatário responde ao remetente apenas com a tag de negação. Em ambos os casos a solicitação é removida da estrutura de solicitações

Após a criação da sessão, o usuário pode encontrá-la ao selecionar a opção de enviar uma mensagem em chat, podendo então, enviar a mensagem naquela sessão. Se ele não estiver em nenhuma sessão de chat, o usuário será notificado.

Para comunicação em grupo o usuário informa o grupo que deseja entrar, se tornando assinante daquele grupo, e assim como nas sessões, para enviar uma mensagem ele deve selecionar a opção e enviá-la.

Em todos os casos as mensagens são recebidas na thread msgarrvd, onde são definidas da seguinte forma:

- As mensagens que chegam no tópico controle são de solicitação de chat, contendo no payload apenas o id do usuário que solicita a sessão.

- As mensagens que chegam no tópico cliente são de confirmação ou não de sessões de chat. O payload é definido pela tag de resposta e em caso de confirmação o tópico de sessão.
- As outras mensagens de outros tópicos são as mensagens de sessões de chat e de grupos. As de grupo contém no payload o grupo que ele está publicando, o id do usuário que está publicando e a mensagem em si. Nos tópicos de chat individual são recebidos no payload o id do usuário que enviou e a mensagem propriamente dita;

As estruturas principais do MQTT que foram utilizadas na comunicação são:

MQTTAsync_sendMessage e *MQTTAsync_subscribe*, que publicam e assinam em tópicos respectivamente.

As principais estruturas criadas no trabalho são:

1. **delete_solicitation** - Função para excluir solicitações de mensagem;
2. **pub_msg** - Recebe um tópico e um payload. Cria a estrutura de mensagem do MQTT e envia no tópico recebido;
3. **sub_topic** - Recebe o nome do tópico e o assina com as funções do MQTT;
4. **temp_chat** - Recebe a solicitação de mensagem e a salva em uma estrutura de controle;
5. **accept_chat** - Em caso de aceite de uma solicitação para um nova sessão chat, cria o tópico para a sessão e o assina. Além de enviar o tópico para o solicitante;
6. **deny_chat** - Em caso de negação de uma solicitação, envia para o destinatário a negação da sessão;
7. **handle_new_chat** - Função para aceitar ou negar as solicitações de chat, buscando-os na estrutura de controle;
8. **send_msg_chat** - Recebe do usuário a mensagem e chama a função para enviá-la no tópico da sessão escolhida;
9. **send_msg_group** - Recebe do usuário a mensagem e chama a função para enviá-la no tópico do grupo escolhido.
10. **sub_group** - Pede ao usuário em qual grupo ele quer se cadastrar e chama a função para assinar o tópico correspondente;
11. **ini_chat** - Inicia o procedimento de solicitação para um novo chat, pedindo ao usuário o id do destinatário e chamando a função para publicá-la no tópico de controle do destinatário.

Para tratar eventuais erros de comunicação com o broker foram utilizadas as funções fornecidas nos exemplos da biblioteca paho, onde são definidos callbacks para cada tipo de ocorrência de erro e de sucesso. São elas as funções:

- **connlost** - tenta uma reconexão em caso de desconexão
- **onDisconnectFailure** - Caso a desconexão intencional do broker falhe;
- **onDisconnect** - Caso a desconexão intencional funcione;
- **onSendFailure** - Caso o envio de alguma mensagem falhe;
- **onSend** - Caso o envio de uma mensagem ocorra corretamente;
- **onConnect** - Caso a conexão com o broker ocorra corretamente;
- **onConnectFailure** - Caso a conexão com o broker falhe;

IMPLEMENTAÇÃO

Para a implementação foram utilizadas as funções assíncronas da biblioteca Paho-MQTT-C. Utilizando o broker mosquitto para realizar a comunicação dos tópicos do cliente MQTT.

Em grande parte da aplicação são tratadas as strings de entrada do usuário para que possa ser feita a comunicação entre os usuários, tanto um-a-um como em grupo. O limite de usuários da aplicação é de 99, enquanto as mensagens e os tópicos também são strings limitadas por valores estáticos, 64 e 30 caracteres respectivamente. O usuário pode se conectar até 100 chats únicos e 99 em grupo.

Foram utilizados também mutexes para controle de áreas de acesso por múltiplas threads, como o *sub_topic* e *pub_msg* e a estrutura de tópicos assinados, que podem ser acessadas tanto pela thread do usuário como pela thread *msgarrvd*, ou seja, pelas solicitações e confirmações recebidas. Apesar das funções da biblioteca do paho-MQTT serem thread safe, outros parâmetros são configurados além do envio e assinatura de tópicos.

UTILIZAÇÃO

Download e instalação:

Para utilizar a aplicação é necessário ter instalado o broker mosquitto e a biblioteca paho-MQTT. Abaixo são demonstradas as instruções de utilização no Ubuntu 20.04.

Mosquitto Broker

```
sudo apt update
sudo apt install mosquitto
sudo apt install mosquitto-clients
```

Paho-MQTT-C

```
git clone https://github.com/eclipse/paho.mqtt.c.git
cd paho.mqtt.c
make
sudo make install
```

Execução:

Para execução da aplicação deve se compilar o arquivo utilizando a biblioteca do paho mqtt e posteriormente executá-lo, como demonstrado abaixo:

```
gcc chat-mqtt.c -o chat -lpaho-mqtt3as -lpthread
./chat
```

Utilização:

Assim que executado, para usar o aplicativo, o usuário deve inserir o seu id único (01 - 99) com dois dígitos. Após a conexão, as funções do menu podem ser utilizadas escrevendo no terminal. A todo momento, independente da chegada de mensagens, os comandos dados no terminal serão direcionados à seleção do menu.

```
Digite seu ID único:
02
Bem-vindo! Agora você está online!

-- Escolha uma opção --
1. Iniciar um chat com um usuário
2. Aceitar uma solicitação de chat
3. Enviar mensagem em um chat
4. Entrar em um grupo
5. Enviar mensagem em um grupo
6. Desconectar do broker
0. Encerrar a aplicação
```