

# Relatório - MAC0352 - EP1

Matheus Barbosa Silva

Instituto de Matemática e Estatística

26 de Abril de 2022

# Sumário

- 1 Organização da Implementação
- 2 Interpretação de pacotes de controle

# Sumário

- 1 Organização da Implementação
- 2 Interpretação de pacotes de controle

# Organização da Implementação

A solução criada para o EP1 é baseada no código de exemplo `mac0352-servidor-exemplo-ep1+pipe.c`. Tópicos e subscribers são identificados por meio do sistema de arquivos:

- **Tópicos:** A cada novo tópico (descrito por um *subscriber*), associa-se um diretório dentro de `/tmp` **identificado pelo hash do nome do tópico**. Dessa forma, são evitados alguns problemas de segurança (já que os arquivos são mantidos todos no mesmo nível de hierarquia e um provável atacante não é capaz de percorrer a árvore de arquivos);
- **Subscribers:** Cada novo subscriber tem associado a si um pipe dentro do diretório referente ao tópico em que está inscrito. O nome do pipe de cada subscriber é dado de acordo com a ordem de chegada (partindo de zero);

Note que nenhuma estrutura de dados adicional foi mantida internamente no broker.

# Bibliotecas

Em comparação ao código de exemplo fornecido, foi acrescentado o uso da biblioteca `dirent.h`.

Essa biblioteca foi utilizada para permitir **percorrer os diretórios** e verificar quantos pipes (*subscribers*) existem associados a cada tópico. Assim, é possível atribuir um identificador único a cada novo *subscriber* (de acordo com a ordem de chegada) sem precisar manter um contador.

## Funções Auxiliares

Para melhor legibilidade e estruturação da implementação, foram criadas algumas funções auxiliares que são acionadas nos tratamentos de diversos tipos de pacotes:

- `hash`: algoritmo de hash djb2 de Dan Bernstein. Recebe um vetor de `unsigned char` e retorna um `unsigned long` associado a essa string;
- `getIntValueFromBytes`: recebe dois bytes (`a` e `b`) e retorna o valor inteiro correspondente aos bytes concatenados (`ab`). Usa uma **máscara** para obter o valor inteiro referente a cada byte;
- `getDirPathString`: recebe o nome de um tópico e retorna a string do caminho até ele;

## Funções Auxiliares

Também foi implementada a função auxiliar `sendPublishMessage`, utilizada dentro da interpretação de pacotes SUBSCRIBE:

- `sendPublishMessage`: envia um pacote PUBLISH do servidor a um dado cliente. Essa função recebe o nome do tópico do subscriber, a mensagem a ser enviada e dois bytes `topicLenb1` e `topicLenb2` que especificam a largura do conteúdo a ser enviado (mensagem + tópico). Esses bytes são lidos e interpretados concatenados, logo estão relacionados ao comprimento total da mensagem e do tópico;

# Sumário

- 1 Organização da Implementação
- 2 Interpretação de pacotes de controle



# Interpretação de pacotes de controle

Cada pacote recebido pelo broker tem um identificador de um byte que permite descobrir o seu tipo. Cada tipo de pacote recebe uma tratativa distinta:

- **PINGREQ**: a interpretação desses pacotes é essencial para evitar desconexões de clientes. O broker recebe um PINGREQ (código 12) e retorna um PINGRES (código 13). Ambas as mensagens têm 2 bytes.
- **CONNECT**: é um pacote de código 1. O servidor envia um pacote CONNACK (código 2, tem 4 bytes) se e somente se o pacote recebido tiver a escrita "MQTT" correta em seu conteúdo.

# Interpretação de pacotes de controle

- **SUBSCRIBE**: é um pacote de código 8 que contém o nome do tópico no qual o *subscriber* deseja se inscrever. A largura da string do nome do tópico é descrita no quarto e no quinto bytes recebidos (aqui usa-se a função auxiliar `getIntValueFromBytes` para obter o valor inteiro da largura). O broker retorna ao cliente uma **mensagem SUBACK** (de código 9, com 5 bytes) contendo o mesmo id enviado – o que confirma o recebimento do pedido de inscrição em um tópico. Usa-se a função auxiliar `getDirPathString()` para obter o caminho do diretório do tópico. Assim, usa-se a biblioteca `dirent` para contar quantos subscribers já existem nesse tópico e, em seguida, um pipe é criado.

# Interpretação de pacotes de controle

- **PUBLISH:** é um pacote de código 3 que tem o seguinte formato:

**Largura do Conteúdo - Largura do Tópico (2 bytes) - Tópico - Mensagem**

O broker usa a função auxiliar `getDirPathString()` para obter o caminho do diretório do tópico. Assim, usa-se a biblioteca `dirent` para acessar o nome de todos os subscribers do tópico e envia-os a mensagem do pacote.

## Recebimento da mensagem no pipe

Cada processo filho que recebe a mensagem é responsável por enviar um pacote PUBLISH (idêntico ao recebido pelo broker) ao seu subscriber correspondente. Esse processo é feito usando a função auxiliar `sendPublishMessage`.