Relatório - MAC0352 - EP1

Gustavo Santos Morais

Instituto de Matemática e Estatística

Abril de 2022

Sumário

Sumário

Implementação

Como base, foi utilizado o código disponibilizado pelo professor no e-Disciplinas (servidor de eco + pipe). Em cima desse código, foram feitas as mudanças necessárias para conseguir atender aos requisitos pedidos no enunciado.
Removi trechos de código que não utilizei na minha implementação, assim como alguns dos comentários que o código tinha, para facilitar a leitura dos trechos referentes as mudanças feitas durante o desenvolvimento do EP1.

Implementação

A parte que sofreu maior modificação foi a que vem após o uso da função *fork*, que cria um processo filho para cada cliente novo que se conectou na porta do servidor. Dentro deste trecho, eu capturo os dados enviados pelo cliente ao servidor por meio do vetor *recvline*.
Para reconhecer cada tipo de pacote *MQTT* vindo do cliente, eu verifico a primeira posição do vetor *recvline*, e após o uso de uma *bitmask* e o uso de uma operação de *shift*, eu consigo recuperar o código referente ao tipo de pacote, e assim, separar o que o servidor deve em cada caso. O código do tipo de pacote MQTT é guardado na variável *packetType*.

Ações do servidor para cada tipo de pacote

- **packetType == 8 (SUBSCRIBE)**: Primeiramente, o servidor reconhece cada um dos tópicos que o cliente deseja se inscrever. Para cada tópico reconhecido, é criada uma pasta de caminho `\tmp\<nome_do_topico>`. Cria-se um arquivo referente à este cliente em cada uma dessas pastas. O nome desse arquivo será o valor de um contador que é incrementado a cada novo cliente conectado no servidor (o que garante a unicidade dos nomes dos arquivos dentro das pastas dos tópicos). Este processo referente a este cliente será dividido em múltiplas threads, uma para cada tópico inscrito, onde cada thread será responsável por capturar o conteúdo que será escrito nesses arquivos, e enviar uma mensagem de volta para o cliente, no formato de um pacote do tipo *PUBLISH*. Também é enviado um pacote MQTT para o cliente do tipo CONNACK.

Ações do servidor para cada tipo de pacote

- **packetType == 3 (PUBLISH)**: Primeiramente, o servidor irá reconhecer o tópico que este cliente deseja fazer a publicação e a mensagem que será publicada. Após isso, abre-se o diretório `\tmp\<nome_do_topico>` e itera-se sobre todos os arquivos que estão dentro dele, que são referentes aos clientes *sub* que se inscreveram no tópico em questão. Dessa forma, o cliente escreve a mensagem em todos os arquivos dentro daquele diretório.

Ações do servidor para cada tipo de pacote

- **packetType == 1 (CONNECT)**: Nesse caso, o servidor apenas responde o cliente enviando um pacote MQTT do tipo CONNACK para validar a conexão do cliente.
- **packetType == 12 (PINGREQ)**: Nesse caso, o servidor apenas responde o cliente enviando um pacote MQTT do tipo PINGRESP.
- **packetType == 14 (DISCONNECT)** : Nesse caso, o servidor apenas verifica se o cliente que está desconectando é um cliente *sub*. Se sim, ele irá encerrar todas as threads que foram iniciadas e excluir os arquivos criados por esse cliente.

Sumário

Teste

Não consegui realizar os testes a tempo.

Sumário

Obsevações

Neste EP, estou supondo que que os nomes dos tópicos não serão tão grandes, pois para capturar o tamanho desse tópico, eu estou pulando um byte que apenas seria usado quando o tamanho do tópico fosse grande.