Relatório

Fundamentos de Sistemas Paralelos e Distribuidos

Resumo:

Nesse trabalho foi implementado um servidor e cliente que utilizam o gRPC(google RPC) para comunicação sincrona em uma rede utilizando o protocolo http2.

gRPC:

A interface implementada pode ser encontrada em protos/informer.proto, nesse arquivo foi definido o contrato do serviço Informer, esse contrato define dois métodos para comunicação RPC, GetDescription e GetPort que são responsáveis por retornar a descrição de um serviço e a Porta em que esse serviço está escutando respectivamente.

Esse contrato pode ser compilado em código c++ através do comando:

```
$ make compile-protos
```

Esse comando irá executar o compilador protoc para gerar os tipos e stubs utilizados para comunicação RPC.

Servidor

A implementação do servidor está em server/main.cc dentro da classe InformerImp, ela utiliza o código gerado pelo protoc para definir a implementação dos métodos RPC. Os métodos GetDescription e GetPort são responsáveis por abrir o arquivo que contem as informações dos clientes, procurar no conteúdo desse arquivo pelo serviço desejado, e a partir disso extrair as informações relevantes para cada método, no caso, descrição e porta respectivamente.

CLiente

O cliente é ainda mais simples que o servidor, ele usa o endereço passado para o cliente para inicializar o stub, uma classe que abstrai o stub em algo mais nativo no ambiente c++, escondendo detalhes de rede como status e o stub propriamente dito.

Formato do arquivo de serviços

Como solicitado, o arquivo de serviços segue a mesma syntaxe que a usada no arquivo /etc/services, ela foi retirada de https://www.lifewire.com/what-is-etc-services-2196940. Uma estrutura foi definida no arquivo proto para conter os campos que podem ser inseridos nesse arquivo, como nome, porta/protocolo e etc, todos os campos não definidos na linha são retornados como uma string de tamanho 0.

Em cada endpoint do servidor o arquivo é aberto e cada linha é processada de modo sequêncial.