

Trabalho Prático

Twitter

Henrique Cesar Barbosa
João Victor Teixeira
Rômulo Rafael da Silva

INTRODUÇÃO

Com o objetivo de desenvolver uma aplicação de redes que usa sockets, o seguinte trabalho implementa uma aplicação cliente-servidor para comunicação entre diferentes agentes na qual cada um pode se inscrever ou desinscrever de um canal para receber mensagens de uma tag específica. O trabalho simula a ideia de um twitter, mas com canais específicos para seus assuntos de interesse. Foi usado o UDP como protocolo de comunicação devido a sua fácil implementação e simples comunicação.

IMPLEMENTAÇÃO

O trabalho funciona com um servidor, que é responsável por receber e transmitir todas as mensagens, e vários clientes, que enviam ao servidor suas mensagens e quais as tags que desejam se inscrever (ou se desinscrever) para receber suas mensagens. O trabalho foi desenvolvido em dois arquivos diferentes, um para implementação do servidor e outro para implementação do cliente.

Servidor

O servidor abre uma conexão informando a porta e fica aguardando por mensagens. Usou-se um dicionário para armazenar as tags de interesse de cada cliente, onde a chave é o IP junto à porta do cliente e o valor de cada posição é uma lista de tags representando os interesses do cliente. Portanto, ao receber uma mensagem, o servidor deve verificar se os caracteres desta estão dentre os caracteres aceitos e, caso estejam, continuar o fluxo do programa. Caso a mensagem não possua os caracteres válidos, o programa encerrará o seu fluxo.

Após validar a validade da mensagem, o próximo passo é decodificá-la de modo a identificar as tags que ela possui. Tal decodificação foi feita com o uso de expressões regulares já que as tags não possuem espaço e estão sempre acompanhadas de um #, + ou - no seu início.

Exemplo de mensagem: *A prova de #redes foi adiada.*

TAG: *#redes*

Após a identificação das tags, avalia-se qual ação que será feita com base no símbolo que a precede. Caso seja um +, então essa tag será adicionada à lista de interesse do cliente que a enviou. Caso seja um -, então essa tag será removida da lista de interesse do cliente que a enviou. Caso seja um #, então essa mensagem será enviada para todos os clientes que possuem a tag em sua lista de interesse, menos para o cliente de onde a mensagem se originou. Caso o cliente tente adicionar uma tag já existente ou remover uma tag inexistente da sua lista, uma mensagem de erro será mostrada.

Cliente

Ao iniciar a conexão por meio do ip e da porta informada, o cliente fica no aguardo de mensagens do servidor ou de comandos digitados pelo usuário. O cliente pode solicitar adicionar ou remover tags sozinhas ou em meio às frases, como por exemplo:

Mensagem: *Quero saber sobre as matérias do +dcc e esquecer as matérias da -física.*

TAG adicionada: *dcc*

TAG removida: *física*

O cliente pode mandar mensagens com tags também, basta as colocar com um # antes da tag e enviá-la que todos os outros clientes com essa tag em sua lista de interesse irão recebê-la. Um ponto importante é que um cliente não recebe as mensagens que ele mesmo enviou.

EXECUÇÃO

Para executar o projeto, pode-se rodar o script `my_tmux.sh` que cria as instâncias do servidor e de 3 clientes com o auxílio do `tmux`, mas caso queira rodar no terminal mesmo, abaixo seguem os comandos com uma breve explicação deles.

Para inicialização do servidor e dos clientes, os comandos abaixo devem ser rodados.

- Para iniciar o servidor: `python server.py` (número da porta desejada)
- Para iniciar o cliente: `python client.py` (número da porta desejada) (endereço de IP) (porta do servidor)

Agora basta escrever as mensagens e enviá-las como nos exemplos mostrados nas seções acima.

CONCLUSÃO

O trabalho foi de grande contribuição para aprendizado e entendimento de alguns tópicos aprendidos em aula. Este nos permitiu entender um pouco melhor como funciona o UDP além de termos visto como fazer uma aplicação com o uso de sockets. O trabalho foi feito sem grandes dificuldades, já que nosso maior problema era entender como as coisas funcionavam, mas algumas pesquisas já nos esclareceram as dúvidas. Por fim, foi disponibilizado um script para testes utilizando o tmux por ser mais prático de executar e ver o funcionamento do programa e um vídeo mostrando o funcionamento do sistema se encontra junto ao seu código fonte. Com isso, conclui-se o trabalho prático da disciplina de Redes de Computadores.