REDES DE COMPUTADORES

TRABALHO PRÁTICO II - DOCUMENTAÇÃO

Wilson Moreira Tavares
2014068334

1. INTRODUÇÃO

Este trabalho prático envolve os conceitos trabalhados em sala, sobretudo o protocolo UDP e a biblioteca Socket. Foi implementado um sistema de gerenciamento de preços de combustíveis, com as informações sendo trafegadas através da rede.

Um socket de rede é o ponto-final de um fluxo de comunicação entre 2 aplicativos através de uma rede. Em cada ponta da comunicação implementada há um aplicativo, cliente e servidor, que compartilham informações. Um socket de rede em um computador é definido como a combinação de:

- um endereço IP
- o número de uma porta

2. O SISTEMA

Foi implementado um sistema de gerenciamento de preços de combustíveis. O sistema armazena dados sobre o preço de combustível e é capaz de fornecer informações sobre o posto de combustível com o menor preço dentro de um raio estipulado.

Na aplicação desenvolvida, há duas aplicações, cliente e servidor. O programa cliente é responsável por enviar mensagens que podem ser do tipo **inserção** de dados ou **pesquisa**. O programa servidor é responsável por receber as mensagens do cliente e interpretá-las, **salvando** os dados em um arquivo **ou pesquisando** o posto que apresenta menor preço de acordo com os parâmetros estabelecidos. O servidor **sempre retorna** uma mensagem de confirmação, contendo o id da mensagem e, caso seja necessário, dados adicionais de resposta.

As mensagens enviadas possuem uma estrutura definida, de forma a possibilitar a compreensão delas, seja por parte do cliente ou do servidor. As mensagens variam com relação ao tipo de parâmetros enviados entre elas, assim como o conteúdo dos parâmetros.

O programa cliente envia os seguintes tipos de mensagem:

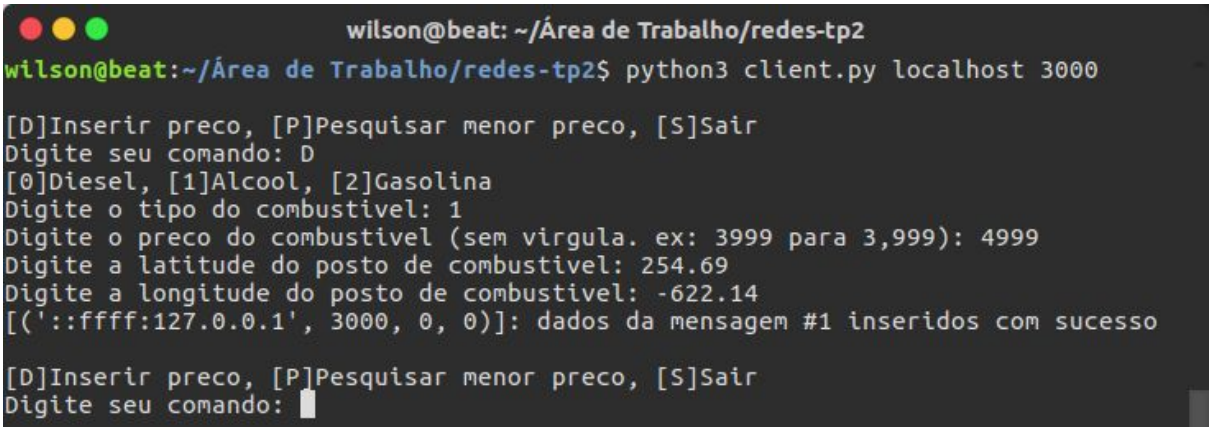
- Pesquisa: a mensagem possui a estrutura
    - P_[id]_[combustivel]_[raio de busca]_[latitude]_[longitude]
    - ex: P_2_1_100_-12.1_33
        - P: tipo de mensagem (pesquisa)
        - 2: id da mensagem (gerado automaticamente)
        - 1: tipo de combustível (álcool)
        - 100: raio de busca
        - -12.1: latitude
        - 33: longitude
- Inserção de dado
    - D_[id]_[combustível]_[preço]_[latitude]_[longitude]
    - ex: D_5_0_2999_298.1_0.12
        - D: tipo de mensagem (inserção de dado)
        - 5: id da mensagem (gerado automaticamente)
        - 0: tipo de combustível (diesel)
        - 2999: preço do combustível
        - 298.1: latitude
        - 0.12: longitude

O programa servidor envia os seguintes tipos de mensagem:

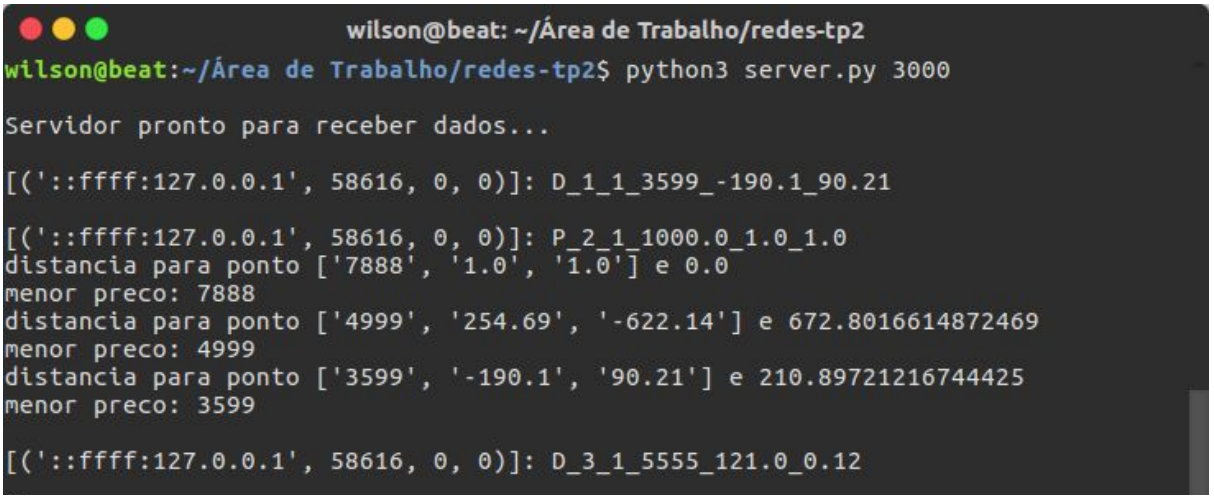
- Mensagem de confirmação de inserção de dados
- Mensagem de resposta a uma pesquisa
    - [id]_[preço]_[latitude]_[longitude]
    - [id]_N: caso nenhuma resposta for encontrada

3. IMPLEMENTAÇÃO

Como foi dito anteriormente, a função do cliente é enviar mensagens de pesquisa ou de inserção de dados, enquanto o servidor recebe tais mensagens e as interpreta, salvando os dados em um arquivo ou realiza a pesquisa e envia a resposta ao cliente. O conteúdo da mensagem do cliente é coletada através da entrada padrão, de forma separada, como mostra a imagem abaixo.

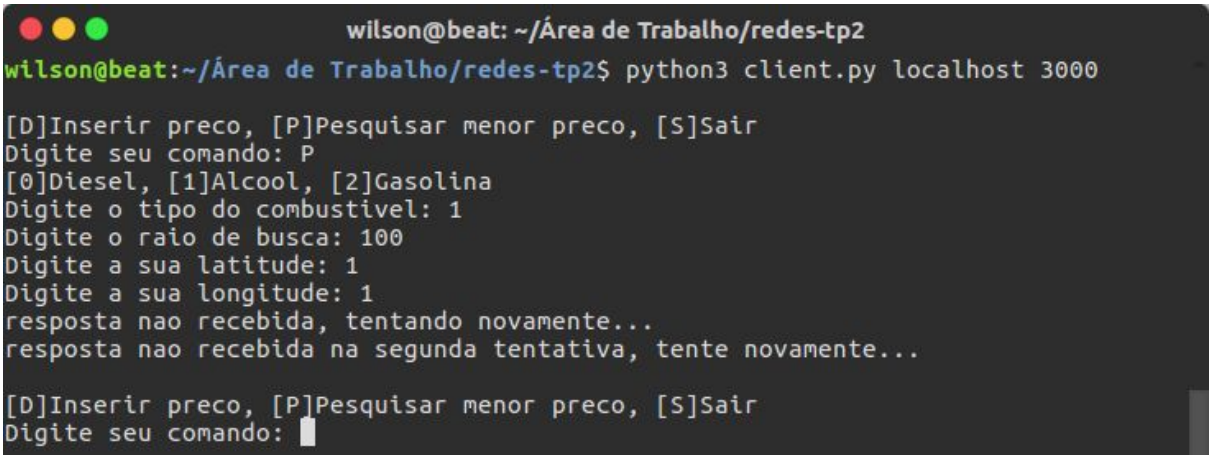


O servidor recebe a mensagem do cliente, interpreta a mesma e retorna uma resposta. Além disso, o mesmo imprime o conteúdo das mensagens recebidas pelo cliente, assim como informações relacionadas à busca de informações nos arquivos, como é possível ver abaixo.



Como foi dito anteriormente, os dados sobre os preços dos combustíveis são salvos em arquivos. Decidi fazer uso de três arquivos, um para cada tipo de combustível, sendo chamados “diesel.txt”, “alcool.txt” e “gasolina.txt”. Tal decisão foi tomada para tornar a busca mais rápida, tendo em vista que o escopo de busca é reduzido.

Como especificado, foi feito uso do protocolo UDP para transmissão dos dados. Tal protocolo não implementa confirmação do recebimento de mensagens. Por esse motivo, foi implementado manualmente um mecanismo de confirmação de recebimento de mensagens e retransmissão, caso seja necessário. Tal mecanismo funciona da seguinte forma: a mensagem é lida e enviada pelo cliente, que aguarda uma resposta do servidor por 5 segundos, caso nenhuma resposta seja enviada de volta, ocorre a retransmissão da mensagem.



A implementação seguiu os padrões definidos pela especificação. A aplicação servidor faz uso de IPv6, assim como a aplicação cliente. Caso seja passado um endereço IPv4 para a aplicação cliente, é identificado que o endereço é IPv4 e o mesmo é convertido para IPv6.

4. TESTES

Foram realizados testes com as seguintes situações:

- inserção de preço
- pesquisa por preço
- pesquisa que não possui nenhum resultado
- combustíveis inexistentes

Em todos os cenários o software se manteve robusto e soube contornar as situações descritas. Não houveram tempos de espera, exceto no contexto de não recebimento de resposta, onde há um tempo de espera definido para o reenvio de mensagem.

O software foi desenvolvido em **Python 3**.

5. CONCLUSÃO

A implementação deste trabalho prático foi de extrema importância para afixar e praticar os conhecimentos adquiridos em sala. Os maiores problemas que tive foram mais no que diz respeito à implementação do mecanismo de confirmação de mensagens.

Acredito que seja possível melhorar a função de entrada de dados, utilizando exceções de forma a tornar o software mais robusto, e talvez a implementação de janelas deslizantes, para garantir a integridade das mensagens.