

Universidade Estadual de Mato Grosso do Sul

Curso de Ciência da Computação

Disciplina de Redes de Computadores

Trabalho de Redes de Computadores

IMPLEMENTAÇÃO DE UMA APLICAÇÃO DE SERVIÇOS DE
COTAÇÃO DE PREÇOS DE COMBUSTÍVEIS

Dourados (MS), 23 de maio de 2025.

INTRODUÇÃO

O objetivo deste trabalho é criar uma aplicação cliente-servidor utilizando a biblioteca **Socket** e o protocolo **UDP**.

O trabalho consiste em desenvolver um sistema capaz de enviar o preço de vários postos de combustíveis e requisitar o preço mais barato em uma dada região. O sistema será composto de dois programas: um cliente e um servidor.

SISTEMAS

Tanto o cliente quanto o servidor receberão parâmetros de entrada do seguinte tipo:

- cliente <ip/nome> <porta>
- servidor <porta>

O programa cliente se conectará ao servidor por meio do endereço IP (ou pelo nome da máquina) e porta, ambos passados como parâmetro.

O programa servidor executará o serviço de comunicação. Esta comunicação poderá se comunicar com vários clientes, mas deverá ser feita uma por vez. A porta do servidor deverá ser passada como parâmetro.

PROTOCOLO

O programa cliente poderá enviar dois tipos de mensagens ao servidor: dados (D) e pesquisa (P). Como o UDP não garante a entrega de mensagens, o cliente deverá implementar o **RDT 2.0**, isto é, **ACKs e NAKs**.

Observação: A topologia do laboratório onde testaremos o trabalho é muito pequena. Desta forma, há uma probabilidade muito pequena de termos pacotes com erros. Para contornarmos essa situação, todo pacote que for enviado pelo cliente, terá um campo erro. Este campo será do tipo booleano. Assim, antes do pacote ser enviado, deve-se executar uma função randômica onde, se o valor for maior que meio ($> 0,5$), o pacote sairá com erro. Se o valor for menor ou igual a meio ($\leq 0,5$) o pacote sairá sem erro. O restante da implementação segue a **Máquina de Estado Finito (FSM)** da bibliografia (página 26 no slide).

As mensagens de dados começam com a letra D, seguida de um inteiro indicador da mensagem; um inteiro que indica o tipo de combustível (0 – diesel, 1 – álcool, 2 – gasolina), um inteiro com o preço (multiplicar o preço por 1000 - Exemplo: 4,449 fica 4449) e as coordenadas do posto de combustível (latitude e longitude em Dourados (MS)).

O servidor deverá confirmar a recepção da mensagem e adicionar a informação em um arquivo. Vários clientes podem enviar dados e, mesmo com o término da comunicação entre um cliente e um servidor os dados enviados devem ser salvos no arquivo para consultas de outros clientes.

As mensagens de pesquisa começam com a letra P seguida de um inteiro identificador da mensagem, um inteiro que indica o tipo de combustível, um inteiro com o raio de busca e as coordenadas do centro de busca (latitude e longitude).

O servidor deverá responder com o menor preço para aquele combustível, para postos que estejam no centro de busca mais perto do raio de busca.

O cliente e o servidor devem imprimir na tela o conteúdo das mensagens que eles receberem.

ENTREGA DO CÓDIGO

O código e a documentação devem ser entregues em um arquivo Zip. Dentro deste arquivo Zip devem conter um **readme.txt** com o nome do aluno e o comando para a execução do código e, um arquivo PDF da documentação. Inclua todos os arquivos fontes (.c, .h, makefile, **não inclua**

executáveis ou arquivos objetos), em um único diretório. Um **Makefile** deve ser fornecido para a compilação do código.

Parte desse trabalho envolve o aprendizado de como construir um makefile e utilizar a ferramenta make. Este makefile, quando executado sem parâmetros, irá gerar os dois programas: cliente e servidor, EXATAMENTE com esses nomes.

Submissões onde os programas não sigam as especificações de parâmetro e nomes, makefiles não funcionando ou arquivos necessários faltando **NÃO SERÃO CORRIGIDOS**. Programas que não compilem também não serão corrigidos.

DOCUMENTAÇÃO

O texto da documentação deve ser breve, de forma que o professor possa entender o que foi feito no código sem ter que entender linha a linha dos arquivos. Implementações modularizadas deverão mencionar quais funções são implementadas em cada módulo ou classe. A documentação deverá conter os seguintes itens:

1. O sumário do problema a ser tratado;
2. Uma descrição sucinta dos algoritmos e dos tipos abstratos de dados, das principais funções, procedimentos e das decisões de implementação;
3. Decisões de implementação que porventura estejam omissos na especificação;
4. Como foi tratada a retransmissão de mensagens;
5. Testes, mostrando que o programa está funcionando de acordo com as especificações, seguidos de sua análise;
6. Print screens mostrando o correto funcionamento do cliente e do servidor e exemplos de testes executados;
7. Conclusão e referências bibliográficas.

AVALIAÇÃO

A avaliação do trabalho será composta pela execução dos programas desenvolvidos e pela análise da documentação. Todos os alunos deverão apresentar e explicar em laboratório o trabalho.

Os seguintes itens serão avaliados:

- A qualidade do código (código bem organizado, estruturado, com comentários explicativos, variáveis com nomes intuitivos, modularidade, etc);
- Execução correta do código com entrada de testes a serem definidas no momento da avaliação. As entradas de testes irão exercitar a funcionalidade completa do código e, testar casos especiais ou de maior dificuldade de implementação, mas que devem ser tratados por um programa correto;
- A aderência ao protocolo especificado. Usarei ferramentas de captura de tráfego da rede (wireshark) e analisarei o código para verificar se a comunicação está da forma definida na documentação;
- Conteúdo da documentação que deve conter os itens mencionados anteriormente;
- Coerência e coesão da documentação (apresentação visual e organização, uso correto da língua portuguesa, qualidade textual e facilidade de compreensão);
- **CASOS DE CÓPIA NÃO SERÃO TOLERADOS;**
- Qualquer elemento que não esteja especificado neste documento, mas que tenha que ser inserido para que o protocolo funcione, deve ser descrito na documentação de forma explícita.

REFERÊNCIA

Para uso do programa Make e escrita de Makefiles:

<http://www.gnu.org/software/make/manual/make.html>

<http://www.cs.colby.edu/maxwell/courses/tutorials/maketutor/>

ENTREGA E APRESENTAÇÃO DO TRABALHO

Data: 30/07/2025

O trabalho deverá ser feito individualmente.