REDES DE COMPUTADORES: PROJETO

Alunos(as): João Levi e Monique Moreira Data: 14/07/22

Professor: Leandro Sales

INTRODUÇÃO:

Para nosso projeto da disciplina de redes de computadores 1, decidimos fazer a implementação de uma **Calculadora** por meio da comunicação por meio de socket(), como proposto pelo professor Leandro Sales. Assim a ideia do projeto veio do fato de querer criar uma implementação que não necessitasse de muitas funções ou classe para entregar uma resposta para o cliente, fosse fácil de testar todas as funcionalidades, não precisasse guardar dados por lista ou banco de dados. Além disso, por meio da calculadora poderíamos trabalhar bem a parte do protocolos, que irá reger a troca de mensagens entres os dois hospedeiros.

FORMATO DA TROCA DE MENSAGEM

Formato instrução:

Comando\tParâmetro1=Valor1\tParâmetro2=Valor2\t…ParâmetroN=ValorN\t\n
…
Comando\tParâmetro1=Valor1\tParâmetro2=Valor2\t…ParâmetroN=ValorN\t\n
Return Resultado

FUNCIONALIDADES

1. Sair(): Solicitar o fim da conexão com servidor, servidor aceita a requisição e fechando o socket e dissolvendo a thread;
   Exemplo Requisição: SAIR
2. Soma(Valor1, Valor2): Recebe dois valores Float, e retornar a soma de ambos;
   Exemplo Requisição: SOMA 2 2
3. Subtração(Valor1, Valor2): Recebe dois valores Float, e retornar a subtração entre ambos;
   Exemplo Requisição: SUB 2 2
4. Multiplicação(Valor1, Valor2): Recebe dois valores Float, e retornar a multiplicação entre ambos;
   Exemplo Requisição: MULT 2 2
5. Divisão(Valor1, Valor2): Recebe dois valores Float, e retornar a divisão entre ambos;
   Exemplo Requisição: DIV 2 2
6. Mod(Valor1, Valor2): Recebe dois valores Float, e retorna à resto da divisão entre ambos;
   Exemplo Requisição: MOD 2 2
7. Exponencial(Valor1, Valor2): Recebe dois valores Float(base e expoente respectivamente), e retornar a base elevado a expoente entre;
   Exemplo Requisição: EXP 2 2
8. Eh_Par(Valor1, 0): Recebe um valores Float, e retornar string True ou False;
   Exemplo Requisição: PAR 2 0

9. Eh_Ímpar(Valor1, 0): Recebe um valores Float, e retornar string True ou False; Exemplo Requisição: IMP 2 0
10. Logaritmo(Valor1, 0): Recebe um valores Float, e retornar logaritmo na base 2; Exemplo Requisição: LOG 2 0
11. Raiz_Quadrada(Valor1, 0): Recebe um valores Float, e retornar a raiz quadrada; Exemplo Requisição: SQRT 2 0

- Por padrão as requisições que ***8 a 11*** deve ter dois parâmetros como as demais requisições, valor a ser calculado e 0 respectivamente para funcionar corretamente;

PROTOCOLO DE TRANSPORTE

Como estamos lidando com números e qualquer alteração irá afetar a conta, optamos pelo protocolo ***TCP*** para não perder nem um dado ou distorcer o resultados da operação.

MODO DE EXECUÇÃO

1. Baixe o repositório git ***ou*** baixe o arquivo zip:
   Repositório Git:
   a. * Inicie o git: ***git init***
   b. * Baixe o repositório git:
   git clone https://github.com/joaolevi/redes_2021.2.git
   Arquivo Zip:
   https://github.com/joaolevi/redes_2021.2/archive/refs/heads/main.zip

2. Abra o terminal na pasta **C:\...\redes_2021.2\src,** de acordo com caminho que você salvou o repositório.
3. ** Execute no terminal o comando: **python *server.py***
4. Em um novo terminal abra a pasta **C:\...\redes_2021.2\src,** de acordo com caminho que você salvou o repositório.
5. ** Execute no terminal o comando: ***python client.py***
6. Como ***client.py*** executando, siga as etapas, escolha a operação, depois escolha os números para executar a operação.
7. Espere o servidor enviar a resposta
8. Repita os passos ***4 - 7*** para ver a função **Multi Threads**, ou executar novo comando.

* É necessário ter o ***Git*** instalado corretamente na maquina, siga as instruções **https://git-scm.com/**

** É necessário o ***python 3.10.4*** e o ***path*** instalados corretamente na máquina, siga as instruções **https://www.python.org/**

DIFICULDADES

Nosso principal desafio foi conseguir fazer a troca de dados entre os sistemas finais, pois o método ***send()***, só suporta ***Str bytes***, fazendo com que toda vez que o servidor recebe uma requisição e enviar uma resposta ele tem que fazer a transformação de ***bytes*** para ***string*** para ***float*** e vice-versa.Este mesmo problema também impediu a troca de dados sem ser string, onde para enviar outros tipos de dados como array, objetos, entre outros seria necessário usar Json, por meio de um biblioteca, tanto no sistema do hospedeiro quanto no servidor.