

UNIVERSIDADE FEDERAL DE MINAS GERAIS

CALC. SERVER

Redes de Computadores

Guilherme Henrique Ferreira Sales

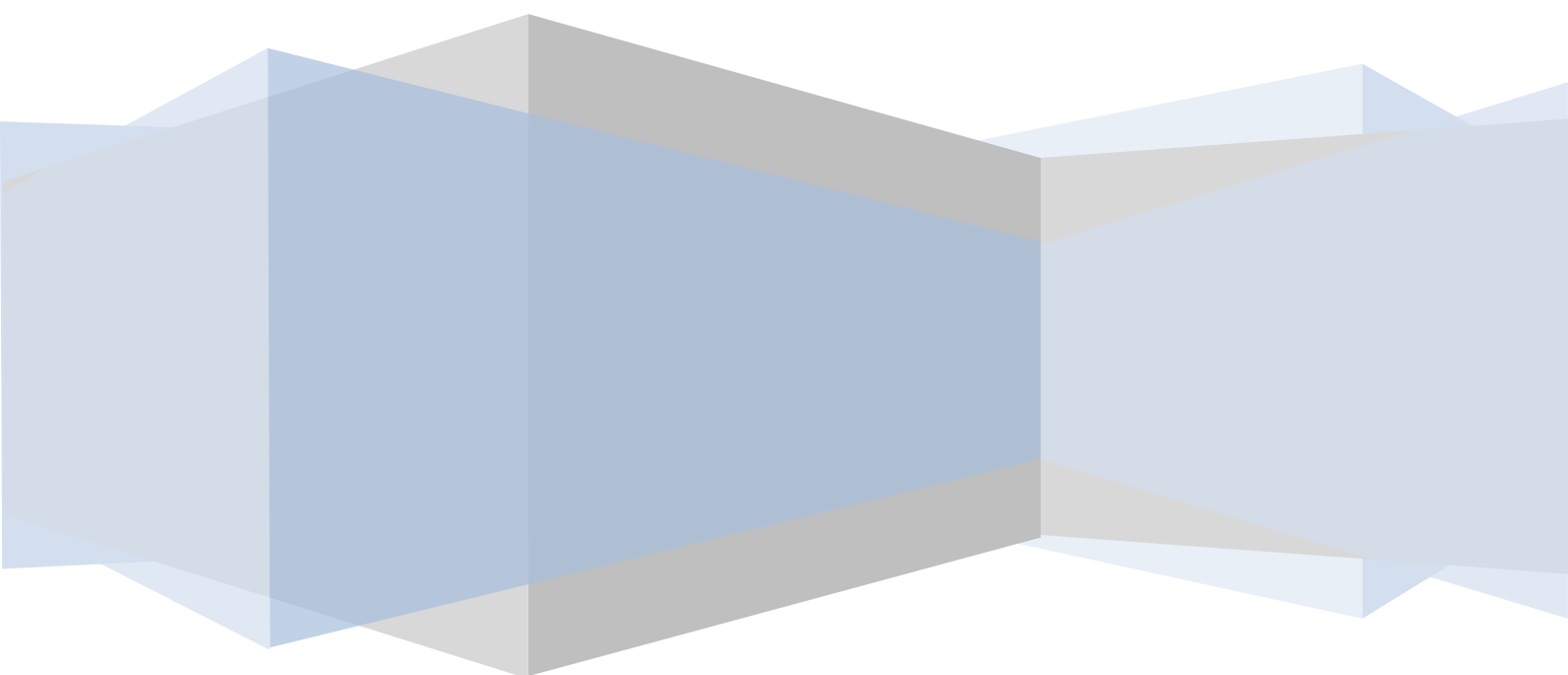
2011017984

Kamila Lissoni Aguiar

2013024325

Samara Fernanda Silva

2012018607



Introdução

Sumário do Problema

O objetivo desse trabalho é implementar uma calculadora com as seguintes operações: soma, subtração, multiplicação, divisão, resto, exponencial, raiz quadrada e fatorial. A mesma funciona, com IPv4, via rede com utilização de datagramas do tipo UDP e os dados trocados entre cliente e servidor são enviados no formato ASCII.

Dois programas foram desenvolvidos: cliente e servidor. O primeiro, no ponto de vista do usuário, funciona como uma calculadora comum, em que, após um menu de operações ser disponibilizado, ele entra com os números e operação desejados, seguindo as instruções indicadas. No entanto, o servidor é o responsável por processar as operações requisitadas pelos clientes e retornar o resultado. Além disso, o servidor possui uma função para fornecer um relatório de quantas operações de cada tipo já foram efetuadas.

Ambos vão receber dois parâmetros de entrada:

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

O programa cliente vai se conectar ao servidor através do endereço IP e porta passados por parâmetro. Após isso o servidor irá executar o serviço que tratara varias solicitações e só vai sair quando o usuário apertar Ctrl C.

Procedimentos

Algoritmo

O algoritmo utilizado para implementar a calculadora está descrito abaixo:

- 1- O cliente conecta com o servidor através do número da porta e do IP, que são passados como parâmetros. O cliente também pode requisitar um relatório de quantas operações de cada tipo foram efetuadas pelo servidor.
- 2- O usuário entra com os dados: operação desejada e valores a serem calculados.
- 3- O servidor recebe as informações desejadas pelo cliente, realiza a operação e retorna o resultado para o mesmo. Caso requisitado pelo cliente, o servidor fornece um relatório de quantas operações de cada tipo foram efetuadas.

Principais Funções

Função die

A função die é utilizada para imprimir o último erro ocorrido no formato padrão do sistema concatenado a uma mensagem passada como parâmetro e, em seguida, encerra a execução.

Função fatorial

A função fatorial é utilizada para calcular o fatorial de um número. Recebe um inteiro e retorna o valor do fatorial, que também é um inteiro. Esse valor é calculado dentro de um FOR, onde ocorre a multiplicação do número enviado pelo usuário (n) vezes uma variável i, que é o número enviado menos um. Essa multiplicação é feita até o valor da variável i for igual a um, pois deste modo terá ocorrido o produto de todos os inteiros consecutivos até o inteiro n.

Função principal cliente

A função principal cliente é utilizada para o cliente conectar com o servidor, através do número da porta e do IP que são passados como parâmetros. A espera dos dados do usuário ocorre dentro do WHILE. O usuário entra com os dados, operação desejada e os números para efetuar o cálculo. Isso ocorre dentro do SWITCH CASE, as operações da calculadora estão ligadas a cada case. Esses dados são enviados para o servidor através da chamada SENDTO - função utilizada para transmissão de dados. Após receber o resultado, enviado pelo servidor, imprime o valor na tela. O cliente também pode requisitar para o servidor, o número de operações de cada tipo que foram realizadas no total.

Função principal servidor

A função principal servidor é utilizada para criação do socket UDP. Para inicializar o socket, é preciso preencher a estrutura de endereços do servidor com a família do endereço (AF_INET), o endereço IP do servidor e a porta na qual o servidor receberá pacotes UDP.

```
si_me.sin_family = AF_INET;
```

```
si_me.sin_port = htons(PORT);
```

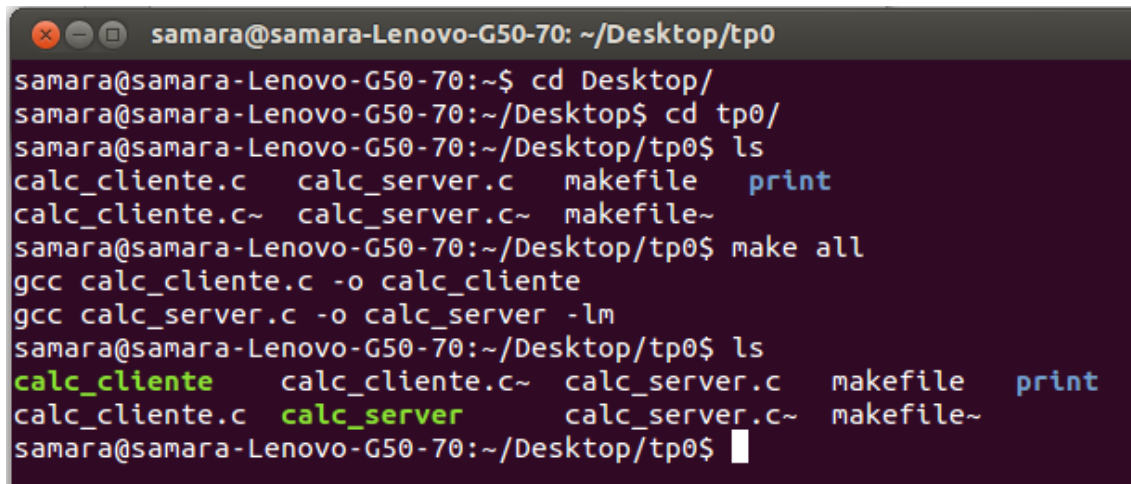
```
si_me.sin_addr.s_addr =htonl(INADDR_ANY);
```

Depois que a estrutura de endereços sockaddr_in está preenchida, basta chamar a função socket para criar o socket e a função bind para associar a estrutura de endereços com o socket criado. Isso é feito através do IF.

O servidor fica a espera de dados, isso ocorre dentro do WHILE. Caso receba esses dados, avisa ao cliente mandando um ack através da chamada SENDTO. Verifica a operação recebida através do comando SWITCH e recebe os demais números referente a cada CASE. O servidor processa a operação requisitada pelo cliente e retorna o resultado para o mesmo. As operações são realizadas dentro das cases, exceto a operação fatorial, que possui uma função para ser executada. Se o cliente solicitar o número de operações de cada tipo que foram efetuadas no total, o servidor envia um relatório contendo as informações desejadas. O envio desse relatório e do resultado obtido para o cliente através da função SENDTO.

Testes

Teste 1 (Makefile)



```
samara@samara-Lenovo-G50-70: ~/Desktop/tp0
samara@samara-Lenovo-G50-70:~$ cd Desktop/
samara@samara-Lenovo-G50-70:~/Desktop$ cd tp0/
samara@samara-Lenovo-G50-70:~/Desktop/tp0$ ls
calc_cliente.c  calc_server.c  makefile  print
calc_cliente.c~ calc_server.c~  makefile~
samara@samara-Lenovo-G50-70:~/Desktop/tp0$ make all
gcc calc_cliente.c -o calc_cliente
gcc calc_server.c -o calc_server -lm
samara@samara-Lenovo-G50-70:~/Desktop/tp0$ ls
calc_cliente  calc_cliente.c~  calc_server.c  makefile  print
calc_cliente.c  calc_server  calc_server.c~  makefile~
samara@samara-Lenovo-G50-70:~/Desktop/tp0$
```

Figura 1: Teste do Makefile

A figura 1 mostra o funcionamento do arquivo Makefile. Como descrito na especificação do trabalho:

" 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 [3]. Este makefile, quando executado sem parâmetros, irá gerar os dois programas, calc_cliente e calc_server, EXATAMENTE com esses nomes."

No diretório do arquivo (pasta tp0) digitamos o comando make all e, assim, o executável é gerado automaticamente.

Teste 2 (Operações e sequência de instruções)

```
samara@samara-Lenovo-G50-70: ~/Desktop/tp0
samara@samara-Lenovo-G50-70:~$ cd Desktop/
samara@samara-Lenovo-G50-70:~/Desktop$ cd tp0/
samara@samara-Lenovo-G50-70:~/Desktop/tp0$ make runc
./cliente 127.0.0.1 8888
SELECIONE A OPERACAO DESEJADA:
Soma: +
Subtracao: -
Multiplicacao: *
Divisao: :
Resto: r
Exponencial: e
Raiz Quadrada: s
Fatorial: !
Relatorio de operacoes efetuadas pelo servidor: h
Sair: 0

+
Digite os dois numeros desejados para a operacao selecionada:3.1
-2.4

3.10 + -2.40 = 0.70
```

Figura 2.1: Teste da operação "soma"

```
samara@samara-Lenovo-G50-70: ~/Desktop/tp0
./cliente 127.0.0.1 8888
SELECIONE A OPERACAO DESEJADA:
Soma: +
Subtracao: -
Multiplicacao: *
Divisao: :
Resto: r
Exponencial: e
Raiz Quadrada: s
Fatorial: !
Relatorio de operacoes efetuadas pelo servidor: h
Sair: 0

-
Digite os dois numeros desejados para a operacao selecionada:5.3
2.1
```

Figura 2.2: Teste da operação "subtração"

```
samara@samara-Lenovo-G50-70: ~/Desktop/tp0
samara@samara-Lenovo-G50-70:~$ cd Desktop/
samara@samara-Lenovo-G50-70:~/Desktop$ cd tp0/
samara@samara-Lenovo-G50-70:~/Desktop/tp0$ make runc
./cliente 127.0.0.1 8888
SELECIONE A OPERACAO DESEJADA:
Soma: +
Subtracao: -
Multiplicacao: *
Divisao: :
Resto: r
Exponencial: e
Raiz Quadrada: s
Fatorial: !
Relatorio de operacoes efetuadas pelo servidor: h
Sair: 0

*
Digite os dois numeros desejados para a operacao selecionada:4
5

4.00 * 5.00 = 20.00
```

Figura 2.3: Teste da operação "multiplicação"

```

samara@samara-Lenovo-G50-70: ~/Desktop/tp0
samara@samara-Lenovo-G50-70:~$ cd Desktop/
samara@samara-Lenovo-G50-70:~/Desktop$ cd tp0/
samara@samara-Lenovo-G50-70:~/Desktop/tp0$ make runc
./calc_cliente 127.0.0.1 8888
SELECIONE A OPERACAO DESEJADA:
Soma: +
Subtracao: -
Multiplicacao: *
Divisao: :
Resto: r
Exponencial: e
Raiz Quadrada: s
Fatorial: !
Relatorio de operacoes efetuadas pelo servidor: h
Sair: 0

:
Digite os dois numeros desejados para a operacao selecionada:5
2
5.00 / 2.00 = 2.50

```

Figura 2.4: Teste da operação "divisão"

```

samara@samara-Lenovo-G50-70: ~/Desktop/tp0
SELECIONE A OPERACAO DESEJADA:
Soma: +
Subtracao: -
Multiplicacao: *
Divisao: :
Resto: r
Exponencial: e
Raiz Quadrada: s
Fatorial: !
Relatorio de operacoes efetuadas pelo servidor: h
Sair: 0

r
Digite dois numeros inteiros para se obter o resto da divisao: 5
3
Resto de 5/3 = 2

```

Figura 2.5: Teste da operação "resto"

```

samara@samara-Lenovo-G50-70: ~/Desktop/tp0
SELECIONE A OPERACAO DESEJADA:
Soma: +
Subtracao: -
Multiplicacao: *
Divisao: :
Resto: r
Exponencial: e
Raiz Quadrada: s
Fatorial: !
Relatorio de operacoes efetuadas pelo servidor: h
Sair: 0

e
Digite os dois numeros desejados para a operacao selecionada:4
3
4.00 ^ 3.00 = 64.00

```

Figura 2.6: Teste da operação "exponencial"

```
samara@samara-Lenovo-G50-70: ~/Desktop/tp0
SELECIONE A OPERACAO DESEJADA:
Soma: +
Subtracao: -
Multiplicacao: *
Divisao: :
Resto: r
Exponencial: e
Raiz Quadrada: s
Fatorial: !
Relatorio de operacoes efetuadas pelo servidor: h
Sair: 0

s
Digite um numero para se obter sua raiz quadrada: 64
raiz quadrada de 64.00 = 8.00
```

Figura 2.7: Teste da operação "raiz quadrada"

```
SELECIONE A OPERACAO DESEJADA:
Soma: +
Subtracao: -
Multiplicacao: *
Divisao: :
Resto: r
Exponencial: e
Raiz Quadrada: s
Fatorial: !
Relatorio de operacoes efetuadas pelo servidor: h
Sair: 0

!
Digite um numero inteiro para se obter o fatorial do mesmo:5
5! = 120
```

Figura 2.8: Teste da operação "fatorial"

As figuras acima mostram o funcionamento de cada operação disponível para o usuário, atendendo à especificação:

" A aplicação a ser desenvolvida é uma calculadora via rede que deve oferecer as seguintes operações:

- soma, subtração, multiplicação, divisão, resto, exponencial, raiz quadrada e fatorial."

Da mesma forma, as figuras também explicitam a sequência de operações a serem seguidas pelo cliente. Elas também mostram cada passo a ser seguido pelo usuário: primeiro ele deve selecionar o tipo de operação desejada, de acordo com os caracteres indicados no "menu de operações" e, logo em seguida, é solicitado que ele digite os números desejados (um número no caso de fatorial ou raiz quadrada e dois números para as demais operações). Atendendo ao requisito do trabalho:

" Deve-se criar uma uma sequência de instruções que facilite a utilização do programa".

Teste 3 (Relatório de operações gerado pelo servidor)

```

samara@samara-Lenovo-G50-70: ~/Desktop/tp0
SELECIONE A OPERACAO DESEJADA:
Soma: +
Subtracao: -
Multiplicacao: *
Divisao: :
Resto: r
Exponencial: e
Raiz Quadrada: s
Fatorial: !
Relatorio de operacoes efetuadas pelo servidor: h
Sair: 0

h

soma: 1
subtracao: 1
multiplicacao: 1
divisao: 0
resto: 1
exponencial: 1
raiz: 1
fatorial: 0

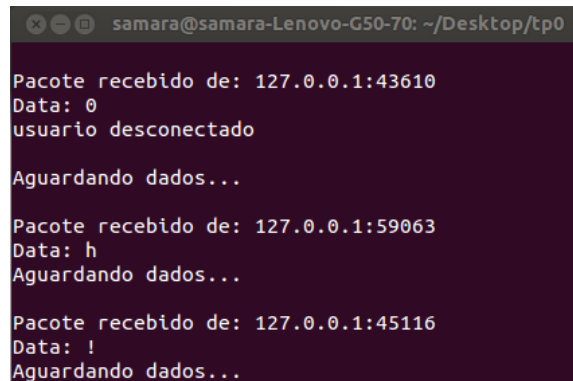
```

Figura 3: Teste da opção "Gerar relatório das operações efetuadas pelo servidor"

A figura 3 exemplifica o relatório gerado pelo servidor, que mostra quantas operações de cada tipo já foram realizadas pelo mesmo. Esta opção fica disponível para acesso de qualquer cliente conectado ao servidor, assim como as operações da calculadora. Atendendo à especificação:

" Além disso, o servidor deverá ter uma opção para fornecer um relatório de quantas operações de cada tipo foram efetuadas no total. O relatório deverá ser simples contendo em cada linha o tipo de operação e a quantidade total de operações daquele tipo já realizadas. Note que não é preciso contar a quantidade de operações por clientes."

Teste 4 (Requisições simultâneas)



```

samara@samara-Lenovo-G50-70: ~/Desktop/tp0
Pacote recebido de: 127.0.0.1:43610
Data: 0
usuario desconectado

Aguardando dados...

Pacote recebido de: 127.0.0.1:59063
Data: h
Aguardando dados...

Pacote recebido de: 127.0.0.1:45116
Data: !
Aguardando dados...
  
```

Figura 4: Teste de requisições simultâneas

Como os programas se comunicam através de redes de computadores com a utilização de datagramas do tipo UDP eles não estabelecem uma conexão. Portanto o servidor pode receber requisições de vários cliente simultaneamente. A figura 4 mostra que vários clientes estão mandando requisições ao mesmo tempo.

Conclusão

Tendo em vista os resultados observados, o Trabalho Prático 0 conseguiu alcançar o seu objetivo e foi possível simular a calculadora fazendo a conexão necessária entre o cliente e o servidor. Na resolução do trabalho foi cumprido todos os problemas enunciados e foi utilizado datagramas UDP para a comunicação em redes.

O código do programa encontra-se todo comentado, para facilitar o entendimento e possível correção de erros. No geral o TP0 não teve grandes problemas e foi de grande ajuda para relembrar vários conceitos de programação em C e que serão necessários em futuros projetos.

Referências Bibliográficas

- <http://haxent.com.br/people/ruda/make.html>
- <http://www.dcc.ufla.br/~giacomini/Textos/tutorialc.pdf>
- Computer Networks - A Tanenbaum - 4th edition