

Universidade de Brasília - Faculdade do Gama
Fundamento de Redes de Computadores - Turma: A
Professor: Fernando William Cruz
Aluno: Marcos Vinícius Rodrigues da Conceição
Matrícula: 170150747
Data: 26 de Julho de 2021

Lab01 – Construindo aplicações distribuídas usando RPC

Introdução:

Esta atividade consistiu em aprender os conceitos sobre o RPC (Chamada de procedimento remoto) e aplicá-los de maneira prática com um problema simples de construção de uma calculadora.

Descrição da solução:

A solução consistiu em entender o funcionamento da comunicação entre cliente e servidor através das chamadas de procedimento remoto.

O primeiro passo foi a criação de um arquivo **calcula.x** onde são passados os parâmetros para que o rpcgen possa criar todos os arquivos necessários para a transmissão correta dos dados entre cliente e o servidor.

```
lab01 > ≡ calcula.x
1  struct operandos {
2      int x;
3      int y;
4      int z;
5  };
6  program PROG {
7      version VERSAO {
8          int ADD(operandos) = 1;
9          int SUB(operandos) = 2;
10         int MUL(operandos) = 3;
11         double DIVI(operandos) = 4;
12     } = 100;
13 } = 55555555;
```

figura 01 - calcula.x

Uma vez que toda a lista de arquivos foi criada, bastou alterar os dois arquivos de cliente e servidor. Adicionando função main no client e as funções que calculam no servidor.

O cliente recebe 3 argumentos passados pelo usuário, os dois primeiros N1 e N2 são os números os quais deseja operar e o N3 é o seu operador.

N1 - número inteiro

N2 - número inteiro

N3 - 1 para adição, 2 para subtração, 3 para multiplicação e 4 para divisão.

No arquivo **calcula_client.c** foi criado um switch case para selecionar qual operação a ser realizada no servidor.

```
switch(z){
    case 1:
        printf("%d + %d = %d\n",x,y, add(clnt,x,y));
        break;
    case 2:
        printf("%d - %d = %d\n",x,y, sub(clnt,x,y));
        break;
    case 3:
        printf("%d * %d = %d\n",x,y, mul(clnt,x,y));
        break;
    case 4:
        printf("%d / %d = %lf\n",x,y, divi(clnt,x,y));
        break;
}
```

No programa **calcula_server.c** ficam todas as funções que são responsáveis por realizar o cálculo das operações. Na imagem abaixo um exemplo de uma das funções.

```
// Função de adição realizada pelo Servidor

int * add_100_svc(operandos *argp, struct svc_req *rqstp)
{
    static int result;

    printf("Requisicao de adicao para %d e %d\n", argp->x, argp->y);
    result = argp->x + argp->y;
    return (&result);
}
```

screenshot do cliente -> onde são passados 3 argumentos onde os dois primeiros são números inteiros e o terceiro é um número entre 1 e 4 que indica a operação a ser realizada.

```
~/A/U/1.2021/S/lab01: ~P:main:73$ ./calcula_client localhost 5 6 3
Recebe 3 argumentos os dois primeiros são numeros e o terceiro é a operação 1 para adicao, 2 para subtracao, 3 para multiplicao, 4 para divisao.
5 * 6 = 30
```

screenshot do servidor -> onde são passados 3 argumentos onde os dois primeiros são números inteiros e o terceiro é um número entre 1 e 4 que indica a operação a ser realizada.

A terminal window with a dark background. The prompt is '~/\$' followed by the path '/A/U/1.2021/S/lab01'. The command './calcula_server' has been executed. The output is 'Requisicao de multiplicacao para 5 e 6'.

```
~/A/U/1.2021/S/lab01$ ./calcula_server
Requisicao de multiplicacao para 5 e 6
```

Foi uma ótima atividade de introdução, já que aborda muito bem o funcionamento do rpc. Seguindo o material e as aulas não foram encontradas dificuldades para a conclusão do exercício.

Limitações :

- recebe apenas argumentos inteiros e quando o terceiro for intervalo é preciso re-executar o programa.
- Executa apenas uma operação por vez.
- Uma vez que a operação é realizada é necessário re-executar o programa do cliente.