

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: 12 de Julho de 2021

Lab01 – Construindo aplicações distribuídas usando sockets

Introdução:

Esta atividade consistiu em aprender os conceitos sobre o Sockets utilizando os protocolos TCP e UDP 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 dos dois tipos de protocolo, cada um com suas particularidades.

O primeiro passo foi a criação de dois arquivos um para o cliente **client-tcp.c** e **cliente-udp.c** e outro para o **servidor-tcp.c** e **servidor-udp.c**.

TCP, sigla para Transmission Control Protocol (“Protocolo de Controle de Transmissão”), é um protocolo base da internet, complementado pelo Internet Protocol (IP), sendo um tipo bastante versátil e robusto de protocolo, fazendo com que ele seja adequado para grandes redes, como a rede mundial de computadores (World Wide Web).

Basicamente, a função principal do TCP é verificar se os dados que circulam entre os dispositivos de uma rede são enviados de forma correta e na sequência apropriada.

No caso do experimento, uma vez que o servidor é iniciado ele fica esperando por alguma conexão do cliente. Quando uma conexão é realizada ele começa a esperar pelos dados do cliente e confirma para o mesmo que os dados foram recebidos com sucesso.

Já no caso do UDP (User Datagram Protocol), é um tipo de protocolo mais simples quando comparado com o TCP. Ele permite que a aplicação envie um datagrama em um pacote IPv4 ou um IPv6 para determinado destino, sem garantias de que o pacote de dados chegue ao destino da forma correta.

Esse tipo de protocolo, diferente do TCP, não é confiável e não oferece o nível de proteção e verificação dos dados transmitidos entre os usuários e dispositivos de uma determinada rede.

Diferente do TCP, o UDP é um registro indivisível, voltado à transmissão de bytes sem um começo e sem um fim determinado. O UDP é utilizado quando a velocidade da transmissão de dados é priorizada em detrimento da segurança desses dados.

No experimento Adicionando função main no cliente e as funções que calculam no servidor isso para os dois protocolos, também nos dois protocolos são necessários a criação dos sockets para que ocorra a transmissão correta dos dados.

O cliente recebe 3 argumentos passados pelo usuário, operando e mais dois números N2 e N3.

N1 - 1 para adição, 2 para subtração, 3 para multiplicação e 4 para divisão.

N1 - número inteiro

N2 - número inteiro

No arquivo **servidor-udp.c** e **servidor-tcp** foram criado um switch case para selecionar qual operação a ser realizada no servidor.

```
switch (operador)
{
case '+':
    resultado = op1 + op2;
    printf("O Resultado eh: %d + %d = %d\n", op1, op2, resultado);
    break;
case '-':
    resultado = op1 - op2;
    printf("O Resultado eh: %d - %d = %d\n", op1, op2, resultado);
    break;
case '*':
    resultado = op1 * op2;
    printf("O Resultado eh: %d * %d = %d\n", op1, op2, resultado);
    break;
case '/':
    resultado = op1 / op2;
    printf("O Resultado eh: %d / %d = %d\n", op1, op2, resultado);
    break;
default:
    printf("ERRO: Operacao nao descrita.");
}
```

screenshot do cliente-udp -> onde são passados 3 argumentos onde o primeiro é um operador e os dois seguintes são números inteiros.

```
~/De/U/1.2021/S/lab02 main ?4 ./client-udp

Entre com uma operação:
+:Adicao
-: Subtracao
*:Multiplicacao
/: Divisao
+
Entre com os dois operandos:
8
9
resultado da operacao recebido do servidor =17
```

screenshot do servidor-udp -> onde são recebidos os 3 argumentos do cliente e realiza o cálculo..

```
~/De/U/1.2021/S/lab02 main ?4 ./server-udp

0 Resultado eh: 8 + 9 = 17
```

screenshot do cliente-tcp -> onde são passados 3 argumentos onde o primeiro é um operador e os dois seguintes são números inteiros.

```
~/De/U/1.2021/S/lab02 main ?4 ./client-tcp

Socket criado com sucesso!
A conexão com o servidor ocorreu com sucesso!
Entre com uma operação:
+:Adicao
-: Subtracao
*:Multiplicacao
/: Divisao
-
Entre com os dois operandos:
8
5
Resultado da operação recebida do servidor = 3
```

screenshot do servidor-tcp -> onde são recebidos os 3 argumentos do cliente e realiza o cálculo.

```
~/De/U/1.2021/S/lab02 main ?4 ./server-tcp  
Socket criando com sucesso  
bind executado com sucesso  
Servidor esperando por conexoes  
Conexão estabelecida com o cliente.  
0 Resultado eh: 8 - 5 = 3
```

Foi uma ótima atividade de introdução, já que aborda muito bem o funcionamento dos dois protocolos. 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.