

UNIVERSIDADE DE BRASÍLIA

Faculdade do Gama

Laboratório 01

Programação para Sistemas Paralelos e Distribuídos

• ***Autores:*** Ítalo Vinícius Pereira Guimarães - 18/0102656

João Victor Valadão de Brito - 18/0103431

Docente: Fernando Willian Cruz

Brasília, DF

2022

1. Introdução

O experimento tem como objetivo explicar as principais características para o desenvolvimento de aplicações distribuídas para demonstrar o processo de passagens de parâmetros, relativos entre os conceitos comparando o processo RCP com uma aplicação desenvolvida com Sockets TCP/UDP, que foram ministradas no semestre anterior durante a disciplina de “Fundamentos de Redes e Computadores”.

Desse modo, foi proposto aos alunos este laboratório com o intuito de fixar o conteúdo a respeito de RCP por meio do desenvolvimento de uma aplicação de buscas sequencias em um vetor com tamanho fixado, para retornar o maior e o menor elemento desse vetor.

2. Descrição

Para o desenvolvimento do código, os integrantes se reuniram pela plataforma “Discord” e utilizaram o “Live Share” do VScode para desenvolver a ideia proposta do laboratório utilizando os Sockets TCP/UDP, utilizando os conceitos e materiais abordados semestre passado. Partindo para o RCP, onde deve ser instalado preferencialmente no sistema operacional Ubuntu, definimos inicialmente o tamanho do vetor e a versão da aplicação. Então, utilizamos o RCPGen para criar os arquivos base que utilizamos, então criamos 2 funções no arquivo server, que ficará responsável por exibir as chamadas de um usuário. Assim, quando um novo cliente é adicionado um novo vetor é gerado e serão verificadas se as chamadas para o servidor foram bem sucedidas.

3. 1ª Entrega

Durante essa etapa, o foco do laboratório é incentivar seus alunos a desenvolverem a aplicação utilizando apenas um host para requisitar as chamadas do cliente ao servidor. Sendo assim, para o funcionamento do Socket, primeiro o usuário navega até a página de destino (TCP) onde deve ser compilado o arquivo do servidor através do comando “gcc server.c -o server” e então executamos o arquivo nesse terminal “./server”. Então devem ser instanciados os clientes desejados através do mesmo comando “gcc client.c -o client”, porém em um segundo terminal, e após isso devem ser executados “./client”.

Partindo para o segundo foco desta primeira entrega, para utilizar a aplicação RCP, o usuário deve ter instalado em seu dispositivo o RCPGen, que pode ser instalado através do comando “sudo apt-get rpcbind” no sistema operacional Ubuntu e em seguida “sudo rpcbind” para inicializar o portmap. Então o usuário deve executar o servidor com o comando “sudo ./lab_server” e em seguida criar um novo terminal para adicionar a quantidade de clientes desejado com o comando “sudo ./lab_client”.

4. 2ª Entrega

Não conseguimos desenvolver o código da segunda entrega, porém encontramos alguns tutoriais que auxiliavam no seu desenvolvimento, assim tentamos executar o “rpcgen -m” para criar os arquivos de uma maneira segura em multi-thread, porém não conseguimos

fazer as modificações cabíveis para executar a tarefa solicitada de trabalhar com 2 workers, otimizando o processamento de maneira concorrente.

5. FeedBack

Durante o desenvolvimento do laboratório encontramos várias dificuldades, principalmente durante a implementação do RCP, onde encontramos alguns erros de compatibilidade ao utilizá-lo em conjunto com o WSL do VScode, uma vez que o terminal não conseguia localizar o RCPbind. porém ao instalar o portmap no WSL, automaticamente ele já instalava o RPCbind.

Outro problema foi enviar todo o vetor para o servidor utilizando o socket, a solução mais simples que pensamos foi enviar um valor por vez e quando o vetor estivesse completo, aí sim o servidor faria a verificação no menor e maior.

Outro desafio encontrado foi acerca da passagem de parâmetros no formato de um array para definirmos as constraints que foram utilizadas na aplicação. Provavelmente, esse era o intuito de aprendizado desse laboratório e foi onde gastamos a maior parte do nosso tempo para “realizar”, onde consultamos diversos fóruns online a respeito do RCP, documentações a respeito do Data Type e repositórios do Github para descobrirmos como usar e manipular esses arrays pelo código.

Depois da implementação em RPC, tivemos o mesmo problema em razão do tamanho do vetor, já que dava erro informando falha de segmentação e falha ao enviar os parâmetros, depois de pesquisar, identificamos que o tcp não suporta um grande volume de dados e assim mudamos para o protocolo udp e o código foi executado.