

Relatório - PSPD - Lab 1

Aluno

180028324 - Thiago Luiz de Souza Gomes

Introdução

O laboratório trata-se de resolver um problema que envolve a comunicação entre servido(res) utilizando a tecnologia RPC e TCP, que permite chamar um procedimento localizado em outra máquina pela rede como se fosse local.

O problema consiste em enviar um vetor com valores em ponto flutuante para o servidor e ter o retorno do maior e menor valor presente nele.

Solução geral

Solução adotada foi, pelo lado do servidor, uma função que recebe o vetor e retorna um par com o menor e maior valor.

Para atender esse problema, o arquivo .x (IDL) ficou da seguinte maneira:

```
RPC > ≡ projeto_1.x
1  #define SIZE 500
2  struct vetor{
3      double array[SIZE];
4  };
5
6  struct maior_menor{
7      double maior_menor[2];
8  };
9
10 program projeto_1{
11     version projeto_1{
12         maior_menor compara(vetor)=1;
13     }=43;
14
15 }=500;
16
```

A solução com o Socket ocorreu com a mesma lógica porém não foi utilizado struct.

Desenvolvimento do Projeto

Foi utilizado a biblioteca da math.h e stdlib.h para gerar os números aleatórios propostos no enunciado e foi implementada uma função para comparar o maior e menor número de um vetor no caso o servidor utilizava essa função enquanto o cliente preenchia o vetor, porém tive algumas limitações em conseguir enviar um vetor de 500 mil casas e por isso tive que reduzir o tamanho do vetor.

Thiago:

O laboratório foi interessante e um pouco desafiador, principalmente pela dificuldade de relembrar a matéria de FRC já que faz um tempo que eu fiz essa disciplina e por estar sozinho confesso que foi um pouco mais trabalhoso na questão de corrigir erros e debugar o código. No final não consegui passar um vetor com tamanho 500 mil pois não descobri como consertar o erro de Segmentation Fault que estava ocorrendo, porém acredito que me esforcei bastante para este laboratório e confesso que achei muito satisfatório ver um cliente e servidor conversando entre si, e muito interessante comparar o RPC e o TCP na questão de implementação.

Nota: 10/10