

PSPD

PESQUISA

LAB 1

PSPD

UNIVERSIDADE DE BRASÍLIA
PROGRAMAÇÃO PARA SISTEMAS PARALELOS E DISTRIBUÍDOS
FERNANDO WILLIAM
JOÃO VICTOR - 18/0103431
ÍTALO VINÍCIUS - 18/0102656



INTRODUÇÃO

ENTREGA 1

- SOCKET
- RPC

ENTREGA 2

- RPC (2 WORKERS)

SOCKET

STRUCT

```
1 typedef struct Dados{
2     float menor, maior;
3     int menorIndice, maiorIndice;
4 } Dados;
5
6 typedef struct Indice{
7     float numero;
8     int posicao;
9 } Indice;
```

SAÍDA

```
1 Indice do maior: 499205
2 Maior valor: 498517.906250
3 Indice do menor: 130445
4 Menor valor: 0.265625
```

CLIENTE

```
1 // Enviar vetor
2 Indice indice;
3 for(int i = 0; i < TAM; i++) {
4     indice.numero = vetor[i];
5     indice.posicao = i;
6     if(send(clienteSocket, &indice, sizeof(Indice), 0) < 0)
7         printf("Erro no send()\n");
8 }
```

SERVIDOR

```
1 // Recebe o vetor
2 Indice indice;
3 while (indiceAtual < TAM)
4 {
5     tamanhoRecebido = recv(socketCliente, &indice, sizeof(Indice), 0);
6
7     if (tamanhoRecebido == 0)
8     {
9         break;
10    }
11
12    vetor[indice.posicao] = indice.numero;
13    indiceAtual++;
14 }
15
16 dados = menor_maior(vetor);
17
18 send(socketCliente, &dados, sizeof(Dados), 0);
```

RPC

IDF

```
1 struct operandos{
2     float v[500000];
3     int len;
4 };
5
6 struct valor{
7     float menor;
8     float maior;
9     int indiceMaior;
10    int indiceMenor;
11 };
12
13 program PROG {
14     version VERS {
15         valor acharValores(operandos) = 1;
16     } = 110011;
17 } = 55000555;
```

CLIENTE

```
1 result = acharvalores_110011(&arg, clnt);
2 if (result == (valor *) NULL) {
3     clnt_perror (clnt, "call failed");
4 }
5
6 printf("Indice do maior: %d\n", result->indiceMaior);
7 printf("Maior valor: %f\n", result->maior);
8 printf("Indice do menor: %d\n", result->indiceMenor);
9 printf("Menor valor: %f\n\n", result->menor);
```

SERVIDOR

```
1 valor *
2 acharvalores_110011_svc(operandos *argp, struct svc_req *rqstp)
3 {
4     static valor result;
5
6     int i;
7     result.menor = argp->v[0];
8     result.maior = argp->v[0];
9
10
11     printf("Tamanho do vetor: %d\n", argp->len);
12
13     for (i = 0; i < argp->len; i++)
14     {
15         if (argp->v[i] < result.menor)
16         {
17             result.menor = argp->v[i];
18             result.indiceMenor = i;
19         }
20         if (argp->v[i] > result.maior)
21         {
22             result.maior = argp->v[i];
23             result.indiceMaior = i;
24         }
25     }
26
27     return &result;
28 }
```

**OBRIGADO PELA
ATENÇÃO!**