**Relatório Projeto Final de Sistemas Operacionais Compressed Network Communication.**

**Discentes: Angelo Ferro, Kaio Guilherme, Lucas Prado**

**Docente: Herbert Rocha**

Ciência da Computação – Universidade Federal de Roraima (UFRR) –

Departamento de Ciência da Computação

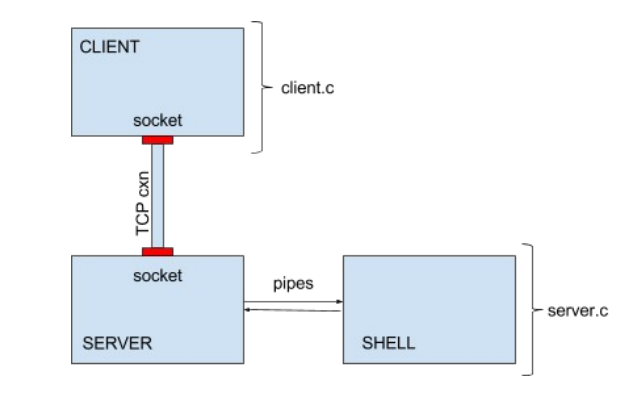
[lukasprado260@gmail.com](mailto:lukasprado260@gmail.com), [kaioguilherme444@gmail.com](mailto:kaioguilherme444@gmail.com), [Angelo.sonic@gmil.com](mailto:Angelo.sonic@gmil.com).

***Abstract****. This report describes the details of creating, configuring and maintaining a communication network through a telnet client and server through a TCP socket with the communication between them compressed.*

***Resumo.*** *Este relatório descreve os detalhes sobre a criação, configuração e manutenção de uma rede de comunicação através de um cliente e um servidor de telnet por meio de um socket TCP com a comunicação entre eles compactada.*

**1. Introdução**

O objetivo do projeto é criar e mostrar o funcionamento de uma comunicação compactada por meio da biblioteca Zlib com o intuito de reduzir a quantidade de dados de espaço ocupados entre um cliente e um servidor na linguagem de programação C, passando a entrada e saída através de um soquete TCP. Além disso, o servidor se comunicará com um shell onde eles serão conectados via pipes que irá receber os comandos do cliente enviado ao servidor e fornecer ao servidor a saída do comando recebido conforme a figura 1 abaixo.



**Figura 1. Estrutura do projeto**

**2. Desenvolvimento**

Para o desenvolvimento desse projeto foi escolhido a linguagem de programação C, juntamente com um laboratório feito na ferramenta Docker onde primeiramente foi criada uma Rede experimental com o IP 192.168.1.0, para hospedagem dos nossos containers através do comando:

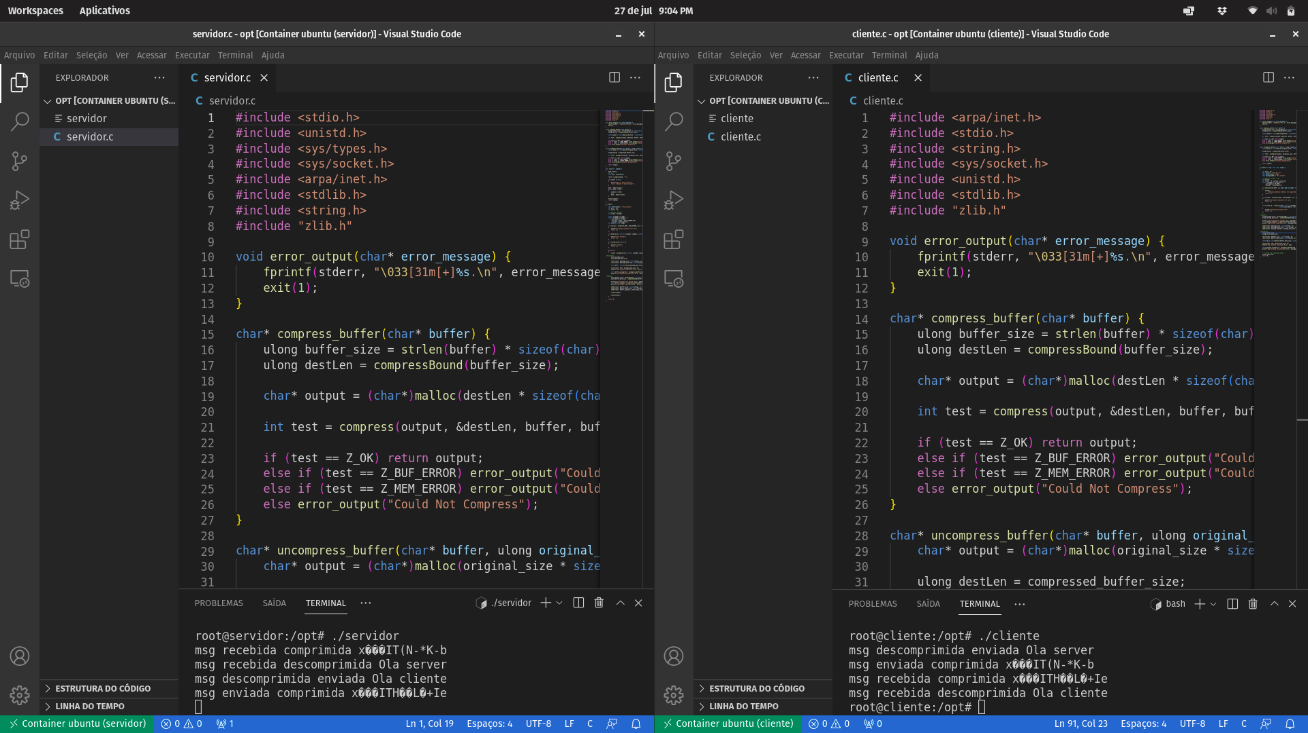
**docker network create --driver=bridge --subnet=192.168.1.0/24 rede\_experimental**

Depois disso, foram criados o container do cliente com o IP 192.168.1.2 e o container do servidor com o IP 192.168.1.3. A criação do container cliente foi feito por meio do comando:

**docker run -it --name cliente --hostname cliente --network rede\_experimental ubuntu**

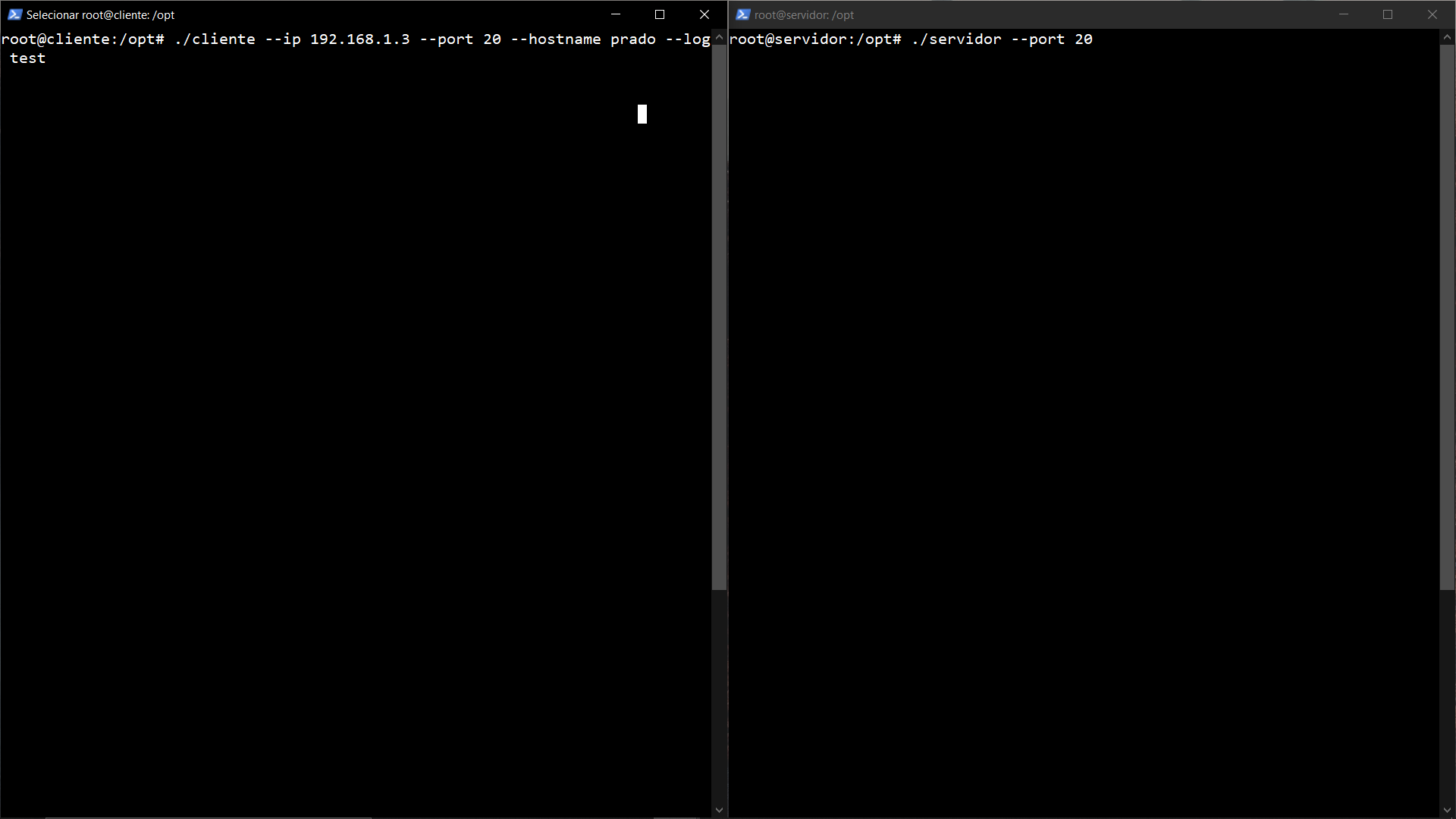
Depois foi feito o container do servidor na mesma rede através do mesmo comando acima, trocando somente o name e o hostname de “cliente” para “servidor”.

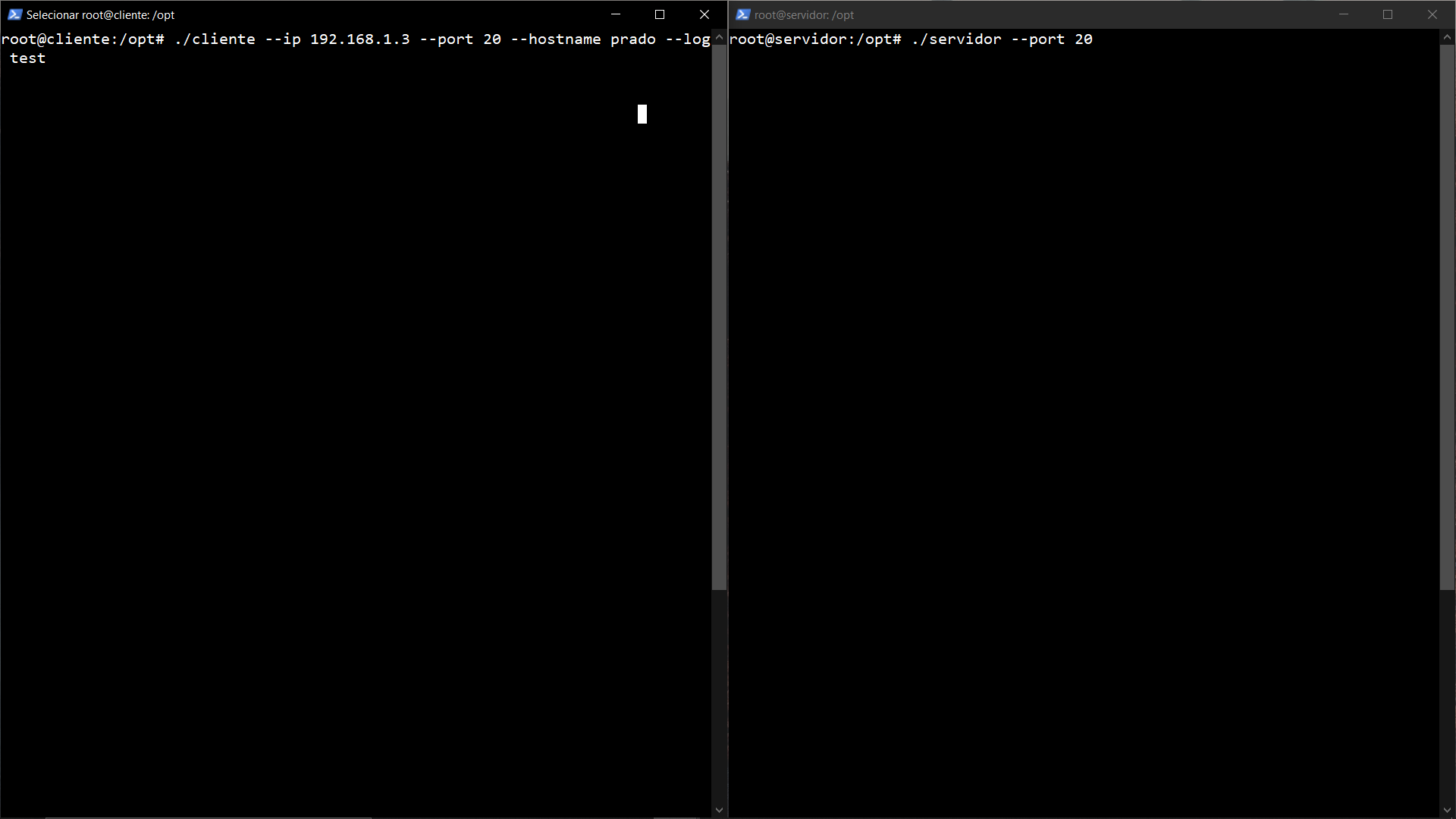
Após isso, foi desenvolvido o código do servidor e do cliente dentro do programa VSCode com as extensões do Docker para que ambos se comunicassem pelo soquete TCP onde estabelecessem uma comunicação e depois disso pudesse haver troca de informações de forma continua. Além disso, também foi chamada a função compress() para fazer a compressão da comunicação dos dois lados e a Uncompress() para descompressão das comunicação compactada, ambas da biblioteca Zlib conforme mostra na figura 2.



**Figura 2. Código do servidor e cliente rodando nos containers do Docker.**

E para garantir que a compactação está sendo feita foi adicionada uma opção --log ao cliente onde todos os dados gravados ou lidos do servidor deverão ser registrados no arquivo especificado. Já no programa do servidor foi feito uma função run() para receber os comandos do cliente criar uma comunicação pipe e redirecionar o resultado obtido para o endereço recebido no parâmetro da função.





**Figura 3. Teste da comunicação do código por meio dos comandos**

**3. Conclusão**

Apesar das dificuldades em elaborar o código, com os estudos e pesquisas feitas pelo grupo para propor a solução foi possível concluir o projeto de comunicação de rede compactada em um cliente e servidor abordando as funções propostas. Foi feito vários testes dentro da ferramenta do Docker envolvendo o programa para demonstrar as funcionalidades afim de concluir a aplicação tornando-o funcional. Além disso, pode ser de grande utilidade para incremento de demais funções como criptografia de comunicação para utilizar de maneira profissional no dia a dia.

**4. Referencias**

Jean-loup Gailly and Mark Adler (1995) “*zlib*”, <https://www.zlib.net/manual.html#Utility>, Julho.

[1] A Funcao getopt() e a Linha de Comando, by Copper Wire (Acessado em: Outubro 2011), <http://unlimitedinc.wikispaces.com/file/view/TUTORIAL_GETOPT%28%29.txt>

COMMER, DOUGLAS; STEVENS, DAVID Internetworking with TCP/IP: volume 3: client-server programming and applications Prentice Hall 1993