

Trabalho feito por:

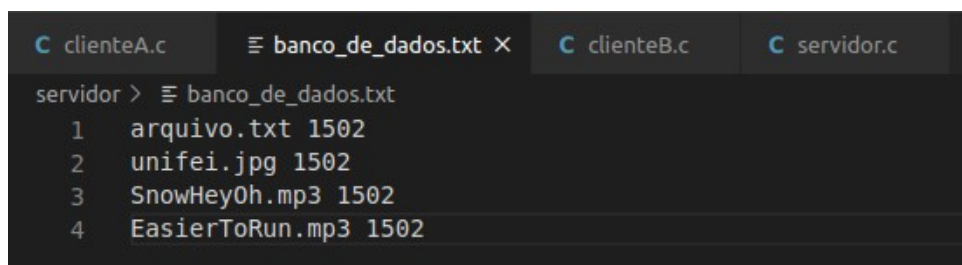
Nome: Rafael Greca Vieira

Matrícula: 2018000434

Considerações importantes

- Os arquivos deverão ser colocados na pasta do cliente B. Também deverá ser atualizado o arquivo txt do banco de dados! Caso o nome do arquivo seja colocado no banco de dados de forma incorreta, o programa não irá funcionar.
- O IP utilizado nos três programas foi o 127.0.0.1, porém cada programa está utilizando uma porta específica. O servidor utiliza a porta 1500, cliente A 1501 e cliente B 1502.
- Programa foi criado utilizando o sistema operacional Linux.
- O tamanho utilizado para o buffer foi de 1024 e 512 para o tamanho do segmento dos pacotes enviados.

O banco de dados está armazenado na pasta do servidor e possui a seguinte formatação:



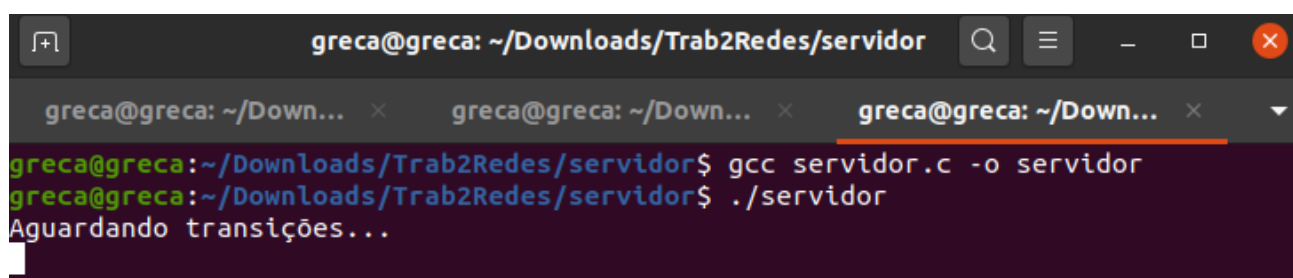
```
servidor > banco_de_dados.txt
1 arquivo.txt 1502
2 unifei.jpg 1502
3 SnowHey0h.mp3 1502
4 EasierToRun.mp3 1502
```

Deverá ser colocado o nome do arquivo e a porta do cliente (nessa caso é a porta do cliente B), separado por espaço e na mesma linha. Como mostra na figura a cima.

Passo a passo da execução

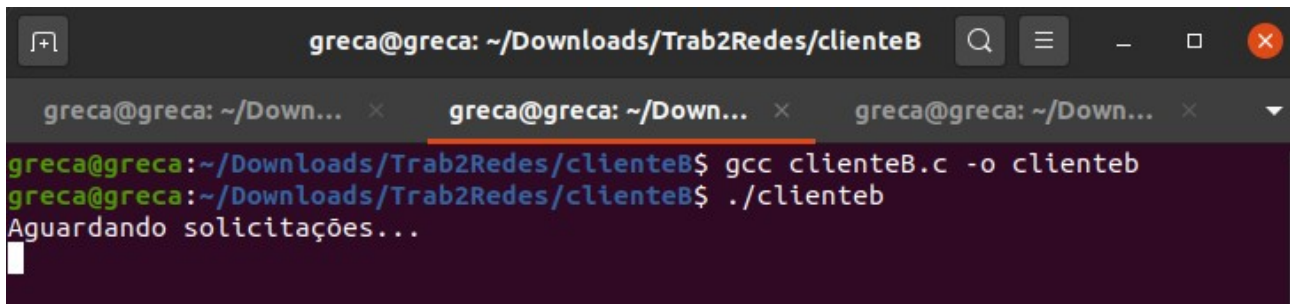
Antes de realizar o passo a passo, será necessário abrir três terminais. Um será usado para o cliente A, um para o cliente B e o outro para o servidor.

1º passo: Abrir o terminal na pasta do trabalho e acessar a pasta do servidor. Após ter feito isso, deverá ser executado o comando **gcc servidor.c -o servidor** para compilar o programa. E depois usar o comando **./servidor** para executar.



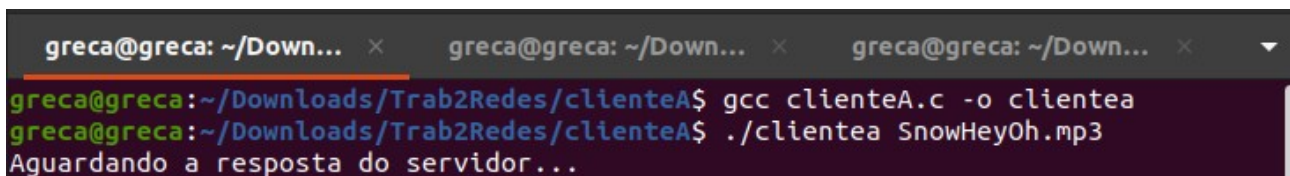
```
greca@greca: ~/Downloads/Trab2Redes/servidor
greca@greca: ~/Downloads/Trab2Redes/servidor$ gcc servidor.c -o servidor
greca@greca: ~/Downloads/Trab2Redes/servidor$ ./servidor
Aguardando transições...
```

2º passo: Abrir o terminal na pasta do trabalho e acessar a pasta do clienteB. Após ter feito isso, deverá ser executado o comando **gcc clienteB.c -o clienteb** para compilar o programa. E depois usar o comando **./clienteb** para executar.

A terminal window with a dark background and light-colored text. The title bar shows 'greca@greca: ~/Downloads/Trab2Redes/clienteB'. The terminal content shows the user running 'gcc clienteB.c -o clienteb' and then './clienteB'. The output is 'Aguardando solicitações...' followed by a cursor.

```
greca@greca: ~/Downloads/Trab2Redes/clienteB$ gcc clienteB.c -o clienteb
greca@greca: ~/Downloads/Trab2Redes/clienteB$ ./clienteB
Aguardando solicitações...
```

3º passo: Abrir o terminal na pasta do trabalho e acessar a pasta do clienteA. Após ter feito isso, deverá ser executado o comando **gcc clienteA.c -o clientea** para compilar o programa. E depois usar o comando **./clientea nome_do_arquivo** para executar.

A terminal window with a dark background and light-colored text. The title bar shows 'greca@greca: ~/Downloads/Trab2Redes/clienteA'. The terminal content shows the user running 'gcc clienteA.c -o clientea' and then './clientea SnowHeyOh.mp3'. The output is 'Aguardando a resposta do servidor...' followed by a cursor.

```
greca@greca: ~/Downloads/Trab2Redes/clienteA$ gcc clienteA.c -o clientea
greca@greca: ~/Downloads/Trab2Redes/clienteA$ ./clientea SnowHeyOh.mp3
Aguardando a resposta do servidor...
```

Se tudo estiver certo e o arquivo estiver no banco de dados, o programa irá rodar, mostrar a transferência dos pacotes e uma mensagem final dizendo que a transferência de arquivo foi bem sucedida.

4º passo: transferência realizada com sucesso! O novo arquivo será armazenado na pasta do cliente A com o mesmo nome que o arquivo original!

Explicação do trabalho:

O trabalho consiste em criar dois clientes, um servidor e realizar a transferência de arquivos entre os dois clientes, utilizando o protocolo UDP. O cliente A irá requisitar um arquivo para o servidor, depois irá acessar o banco de dados e verificar se possui o arquivo. E, se sim, retornará a porta do cliente que possui aquele arquivo. Se não, uma mensagem será retornada para o cliente A e o programa encerrado.

Após isso, o cliente A irá solicitar o cliente A para o cliente B (que possui o arquivo desejado) e a transferência do mesmo começará a ser feita. Depois de receber o arquivo, o cliente A irá enviar uma mensagem para o servidor e o mesmo irá atualizar o banco de dados.

Comentários a respeito do código:

No terminal na área do cliente A serão mostrados os pacotes que serão recebidos do cliente B, como mostra a imagem abaixo.

```
greca@greca: ~/Downl... x greca@greca: ~/Downl... x greca@greca: ~/Downl... x ▼
Pacote 709 recebido!
Numero sequencia: 709
Checksum: 1480
Tamanho: 512

Pacote 710 recebido!
Numero sequencia: 710
Checksum: 85
Tamanho: 512

Pacote 711 recebido!
```

No do cliente B, será mostrado a troca de mensagem entre ele e o cliente A. O cliente A irá retornar ack's com o número de sequência do pacote que recebeu e o cliente B irá mostrar qual pacote foi transferido, como mostrado na imagem abaixo.

```
greca@greca: ~/Downl... x greca@greca: ~/Downl... x greca@greca: ~/Downl... x ▼
Enviando o pacote 41 de numero de sequencia 41. Por favor, aguarde...
ack41 recebido do cliente A!

Enviando o pacote 42 de numero de sequencia 42. Por favor, aguarde...
ack42 recebido do cliente A!

Enviando o pacote 43 de numero de sequencia 43. Por favor, aguarde...
ack43 recebido do cliente A!
```

No servidor, será mostrado se o arquivo foi encontrado no banco de dados e se o banco de dados foi atualizado com a porta do novo cliente que recebeu o arquivo.

```
greca@greca:~/Downloads/Trab2Redes/servidor$ gcc servidor.c -o servidor
greca@greca:~/Downloads/Trab2Redes/servidor$ ./servidor
Aguardando transições...
Arquivo encontrado no IP: 1502
Porta do cliente que possui o arquivo enviado com sucesso

O cliente de porta 1501 agora possui o arquivo teste.txt!
Aguarde enquanto o banco de dados eh atualizado...

Banco de dados atualizado com sucesso!
```