

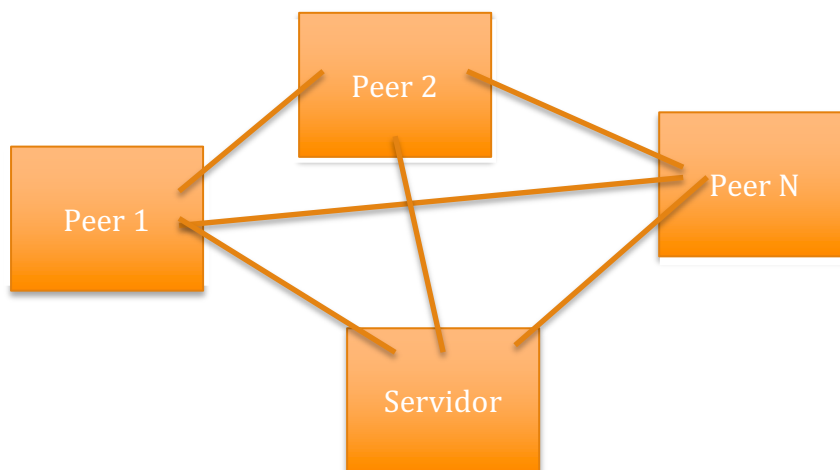
Programação com soquetes TCP

Objetivo: Aplicar os conhecimentos de programação com soquetes para desenvolver uma aplicação que realize a transferência de arquivos entre pares distribuídos. A aplicação seguirá, principalmente, uma arquitetura P2P.

Implementação e formato de entrega

- O trabalho final poderá ser implementado por uma dupla de alunos (no máximo).
- Para entrega final, a dupla deverá demonstrar o trabalho operacional. Usaremos os equipamentos da sala de aula para realizar a demonstração.
- Data limite para entrega e apresentação do trabalho: 28/julho/2022 15:20.
- O trabalho pode/deve ser apresentado antes do prazo final. Para tal, solicite um horário antecipadamente.
- Com qual linguagem devo implementar o trabalho? Você escolhe, apenas informe antecipadamente.

Cenário e descrição do trabalho



Você deverá implementar um aplicativo de transferência de arquivos que utilize uma arquitetura P2P. Todos os pares que desejam se conectar a rede devem solicitar acesso ao servidor (usando a arquitetura cliente/servidor). O servidor possui a lista dos pares que já estão participando da rede, bem como seus endereços IPs. Sempre que um par desejar sair da rede, deve primeiramente informar o servidor. Cada par possui uma lista de arquivos que podem ser transferidos para os outros pares. A listagem deve apresentar o nome do arquivo e sua extensão. Diversos arquivos podem/devem ser transferidos simultaneamente entre os pares.

Um exemplo de uso da aplicação pode ser visto abaixo (considere que o servidor já está em execução no hospedeiro 10.20.128.25, e que 10.20.128.30 é o endereço do hospedeiro do par):

```
$ ./peer 10.20.128.25 meus_arquivos.txt
S: OK. 10.20.128.30 adicionado a rede
C: peers
C: OK. Seus vizinhos são: 10.20.128.24, 10.20.128.26
C: ls 10.20.128.24
```

S: OK. 10.20.128.24. meus arquivos são: a.txt, b.doc, c.mp3
C: *get* 10.20.128.24 aaaaa.txt
S: KO. 10.20.128.24 não possui arquivo aaaaa.txt
C: *get* 10.20.128.24 a.txt
S: OK. 10.20.128.24 disponibilizando a.txt
S: OK. 10.20.128.24 disponibilizou a.txt
C: *rmmmr* 10.20.128.24 a.txt
S: KO. 10.20.128.24 não conhece comando rmmmr
C: *quit*
S: OK. 10.20.128.30 saindo

Ao inicializar o par informamos o endereço IP do servidor e a lista de arquivos que podemos disponibilizar. A lista de arquivos deve ser no formato (no exemplo foi usado meus_arquivos.txt):

a.txt /home/username/arquivos/a.txt
b.txt /home/username/arquivos/b.txt

Desta forma, podemos saber o local no qual cada arquivo está localizado. Após efetuar o *download* de um arquivo (comando *get*) ele deverá ser adicionado a lista de arquivos disponíveis.

A aplicação deve responder aos comandos:

peers: retorna a lista de pares vizinhos. Esta solicitação deve ser respondida pelo servidor.

ls <ip>: lista os arquivos disponibilizados pelo par <ip>

get <ip> <arquivo.txt>: solicita o download de <arquivo.txt> localizado no par <ip>

quit: solicita o termino da conexão com o servidor