

Trabalho 02 – Sockets UDP & TCP

Escolha uma das duas opções de exercício prático (exercício UDP ou TCP) e resolva através de uma linguagem de programação.

- Trabalho em individual ou grupo (até 2 alunos)
- Códigos iguais acarreta nota zero para ambas as equipes
- Pode ser utilizado qualquer linguagem de programação tal como:
python, c, c++, javascript.

UDP

Prática 1: servidor atende um cliente

1. Acessar os arquivos da pasta correspondente “UDPScktCoulouris”
1. Numa janela DOS rodar o servidor:
 - ir para o diretório onde se encontra o Servidor.jar
 - java -jar Servidor.jar
2. Verificar a execução do servidor (em outra janela) através de
 - netstat -a -b -p UDP
3. Rodar o cliente passando com argumentos uma mensagem e o endereço do servidor
 - Ex. java -jar Cliente.jar 127.0.0.1 6789 “MSG TST”

Prática 2: servidor atende vários clientes (multithread)

1. Acessar os arquivos da pasta correspondente “UDPScktCoulourisClienteMultiThread”
 2. Numa console DOS rodar o servidor:
 - ir para o diretório onde salvou o .jar
 - java -jar Servidor.jar
 3. Rodar o cliente passando com argumentos uma mensagem e o endereço do servidor.
- Duas threads clientes serão executadas.
- Ex. java -jar Cliente.jar 127.0.0.1 6789 “mensagem teste”

EXERCÍCIO UDP

- Fazer um sistema cliente-servidor para correção de questionários.
- O servidor deve ser capaz de receber várias conexões de clientes simultaneamente.
- O cliente envia ao servidor, vários datagramas contendo cada um uma resposta do tipo V ou F ao questionário, no seguinte formato:
 - <número da questão>;<número alternativas>;<respostas>
 - Exemplo:
 - 1;5;VVFFV

- 2;4;VVVV

O servidor lê a mensagem e calcula o número de acertos e erros devolvendo uma resposta simples:

- <número da questão>;<número acertos>;<número erros>

TCP

Prática 1: request-reply

1. Acessar os arquivos da pasta correspondente “TCPScktRequestReply”
2. No console DOS fazer:
 - ir para o diretório onde se encontra os .jar
 - Executar: java -jar Servidor ou //porta default 6789
 - Executar: java -jar Servidor <porta>
3. Em outra janela, rodar o cliente passando com argumentos uma mensagem e o endereço do servidor.
 - Ex. java -jar Cliente <SERVIDOR> <PORTA> <MENSAGEM>
 - Ex. java -jar Cliente localhost 6789 “MENSAGEM TESTE TCP”

Prática 2: Servidor de arquivo

1. Acessar os arquivos da pasta correspondente “TCPScktServidorArquivo” e acesse o arquivo .txt da pasta “dist”.
2. No console DOS fazer:
 - ir para o diretório onde se encontram os .jar
 - Executar: java -jar Servidor.jar
3. Rodar o cliente passando como argumentos o servidor, porta e seu nome
 - Ex. java -jar Cliente.jar <SERVIDOR> <PORTA> <NOME>
 - Ex. java -jar Cliente “localhost” 6789 CESAR
4. Executar vários clientes em consoles diferentes para verificar que o servidor atende a todos por ser multithread. Cada cliente recebe um stream de dados que está armazenado no arquivo teste.txt no servidor.

EXERCÍCIO TCP

Modifique o código do servidor de arquivos para que o cliente possa solicitar o arquivo que deseja baixar passando o nome do mesmo.

- Servidor pode retornar o arquivo (se existir) ou uma mensagem de arquivo não encontrado.