

Trabalho Prático

BCC361 - Redes de Computadores

2020/1

Professor Pedro Henrique Lopes Silva
Implementação de Sockets - 2 pontos

1 Instruções sobre o trabalho prático

Leia com atenção:

- Este trabalho de implementação deverá ser desenvolvido em grupo de no **máximo CINCO** integrantes.
- Grupos com número de integrantes maior do que **CINCO** terão a nota final do TP dividido pela quantidade de integrantes do grupo.
- Independente do grupo ter 5 ou menos integrantes, ou algum aluno opte por fazer individualmente, todos os requisitos pedidos no projeto deverão ser cumpridos.
- A entrega deve ser feita a partir do Moodle, em um arquivo zip.
- A data de entrega é **18/04/2021** (domingo).
- Estejam atentos a todas as recomendações constantes no presente enunciado.
- A implementação poderá ser realizada utilizando a linguagem desejada, desde que atenda às especificidades de cada protocolo (TCP e UDP). Em caso de dúvidas, entre em contato com o professor.

Bom trabalho!

2 Implementação sobre TCP

Implementar um par de programas que operem no modelo cliente-servidor sobre o protocolo TCP. O serviço a ser implementado é o de um *contador de vogais, consoantes e números* seguindo as especificações abaixo:

Servidor:

- O servidor deve começar em modo passivo em uma porta específica para receber solicitações de um cliente.
- O servidor ao receber uma mensagem que um cliente está ativo deve registrar esse cliente, o número de solicitações que ele enviará, e também passa a contar com ele como um colaborador.
- Ao receber uma solicitação de um cliente, o servidor seleciona, aleatoriamente, outro cliente ativo e transmite uma mensagem para que ele realize o processamento da string. O servidor deve registrar esta solicitação, e voltar a aguardar uma nova mensagem.
- Ao receber a resposta de uma solicitação de processamento de um cliente, o servidor deve identificar quem a solicitou e encaminhar a resposta ao mesmo.

- Após todas as solicitações aguardadas pelo servidor serem atendidas, o servidor deve enviar uma mensagem notificando os clientes que eles devem encerrar sua execução, terminando em seguida a sua própria execução.

Cliente

- Um cliente quando iniciado deve enviar uma mensagem ao servidor notificando que ele está ativo e o número de solicitações que ele enviará ao servidor. Ao receber esta notificação.
- Após notificar o servidor, o cliente envia a primeira string a ser processada, e fica aguardando uma mensagem do servidor.
- Ao receber a mensagem do servidor para realizar algum processamento de uma string deve-se:
 - Verificar se a string não é vazia e se contém caracteres alfa numéricos.
 - Se a string estiver no formato correto, o servidor deve retornar como resultado a quantidade de vogais, consoantes e números da string enviada. Por exemplo: Pedro5: C=3,V=2,N=5.
 - Se a string não possuir caracteres alfa numéricos, o servidor deve indicar ao cliente uma situação de erro, enviando uma mensagem avisando ao servidor que não é possível processar a string.

Sugestão:

Crie o seu próprio protocolo, por exemplo: um campo para o tipo de mensagem, outro se deu certo ou não, outro para o número de consoantes, outro para vogais e outro para a quantidade de números.

Você pode também só enviar uma string/bits simples e fazer o parse.

- Nota-se que o cliente pode receber três tipos de mensagens:
 - Resposta de uma solicitação enviada ao servidor: neste caso, ele deve imprimir o resultado na tela e enviar uma nova solicitação ao servidor. O cliente deve receber como entrada um conjunto de mensagens, a quantidade de mensagens definidas deve estar contida na primeira mensagem enviada ao servidor.
 - Solicitação de processamento de uma string: neste caso, o cliente responde o servidor e aguarda uma nova mensagem.
 - Notificação de encerramento: neste caso, todas as solicitações foram atendidas e o cliente pode encerrar sua execução.

As informações que serão enviadas pelos clientes quantos as que serão salvas (saída) devem seguir o padrão de dois arquivos em anexo. Os arquivos são exemplos que deverão ser rigorosamente seguidos para avaliação, uma vez que os arquivos de outros grupos serão usados para testar a sua implementação.

3 Experimentos

Para realização dos experimentos deverá ser usado como entrada um arquivo de texto conforme exemplo disponibilizado pelo professor. Cada grupo deve definir outras entradas no arquivo (entradas de outros grupos serão usadas para testar a sua implementação). Há também um arquivo modelo de saída no qual seu programa deverá salvar os resultados. A implementação deste é obrigatória para fins de testes e validação do professor. Utilize esse arquivo para validar seus resultados ou até mesmo registrar métricas definidas pelo grupo.

Os testes da sua implementação podem ser executados na mesma máquina, isto é, a mesma máquina pode conter um servidor e vários clientes.

4 Relatório

O relatório deverá conter uma descrição do problema, solução implementada pelo grupo, os cenários em que os experimentos foram aplicados, análise dos resultados e conclusão. A organização do relatório também será ponto de avaliação. Quanto mais informativo estiver o texto, melhor será para o professor avaliar a solução.

Atenção: As entradas para os clientes podem ser inseridas por meio de um arquivo texto que contenha as strings, uma a cada linha do arquivo, a serem processadas. O número de clientes e mensagens não deve ser tão pequeno de forma que o teste leve milissegundos para executar, nem tão grandes de forma que o teste demore dias para executar. Utilizem o bom senso e realizem testes consistentes. As respostas enviadas pelos servidores devem ser gravadas pelos respectivos clientes. O uso desses arquivos podem facilitar a identificação de bugs ou falhas de rede ao longo das execuções.

5 Entrega

A entrega dos arquivos gerados deverá ser feita pelo Moodle, compactados em um arquivo zip, devidamente separados em pastas de forma organizada, que facilite o entendimento e uso dos mesmos. Para facilitar a correção e execução, forneça todos os arquivos de código-fonte e arquivos de execução (class, exe, etc.). Forneça também arquivos bat ou sh para a execução de diferentes testes, com número variado de clientes e de arquivos de mensagens, além de arquivos texto com instruções que se façam necessárias. O relatório também deve ser enviado compactado no arquivo zip. Somente um arquivo zip poderá ser enviado.

PONTO EXTRA

O trabalho prático foi focado no protocolo TCP, entretanto, ainda há o protocolo UDP. Pensando nesse sentido, será oferecido **um ponto** extra para quem implementar as especificações do serviço descrito para o protocolo TCP (Seção 2) utilizando o Protocolo UDP. Para isso, novas especificações devem ser adicionadas:

- Toda a comunicação é por meio do protocolo UDP.
- Tanto servidor quanto clientes devem possuir *timeout*, para encerrar sua execução em caso de um longo período de inatividade.
- Tempos de espera aleatórios antes do envio de mensagens devem ser adicionados para simular perda de pacotes e atrasos. Isso deve ser feito uma vez que os experimentos ocorrerão somente em uma máquina.

Também deve ser gerado um relatório nos moldes descritos na Seção 4. Não precisa criar um novo relatório, basta adicionar ao relatório já existente.