



Roteiro de Aula Prática – Programação com Sockets

DISCIPLINA: DCA0130 – Redes de Computadores
PROFESSOR: Carlos Manuel Dias Viegas

Esta prática consiste em uma introdução ao desenvolvimento de aplicações de rede por meio de programação com Sockets.

- Os requisitos para a realização desta prática são a instalação do Python na versão 3 e ter assistido às videoaulas sobre o desenvolvimento de aplicações de rede disponibilizadas no SIGAA;
- É importante lembrar que a realização desta prática deve ser gravada em vídeo. Como sugestão, assistam às instruções no seguinte vídeo: <https://youtu.be/0zzt2QWjedY> (outros softwares de captura de vídeo podem ser utilizados);
- Durante a gravação do vídeo, à medida que as tarefas forem sendo realizadas, vocês devem narrar como estão fazendo e os resultados encontrados. Nesta prática em específico, pretende-se que seja apresentado o funcionamento das aplicações desenvolvidas. **O vídeo NÃO PODE ter duração superior a 5:00 minutos;**
- O vídeo deverá ser submetido até o dia 29/06/2020 na plataforma indicada pelo professor. O link para o vídeo deverá ser informado em uma tarefa específica no SIGAA.
- Os códigos fonte desenvolvidos também deverão ser enviados na tarefa criada no SIGAA, juntamente com o link.

Em todas as tarefas desta prática devem ser executados um cliente e um servidor. Esta execução pode ser feita em um mesmo computador (ou seja, rodando o cliente e o servidor no mesmo) ou em computadores/dispositivos diferentes (o cliente rodando em um computador e o servidor em outro). Esta escolha fica a critério de vocês.

Os códigos fonte (em Python) necessários para iniciar esta prática estão disponíveis na seguinte página:

<https://www.dca.ufrn.br/~viegas/disciplinas/DCA0130/files/Sockets/>

Lembrando que, obrigatoriamente, deverão ser utilizados como base os códigos fonte disponibilizados acima!

Tarefa A: Desenvolver um servidor de data

Faça as alterações necessárias tanto no cliente quanto no servidor para que o cliente envie uma solicitação ao servidor e este responda com a data e o horário do sistema.

- Implemente utilizando sockets UDP;
- Utilize a biblioteca `import time` e o método `time.ctime()` para capturar a hora;
- Atente que será necessário converter o método `time.ctime()` em string por meio do método `str(): str(time.ctime())`
- O cliente deve digitar o comando: `data` e aguardar o servidor responder com a data. Outros comandos não devem ser aceitos.

Tarefa B: Desenvolver um servidor de arquivos

Faça as alterações necessárias tanto no cliente quanto no servidor para que o cliente envie uma solicitação ao servidor e este responda com conteúdo de um arquivo texto.

- Implemente utilizando sockets TCP;
- Crie um arquivo de texto simples (por exemplo: `arquivo.txt`) e escreva alguma informação em 1 linha nesse arquivo;
- Faça com que o servidor leia o arquivo (local) e retorne o seu conteúdo para o cliente quando este digitar o comando: `obter arquivo.txt`. Outros comandos não devem ser aceitos;
- Use o método `open(arquivo.txt)` para abrir o arquivo solicitado e o método `.read()` para ler o seu conteúdo.

Tarefa C: Desenvolver um acesso remoto

Faça as alterações necessárias tanto no cliente quanto no servidor para que o cliente envie um comando para o servidor e este o execute localmente.

- a. Implemente utilizando sockets TCP;
- b. Utilize a biblioteca `import os` e o método `os.system(comando)`
- c. O método `os.system(comando)` tem como parâmetro de entrada o comando que o cliente digitar, por exemplo: cliente digita `ls` ou `dir` e o servidor recebe e passa como parâmetro para `os.system('ls')` ou `os.system('dir')` e exibirá o resultado na própria tela do servidor (não é necessário retornar para o cliente neste caso);
- d. O servidor deverá aceitar qualquer comando válido no sistema, não apenas o `ls / dir`.