UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE

CENTRO DE TECNOLOGIA

DEPARTAMENTO DE COMPUTAÇÃO E AUTOMAÇÃO

**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 (versão 3) e ter assistido às videoaulas sobre o desenvolvimento de aplicações de rede disponibilizadas no SIGAA;

- Esta prática deve ser realizada em duplas, podendo ser formadas por alunos de diferentes turmas da disciplina de redes de computadores (DCA0130);

- Esta prática consiste em realizar as tarefas descritas abaixo e desenvolver o respectivo código fonte.

- Os códigos fonte desenvolvidos deverão ser enviados em uma tarefa específica no SIGAA até o dia **04/07/2021**.

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.

* 1. Implemente utilizando sockets UDP;
  2. Importe a biblioteca timee utilize o método time.ctime()para capturar a hora;
  3. Atente que será necessário converter o método time.ctime() em string por meio do método str(): str(time.ctime())
  4. O cliente deve digitar o comando: datae 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.

* 1. Implemente utilizando sockets TCP;
  2. Crie um arquivo de texto simples (por exemplo: arquivo.txt) e escreva alguma informação em 1 linha nesse arquivo;
  3. 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;
  4. 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.

* 1. Implemente utilizando sockets TCP;
  2. Importe a biblioteca subprocess e utilize o método subprocess.check\_output(**comando**, shell=True, universal\_newlines=True, stderr=subprocess.STDOUT)
  3. O método subprocess.check\_output(**comando**, ...) tem como parâmetro de entrada o comando que o cliente digitar, por exemplo: cliente digita ls ou dir e o servidor recebe e enviará o resultado para a tela do cliente;
  4. O servidor deverá aceitar qualquer comando válido no sistema, não apenas o ls / dir.