

## TESTE DE PERFORMANCE 3



**Engenharia da Computação**  
**Projeto em Arquitetura de Computadores,**  
**Sistemas Operacionais e Redes**  
Kaio Henrique Silva da Cunha  
Prof.: Alcione Dolavale

Fortaleza, CE  
28/11/2021

## 1. O que é um processo cliente?

Um cliente é uma máquina ou um programa que solicita serviços por meio da web.

O processo do cliente é executado na máquina do cliente e emite solicitações HTTP uma após a outra, sem nenhum tempo para pensar entre elas. Isso significa que os clientes enviam solicitações tão rápido quanto o proxy pode tratá-las.

## 2. O que é um processo servidor?

Um servidor é uma máquina ou um programa que fornece serviços aos clientes de acordo com as solicitações do cliente. Um processo servidor atende à solicitação do cliente executando a tarefa solicitada. Os programas de servidor geralmente recebem solicitações de programas clientes, realizam recuperação e atualizações de banco de dados, gerenciam a integridade dos dados e despacham respostas a solicitações de clientes.

## 3. A função `socket()` do módulo 'socket' de Python é responsável por criar um socket no processo tanto para protocolo TCP, quanto UDP.

**Como diferenciar se o socket a ser criado é TCP e UDP?**

A seguir está um trecho de socket python:

```
import socket
socket.socket(socket.AF_INET, socket.SOCK_STREAM)
```

O segundo argumento determina o tipo de socket; `socket.SOCK_DGRAM` é UDP, `socket.SOCK_STREAM` é um socket TCP. Isso tudo desde que se esteja usando uma família de sockets `AF_INET` ou `AF_INET6`.

## 4. Para sockets TCP, responda:

- a. Que sequência de chamadas de funções em Python deve ser realizada pelo cliente? (Não precisa especificar os parâmetros).**

Na ordem: socket, gethostname, connect e send.

- b. Que sequência de chamadas de funções em Python deve ser realizada pelo servidor? (Não precisa especificar os parâmetros).**

Na ordem: socket, gethostname, bind, listen, accept e recv.

- c. Quais destas funções são bloqueantes, isto é, o processo fica esperando?**

Recv é bloqueante a não ser que se especifique um timeout.

## **5. Para sockets UDP, responda:**

- a. Que sequência de chamadas de funções em Python deve ser realizada pelo cliente? (Não precisa especificar os parâmetros).**

Na ordem: socket(para criar o socket), connect, send, recv e close.

- b. Que sequência de chamadas de funções em Python deve ser realizada pelo servidor? (Não precisa especificar os parâmetros).**

Na ordem: socket, bind, recvfrom, e sendto.

- c. Quais destas funções são bloqueantes, isto é, o processo fica esperando?**

Recv é bloqueante a não ser que se especifique um timeout.

## **6. Para que serve o comando socket.bind()?**

O comando bind() associa o socket com seu endereço local. É por isso que o lado do servidor se liga, para que os clientes possam usar esse endereço para se

conectar ao servidor. Comparando ao `connect()`, `bind ()` é para o próprio endereço, `connect ()` para o endereço remoto.

## **7. Em sockets Python, como é representado um endereço de um processo remoto?**

O comando `connect()` é usado para conectar a um endereço remoto, é por isso que está do lado do cliente.