

CK0249 - REDES DE COMPUTADORES I

Unidade 2 - Programação em Sockets

Gabriel Camurça Fernandes de Sousa - 420549

Junho 20, 2021

1 Código

```
1  from socket import *
2  import sys
3
4  def main():
5      server_host = ''
6      server_port = 3000
7      socket_instance = socket(AF_INET, SOCK_STREAM)
8      socket_instance.bind((server_host, server_port))
9      socket_instance.listen(1)
10     print("Servidor iniciado e pronto para estabelecer uma nova conexão")
11     while True:
12         conexao, endereco = socket_instance.accept()
13         print("Server conectado por", endereco)
14         try:
15             content = conexao.recv(1024).decode()
16             filename = content.split()[1]
17             file = open(f'./musicas/{filename[1:]}.txt', 'r')
18             output_data = file.read()
19             conexao.send("HTTP/1.1 200 OK\r\n\r\n".encode())
20             for i in range(0, len(output_data)):
21                 conexao.send(output_data[i].encode())
22             conexao.send("\r\n".encode())
23             print("Lyrics enviadas com sucesso!")
24             conexao.close()
25         except IOError:
26             print("Não foi possível carregar as lyrics da musica.")
27             conexao.send("HTTP/1.1 404 Not Found\r\n\r\n".encode())
28             conexao.send("<html><head></head><body><h1>404 Not Found</h1></body></html>\r\n".encode())
29             conexao.close()
30     socket_instance.close()
31
32 if __name__ == '__main__':
33     main()
```

2 Explicação

Encontrado no arquivo ‘server.py’, o código em questão trata do que o servidor deve fazer uma vez recebida a entrada do cliente pelo browser.

Em particular, foi implementado um sistema simples para mostrar as lyrics de uma música.

Primeiramente, definimos o qual host e port serão utilizados pelo server, ou seja, a identificação dele para que o cliente saiba para onde fazer as requisições. No nosso caso, podemos utilizar a notação de uma string vazia para indicar que o endereço a ser utilizado é o da própria máquina; assim como especificamos um dos ports disponíveis da máquina: 3000.

Então, criamos um objeto socket passando como argumentos: `AF_INET`, para indicar que será utilizado o protocolo TCP; e `SOCK_STREAM`, indicando que será utilizado o protocolo IP

Fazendo a atribuição das informações de host e port ao socket, deixamos o servidor ‘escutando’ por uma possível conexão.

Ele então deve ficar disponível para que requisições sejam feitas pelo cliente de forma que mais de uma conexão possa ser feita sem fechar o servidor após a primeira terminar. Isso é feito através do loop infinito. Dentro dele conexões serão estabelecidas e encerradas.

É então que o tratamento da mensagem recebida ocorre: guardamos a mensagem e a analisamos para saber qual o nome da música especificada pelo cliente. Para tanto, decodificamos a mensagem (que estava na forma de bytes pura) e tentamos abrir o arquivo correspondente à música. Se não conseguirmos, lançamos uma mensagem de volta para o cliente informando que não foi possível achar a informação que ele procurava; caso contrário, lemos o arquivo e mandamos o seu conteúdo (lyrics) numa mensagem também de volta para o cliente. Essas mensagens que voltam para o cliente são adicionadas de um Header HTTP com códigos 200 ou 404 e, tanto o Header quanto o conteúdo em si, são encriptados novamente para a forma de bytes puro, naturalmente.

Então o servidor fecha a conexão estabelecida e fica à espera de uma nova.