



Redes de Computadores

PROGRAMAÇÃO COM SOCKETS

Sockets

- ▶ Usados para comunicação de processos em sistemas distribuídos

```
import socket
```

```
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
```

- ▶ O primeiro parâmetro indica que o socket usará endereços IPv4
- ▶ O segundo parâmetro indica que o socket usará o protocolo TCP
 - ▶ Para usar UDP, o parâmetro deveria ser socket.SOCK_DGRAM

Sockets

- ▶ A seguir, há o exemplo de um programa que usa um socket para se conectar ao servidor Web do Google

```
2  # An example script to connect to Google using socket
3  # programming in Python
4  import socket # for socket
5  import sys
6
7  try:
8      s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
9      print ("Socket successfully created")
10 except socket.error as err:
11     print ("socket creation failed with error %s" %(err))
12
13 # default port for socket
14 port = 80
15
16 try:
17     host_ip = socket.gethostbyname('www.google.com')
18 except socket.gaierror:
19
20     # this means could not resolve the host
21     print ("there was an error resolving the host")
22     sys.exit()
23
24 # connecting to the server
25 s.connect((host_ip, port))
26
27 print ("the socket has successfully connected to google")
28
```

Aplicação cliente-servidor

- ▶ Vamos criar um programa cliente e um programa servidor
- ▶ Ambos irão se comunicar via sockets

Servidor



```
1  from socket import *
2
3  serverPort = 12000#define em qual porta o socket irá escutar
4
5  serverSocket = socket(AF_INET,SOCK_STREAM)#cria um socket TCP
6
7  serverSocket.bind(('',serverPort))#atrela ao socket o localhost e a porta de escuta
8
9  serverSocket.listen(1)#começa a escutar
10
11  print('The server is ready to receive')
12
13  while 1:
14      connectionSocket, addr = serverSocket.accept()
15      sentence = connectionSocket.recv(1024)
16      capitalizedSentence = sentence.upper()
17      connectionSocket.send(capitalizedSentence)
18      connectionSocket.close()
```


Cliente

```
1  from socket import *
2
3  #ip do socket
4  serverName = "127.0.0.1"
5  #cria socket TCP, porta 12000
6  serverPort = 12000
7
8  clientSocket = socket(AF_INET, SOCK_STREAM)
9
10 clientSocket.connect((serverName,serverPort))
11
12 sentence = input("Digite algo em letra minúscula:")
13
14 #Envia a mensagem
15 sen = sentence.encode()#formata a mensagem
16 clientSocket.send(sen)#envia
17 modifiedSentence = clientSocket.recv(1024)#recebe a resposta
18 print("From Server: ", modifiedSentence.decode())
19 clientSocket.close()#encerra a conexão
```

Aplicação Cliente-Servidor

- ▶ Primeiro, execute a aplicação Servidor
- ▶ Após isso, execute a aplicação cliente

Envio e recebimento de dados

- ▶ `send()` – envia um segmento TCP
- ▶ `sendall()` – divide a mensagem na quantidade necessária de segmentos TCP e envia todos
- ▶ `recv()` – recebe um segmento TCP pelo socket

Exercício

- ▶ Faça uma calculadora no modelo cliente-servidor
- ▶ O cliente deve enviar uma operação a ser realizada pelo servidor por meio de uma mensagem no seguinte formato:
 - ▶ OP Num1 Num2
 - ▶ onde OP é uma das 4 operações básicas e Num1 e Num2 são os operandos
 - ▶ Exemplo:
 - ▶ SOMA 2 2
 - ▶ SUB 5 3
 - ▶ MUL 2 2
 - ▶ DIV 2 1
- ▶ O servidor deve interpretar a mensagem recebida, realizar a operação e devolver o resultado ao cliente
- ▶ As mensagens trocadas entre cliente e servidor devem ser enviadas como strings

Dúvidas?

