Comunicação entre Processos com Socket em Python Sistemas Operacionais

Charles Tim Batista Garrocho

Instituto Federal do Paraná – IFPR Campus Goioerê

charles.garrocho.com/SO2016

charles.garrocho@ifpr.edu.br

Técnico em Informática



Socket: Visão Geral

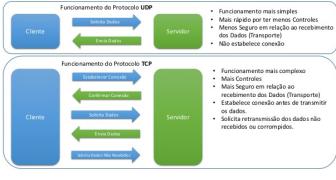
- Permite a troca de mensagens entre processos executando em hosts diferentes.
- Um socket deve ser associado (bind) a um endereço IP e a uma porta (0-65535).
- As portas 0 a 1023 são as portas reservadas por processos (FTP 21, SMTP 25, HTTP 80).
- Um programa servidor cria um socket e o mantém escutando (listening) em determinada porta.
- O programa cliente cria um socket, conecta-o ao servidor (ip/porta) e envia/recebe dados.



Socket: Serviços de Transporte

Há dois tipos de serviços de transportes via sockets:

- Transmission Control Protocol (TCP): confiável, orientado a conexão.
- User Datagram Protocol (UDP): n\u00e3o confi\u00e3vel, cada datagrama \u00e9 um registro \u00fanico e indivis\u00e3vel.



Socket: Implementação em Python

Acesse: http://www.python.org.br/wiki/SocketBasico

Importando a biblioteca socket do Python import socket

Criando um socket TCP

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

Criando um socket UDP

 $s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)$



Socket: Cliente TCP

```
import socket

HOST = '127.0.0.1'
PORT = 5000
tcp = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
dest = (HOST, PORT)
tcp.connect(dest)
```

```
print 'Para sair, digite: SAIR'
msg = raw_input()
while msg != 'SAIR':
tcp.send (msg)
msg = raw_input()
```



tcp.close()

Socket: Servidor TCP

```
import socket
HOST = "
PORT = 5000
tcp = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
orig = (HOST, PORT)
tcp.bind(orig)
tcp.listen(1)
while True:
con, cliente = tcp.accept()
print 'Concetado por', cliente
while True:
msg = con.recv(1024)
if not msg: break
print cliente, msg
print 'Finalizando conexao do cliente', cliente
con.close()
```

Exercícios

Atividade para aula:

- Baixe em seu computador a implementação em Python do Cliente e do Servidor TCP em Python.
- Após baixar, encontre um colega, e combine com seu colega para que ele seja um Servidor e você um Cliente, ou vice-versa.
- Vocês deveram utilizar a mesma porta no código, o a pessoa que for o cliente deverá saber o endereço IP da pessoa que for o Servidor.
- Vocês deveram se comunicarem de forma que o Cliente envie uma mensagem e o Servidor receba essa mensagem.

Atividade extra classe:

• Criar um chat onde o Servidor receba e encaminhe várias mensagens simultaneamente (utilizando threads) para seus Clientes.