

BCC722

Programação de Sistemas em Tempo Real

Introdução a Socket em Python

Prof. Charles Garrocho

Socket

- 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).
- 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

- **Há dois tipos de serviços de transportes via sockets:**
 - *Transmission Control Protocol* (**TCP**): confiável, orientado a conexão.
 - *User Datagram Protocol* (**UDP**): não confiável, cada datagrama é um registro único e indivisível.

Socket

Import socket

dir(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)

- **Acesse:** <http://www.python.org.br/wiki/SocketBasico>

Cliente TCP

```
import socket
HOST = '127.0.0.1'    # Endereco IP do Servidor
PORT = 5000           # Porta que o Servidor esta
tcp = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
dest = (HOST, PORT)
tcp.connect(dest)
print 'Para sair, digite: SAIR\n'
msg = raw_input()
while msg != 'SAIR':
    tcp.send (msg)
    msg = raw_input()
tcp.close()
```

Servidor TCP

```
import socket
HOST = "          # Endereco IP do Servidor
PORT = 5000      # Porta que o Servidor esta
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()
```