“um socket é a combinação de um endereço IP, um protocolo e o número da porta do protocolo”

Material de apoio utilizado nas aulas de Sistemas Distribuidos – UNESA – Petrópolis I

Prof. Luis Rodrigo de O. Goncalves
luisrodrigoog@gmail.com

Atualizado em : 26/03/12 - Impresso em : 26/03/12

Sockets - Conceitos:

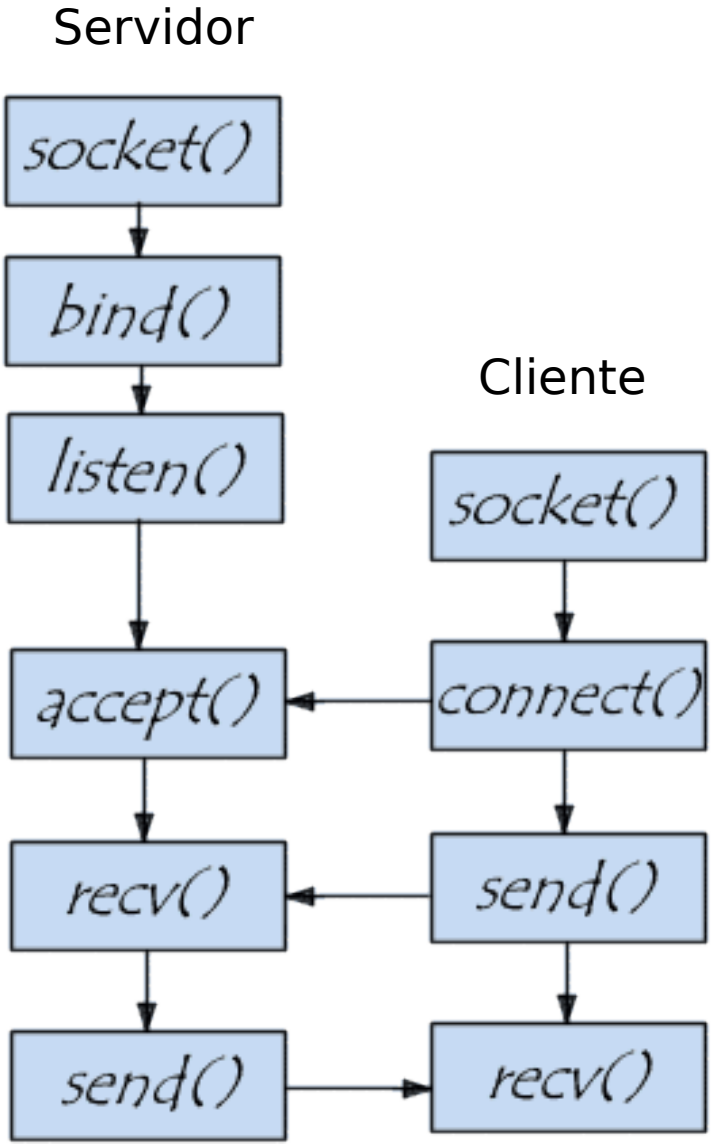
- 1ª Implementação: Unix BSD;
- Berkeley Sockets;
- Possibilita a comunicação bi-direcional;
- Esconde os detalhes de baixo nível;
- Baseados nos descritores de arquivo;
- Chamadas: send () e recv ();
- Tipos Básicos: DARPA, Socket Unix, X.25.
- Internet Sockets: Stream Sockets,
  Datagram Sockets
  e Raw Sockets

Prof. Luis Rodrigo
luisrodrigoog@gmail.com
http:// www.lncc.br/~lrodrigo

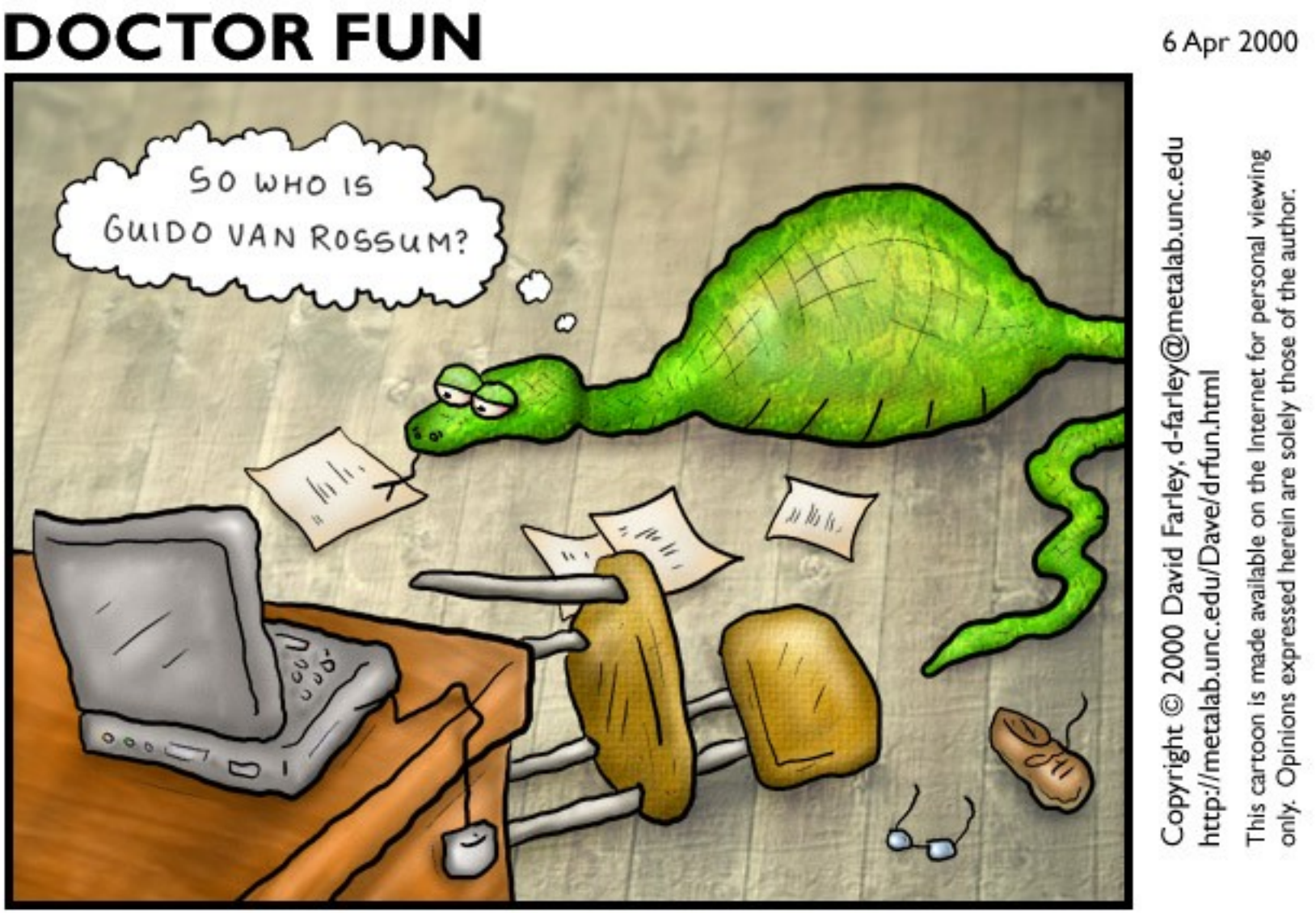
Sockets - Conceitos:

- Datagram Sockets (UDP):
    - Sem conexão;
    - Transferência pacote por pacote;
    - Mais rápido;
- Stream Sockets (TCP) :
    - Com conexão;
    - Fluxo confiável de dados;
    - Envio de uma grade quantidade de dados;
- Raw Sockets :
    - permite acesso aos protocolos de Rede;

Prof. Luis Rodrigo
luisrodrigoog@gmail.com
http:// www.lncc.br/~lrodrigo



Prof. Luis Rodrigo
luisrodrigoog@gmail.com
http:// www.lncc.br/~lrodrigo



Prof. Luis Rodrigo
luisrodrigoog@gmail.com
http:// www.lncc.br/~lrodrigo

Módulos Necessários:

```
// obtendo acesso a interface BSD Sockets
import socket
// obtendo acesso a interface do sistema
import sys
```

Prof. Luis Rodrigo
luisrodrigoog@gmail.com
http:// www.lncc.br/~lrodrigo

Criando o descritor:

dsSocket = socket.socket (family, type)

onde:

dsSocket : contem o valor do descritor;

family: AF_UNIX / AF_INET

type: SOCK_STREAM, SOCK_DATAGRAM

dsSocket= socket.socket (socket.AF_INET,
socket.SOCK_STREAM)

Prof. Luis Rodrigo
luisrodrigoog@gmail.com
http:// www.lncc.br/~lrodrigo

Definindo o endereço:

host="127.0.0.1"

porta="4935"

descEnd = (host , porta)

onde:

host : endereço IP da máquina

porta: porta onde ocorrerá as conexões

descEnd: é a tupla que identifica o endereço

Prof. Luis Rodrigo
luisrodrigoog@gmail.com
http:// www.lncc.br/~lrodrigo

Associando a porta ao processo:

descSock.bind (descEnd)

onde:

descSock : é o descritor criado anteriormente

descEnd: é a tupla que identifica o endereço

endIP="127.0.0.1"

porta="4935"

endServidor (endIP, porta)

sockServidor.bind (endServidor)

Prof. Luis Rodrigo
luisrodrigoog@gmail.com
http:// www.lncc.br/~lrodrigo

Escutando a porta:

descSock.listen (backlog)

onde:

backLog : número máximo de conexões da fila de espera;

sockServidor.listen (1)

Prof. Luis Rodrigo
luisrodrigoog@gmail.com
http:// www.lncc.br/~lrodrigo

Aceitando Conexões

sockCliente , endCliente = descSock.accept ( )

onde:

sockCliente : novo socket associado ao cliente;

endCliente : end. remoto do socket cliente;

Prof. Luis Rodrigo
luisrodrigoog@gmail.com
http:// www.lncc.br/~lrodrigo

Conectando-se ao Servidor

descSocket.connect (endServidor)

onde:

descSocket : descritor anteriormente criado;

endservidor : tupla com os dados do endereço do servidor;

sockCliente.connect (endServidor)

Prof. Luis Rodrigo
luisrodrigoog@gmail.com
http:// www.lncc.br/~lrodrigo

Fechando a conexão:

descSocket.close ( )

onde:

descSocket : descritor do socket a ser encerrado;

descSocket.shutdown (como)

onde:

como : 0 – impede o recebimento;

1 – impede o envio;

2 – impede o envio e o recebimento.

Prof. Luis Rodrigo
luisrodrigoog@gmail.com
http:// www.lncc.br/~lrodrigo

Enviando dados:

qtBytes = descSocket.send (mensagem)

onde:

descSocket : descritor anteriormente criado;

mensagen : conteúdo da msg a ser enviada;

qtBytes : quantidade de bytes enviados;

msg= “Olá, mundo !!! \n ”

bytesEnviados = sockCliente.send (msg)

Prof. Luis Rodrigo
luisrodrigoog@gmail.com
http:// www.lncc.br/~lrodrigo

Recebendo dados:

mensagem = descSocket.recv (qyBytes)

onde:

descSocket : descritor anteriormente criado;

mensagem : conteúdo da mensagem recebida;

qtBytes : quantidade de bytes recebidos;

```
msg = sockCliente.recv ( 1024 )
print msg
```

Prof. Luis Rodrigo
luisrodrigoog@gmail.com
http:// www.lncc.br/~lrodrigo

Exemplo :

servidorSimples.py

Prof. Luis Rodrigo
luisrodrigoog@gmail.com
http:// www.lncc.br/~lrodrigo

```
#!/usr/bin/python
# importando modulos
 import socket
 # definindo o endereço
 HOST = ''              # Endereco IP do Servidor
 PORT = 5000            # Porta que o Servidor esta
 # criando o descritor
 tcp = socket.socket(socket.AF_INET,
                     socket.SOCK_STREAM)
 orig = (HOST, PORT)
 # associando o processo a porta
 tcp.bind(orig)
 # escutando a porta
 tcp.listen(1)
```

Prof. Luis Rodrigo
luisrodrigoog@gmail.com
http:// www.lncc.br/~lrodrigo

```
# laço principal
while True:
    # recebe uma nova conexão
    con, cliente = tcp.accept()
    print 'Concetado por', cliente
    # laço para envio e recebimento de dados
    while True:
        msg = con.recv(1024)
        if not msg: break
        print cliente, msg
    print 'Finalizando conexao do cliente', cliente
    # encerra o socket
    con.close()
```

Prof. Luis Rodrigo
luisrodrigoog@gmail.com
http:// www.lncc.br/~lrodrigo

Exemplo :

servidorConcorrente.py

Prof. Luis Rodrigo
luisrodrigoog@gmail.com
http:// www.lncc.br/~lrodrigo

```
#!/usr/bin/python
# Importando os módulos
import socket
import os
import sys
# Definindo o endereço
HOST = ''        # Endereco IP do Servidor
PORT = 5000   # Porta que o Servidor esta
# Criando o socket
tcp = socket.socket(socket.AF_INET,
                                    socket.SOCK_STREAM)
```

Prof. Luis Rodrigo
luisrodrigoog@gmail.com
http:// www.lncc.br/~lrodrigo

```
...
# Associando a Porta do Processo
orig = (HOST, PORT)
tcp.bind(orig)
# Aguardando por conexões
tcp.listen(1)
#  Laço principal
while True:
        # aceitando conexões
        con, cliente = tcp.accept()
```

Prof. Luis Rodrigo
luisrodrigoog@gmail.com
http:// www.lncc.br/~lrodrigo

```
... ...
        #  gera um novo processo
        pid = os.fork()
        # verifica se é o cliente
        if pid == 0:
            # fecha o socket do proceso Pai
            tcp.close()
           # imprime dados do cliente
           print 'Conectado por, cliente
           # trata a nova conexão
          while True:
                  # recebe dados
                 msg = con.recv(1024)
                  # verifica se recebeu algo
                if not msg: break
                 # imprime dados
               print cliente, msg
```

Prof. Luis Rodrigo
luisrodrigoog@gmail.com
http:// www.lncc.br/~lrodrigo

```
        ...
        print 'Finalizando conexao do cliente', cliente
        # finaliza conexão
        con.close()
        # encerra o processo
        sys.exit(0)
else:
        # se for o pai, fecha o socket filho
        con.close()
```

Prof. Luis Rodrigo
luisrodrigoog@gmail.com
http:// www.lncc.br/~lrodrigo

Exemplo :

servidorThread.py

Prof. Luis Rodrigo
luisrodrigoog@gmail.com
http:// www.lncc.br/~lrodrigo

```
#!/usr/bin/python
# importando os modulos
import socket
import thread
# dados do endereço
HOST = ''
PORT = 5000
# funcao para tratar das conexões
def conectado(con, cliente):
    print 'Conectado por', cliente
    # laço para tratar a coexão
    while True:
        msg = con.recv(1024)
        if not msg: break
        print cliente, msg
```

Prof. Luis Rodrigo
luisrodrigoog@gmail.com
http:// www.lncc.br/~lrodrigo

```
    print 'Finalizando conexao do cliente', cliente
    con.close()
    thread.exit()

# corpo do programa
tcp = socket.socket(socket.AF_INET,
                                    socket.SOCK_STREAM)
# formando o endereço
orig = (HOST, PORT)
# associando o processo a porta
tcp.bind(orig)
```

Prof. Luis Rodrigo
luisrodrigoog@gmail.com
http:// www.lncc.br/~lrodrigo

```
# passa a escutar a porta
tcp.listen(1)

# laço principal do programa
while True:
    # aceita novas coneções
    con, cliente = tcp.accept()
    # cria uma nova thread para tratar a conexão
    thread.start_new_thread(conectado,
                            tuple([con, cliente]))

tcp.close()
```

Prof. Luis Rodrigo
luisrodrigoog@gmail.com
http:// www.lncc.br/~lrodrigo

Exemplo :

cliente.py

Prof. Luis Rodrigo
luisrodrigoog@gmail.com
http:// www.lncc.br/~lrodrigo

```
#!/usr/bin/python
# importa o modelo
import socket
# define o endereço do servidor
HOST = '127.0.0.1'
PORT = 5000
# cria o descritor
tcp = socket.socket(socket.AF_INET,
                    socket.SOCK_STREAM)
dest = (HOST, PORT)
# conecta ao servidor
tcp.connect(dest)
print 'Para sair use CTRL+X\n'
```

Prof. Luis Rodrigo
luisrodrigoog@gmail.com
http:// www.lncc.br/~lrodrigo

```
# obtem os dados a serem enviados
msg = raw_input()
# verifica se deve encerrar ou não
while msg <> '\x18':
    # envia os dados
    tcp.send (msg)
    # obtem novos dados
    msg = raw_input()
# fecha a conexão com o servidor
tcp.close()
```

Prof. Luis Rodrigo
luisrodrigoog@gmail.com
http:// www.lncc.br/~lrodrigo

Referências:

Sockets - Referencias:

Python:

[1] http://www.pythonbrasil.com.br/moin.cgi/CookBook

[2] http://docs.python.org/

[3] http://www.inf.ufrgs.br/~psgrigoletti/docs/
artigo_funcionalidades_python.pdf

[4] http://www-users.cs.york.ac.uk/~aw/pylinda/beginner.html

[5] http://www-users.cs.york.ac.uk/~aw/pylinda/beginner.html

[6] http://www.pyzine.com/

[7] http://aspn.activestate.com/ASPN/Cookbook/Python/

[8] http://pt.wikipedia.org/wiki/Python

[9] http://www.onlamp.com/python/

[10] http://py.vaults.ca/apyllo.py

Prof. Luis Rodrigo
luisrodrigoog@gmail.com
http:// www.lncc.br/~lrodrigo