Roteiro 4: Sockets – parte 1

Objetivos:

- *Criar e manipular sockets utilizando protocolo TCP;*
  - *Enviar e receber mensagens utilizando sockets TCP;*
  - *Controlar o fluxo de programas com o uso de sockets;*

Ferramentas necessárias

Python IDLE 2.7

Introdução

Neste roteiro serão criados programas para uso de sockets. O uso de sockets é importante para criar programas que possam comunicar-se com diferentes computadores para a troca de informações. Com o uso de sockets é possível fazer aplicações que respondam a requisições de outros computadores ou troquem informações entre si para manterem-se funcionando em uma rede.

Sockets consistem em objetos que fornecem um padrão para aplicações que trabalham em rede e se utilizam dos protocolos de rede (tcp, udp, ftp, etc). O python utiliza sockets através da classe socket, que deve ser importada em todos os programas que necessitem a utilizar obedecendo o formato:

```
import socket
```

Programas que utilizam sockets trabalham sempre em formato cliente-servidor, e para isto é sempre necessário criar o programa servidor, que escutará conexões em uma porta, e o programa cliente, que fará a conexão a algum socket disponível em alguma porta. Os principais comandos para uso de sockets são ilustrados na **Tabela 1***:*

Principais Comandos da classe socket em python	***Fonte: http://docs.python.org/library/socket.html***
Comando	***Função***
accept( )	*Aceita uma nova conexão e retorna os valores: o novo objeto socket e o endereço que o socket está comunicando.*
bind(hostname,port)	*Conecta o socket ao endereço da porta.*
close()	*Fecha o socket.*
connect(hostname,port)	*Conecta-se com outro socket, que pode ser externo ou local. Para conexões locais utiliza-se localhost ou 127.0.0.1*
getpeername()	*Retorna o endereço IP e a porta na qual o socket está conectado.*
getsocketnome()	*Retorna o endereço IP da porta do próprio socket.*
listen(max_connections)	*Inicia ouvindo a porta e fica esperando outras conexões. O sistema operacional recusa novas conexões quando ela atingir o valor máximo de conexões.*
send(string)	*Envia uma string de dados pelo socket.*

Tabela 1 - Principais Comandos em Python para manipular sockets

Para a criação de programas que utilizem sockets no ***lado do servidor*** *é necessário principalmente o uso dos seguintes métodos:*

```
objetoSocket = socket.socket(socket.AF_INET,socket.SOCK_STREAM) -> para definir o protocolo a ser utilizado.

objetoSocket.bind((host,porta)) -> iniciando a escuta de conexões no lado do servidor.

objetoSocket.listen(1) -> realizando o controle de conexões simultâneas que o servidor suportará.

objetoSocket.send(dados) -> para enviar dados.

objetoSocket.recv(1024) -> para receber dados controlando o tamanho do buffer.
```

*Para a criação de programas que utilizem sockets no **lado do cliente** é necessário principalmente o uso dos seguintes métodos:*

```
.socket(socket.AF_INET,socket.SOCK_STREAM) -> para definir o protocolo a ser utilizado.

.connect((servidor,porta)) -> para criar o objeto de conexão que irá conectar em um servidor.

.send(dados) -> para enviar dados.

.recv(1024) -> para receber dados controlando o tamanho do buffer.
```

Exemplos de programa utilizando sockets:

Programa Servidor: *Libera conexões para serem utilizadas (bind)*

```
import socket
HOST, PORT = 172.0.0.1, 20000

tcp_server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
tcp_server_socket.bind((HOST, PORT))
tcp_server_socket.listen(1)
conn, addr = tcp_server_socket.accept()
print 'Connected by', addr
while 1:
    data = conn.recv(1024)
    if not data: break
    print "\nReceived message '", data,"'"
conn.close()
```

Figura 1 - exemplo de programa servidor TCP

Programa Cliente: *Se conecta a um servidor que aguarda conexões*

```
import socket
HOST, PORT = 172.0.0.1, 20000

tcp_client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
tcp_client_socket.connect((HOST, PORT))
tcp_client_socket.send('Hello, world')
data = tcp_client_socket.recv(1024)
tcp_client_socket.close()
print 'Received', repr(data)
```

Figura 2 - Exemplo de programa cliente TCP

Tarefas

1. *Crie um programa servidor receba uma requisição na porta 5666 e imprima uma mensagem informando que uma requisição foi recebida + o nome do endereço de origem. Em seguida crie um programa cliente para realizar a requisição ao programa servidor;*
2. *Crie um programa servidor escute uma conexão na porta 5667. Deve ser enviado um usuário e uma senha para o programa servidor nas seguintes condições:*
   a. *No programa cliente, solicitar um usuário e uma senha e em seguida enviar para o programa servidor através do comando **objetoSocket.send(string)**.*
   b. *No programa servidor imprima o nome de usuário e a senha na tela e encerre o socket em seguida utilizando o comando **objetoSocket.recv(tamanhoBuffer)**.*
3. *Altere o programa do **passo 2** para que o servidor fique escutando permanentemente conexões na porta 5668. Teste fazendo com que o programa cliente seja executado mais de uma vez.*
4. *Altere o programa do **passo 3** para que cada par de usuário/senha enviados para o servidor sejam gravados em um arquivo de nome log-acesso.txt;*

5. *Altere o programa do **passo4** para que **não seja permitido** o envio de usuário ou senha vazio, caso isto aconteça o servidor deverá solicitar novamente usuário e senha ao programa cliente.*

Referências:

***"Introdução à programação com Python: algoritmos e lógica de programação para iniciantes"**, MENEZES, Nilo Ney Coutinho São Paulo Novatec 2010;*

***"Aprendendo Python"**, LUTZ, M.; ASCHER, D. 2.ed. Porto Alegre Bookman 2000.*

***"Conceitos básicos da linguagem Python",** disponível em http://web2pybrasil.appspot.com/init/plugin_wiki/page/curso-web2py-003;*

***"Material auxiliar para funções e strings"**, disponível em http://187.7.106.14/andre/redes/prog-internet-II/2012-2/livros-apostilas/roteiro2-material-auxiliar-funcoes-strings.pdf;*

***"Material auxiliar para manipulação de arquivos",** disponível em http://187.7.106.14/andre/redes/prog-internet-II/2012-2/roteiros/roteiro3-arquivos-import/arquivos-material-aux-andre-moraes.pdf.*