Centro de Ciências e Tecnologia

Ciência da Computação

UNIVERSIDADE FEDERAL DO CARIRI

TP11: Threads

Disciplina: Lab Redes (CC0044) Professora: Camila Oliveira

Objetivos

Neste laboratório, você vai desenvolver um servidor web básico que recebe requisições de de dois cliente diferente (navegador) e retorna um hello word como resposta.

O Protocolo de Transferência de Hipertextos (HTTP), trabalha como um protocolo de requisição-resposta entre um cliente e um servidor. Dois navegadores web em máquinas diferentes podem ser os clientes e uma aplicação em um computador que hospeda um site pode ser o servidor.

O funcionamento básico do protocolo é feito da seguinte forma: Um cliente (navegador) envia uma requisição HTTP para um servidor, solicitando algum arquivo desse servidor, como por exemplo o arquivo index.html. O servidor retorna uma resposta para o cliente, contendo informações sobre o status da requisição e, o arquivo solicitado. No entanto, para poder servir dois clientes ao mesmo tempo vamos precisar implementar o uso de threads.

Código thread

Abaixo você encontrará o código servidor com thread. Neste código, o serviço de http ainda não está implementado.

```
import socket
import thread
import sys

#Função para lidar com cada conexão (cliente)
def clientthread(conn):
    #Sai do loop quando o cliente desconecta, pois a variável data não
      #conterá nenhum conteúdo
```

```
    while True:

        #Receiving from client
        data = conn.recv(1024)
        reply = 'OK...' + data

        if not data:
            break

        conn.sendall(reply)

    #destroi o socket e encerra thread ao sair do loop
    conn.close()


host = ''
port = 1234

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

#garante que o socket será destruído (pode ser reusado) após uma interrupção
da execução
s.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)

#Bind socket to local host and port
try:
    s.bind((host, port))
except socket.error as msg:
    print 'Bind failed. Error Code : ' + str(msg[0]) + ' Message ' + msg[1]
    sys.exit()

s.listen(1)


#Continua recebendo conexões

while True:
    #Aceita conexões
    conn, addr = s.accept()

    print 'Connected with ' + addr[0] + ':' + str(addr[1])

    #Cria nova thread para uma nova conexão. O primeiro
    #argumento é o nome da função, e o segundo é uma tupla
    #com os parâmetros da função.
    start_new_thread(clientthread ,(conn,))
```

O código cliente não muda, você pode testar com o cliente TCP que você implementou nos lab anteriores.

Teste o código e confirme se de fato o servidor pode responder a vários cliente ao mesmo tempo! Mostre a professora o resultado.

Servidor web básico

Implemente um simples servidor que recebe requisições de um navegador e retornar um hello word como resposta. Para isso, você vai usar o mesmo código servidor com thread e fazer as modificações necessárias na função clientethread().

Execute seu código servidor e então abra um navegador e digite: http://localhost:1234/hello.

O que aparece no seu navegador?

E no lado do servidor?