COMPUTER NETWORKS

PYTHON-4 ASSIGNMENT

Complete Proxy Server Code:

from socket import \*

import sys

if len(sys.argv) <= 1:

print 'Usage : "python ProxyServer.py server\_ip"\n[server\_ip : It is the IP Address Of Proxy Server'

sys.exit(2)

# Create a server socket, bind it to a port and start listening

tcpSerSock = socket(AF\_INET, SOCK\_STREAM)

# Fill in start.

tcpSerSock.bind(('localhost', 12001))

tcpSerSock.listen(1)

# Fill in end.

while 1:

# Strat receiving data from the client

print 'Ready to serve...'

tcpCliSock, addr = tcpSerSock.accept()

print 'Received a connection from:', addr

message = tcpCliSock.recv(4096)

print message

# Extract the filename from the given message

print message.split()[1]

filename = message.split()[1].partition("/")[2]

print 'filename: %s' % (filename)

fileExist = False

filetouse = "/" + filename

print filetouse

try:

# Check wether the file exist in the cache

f = open(filetouse[1:], "r")

outputdata = f.read() #lines()

print 'filetouse: %s' % (filetouse)

fileExist = True

# ProxyServer finds a cache hit and generates a response message

tcpCliSock.send("HTTP/1.0 200 OK\r\n")

tcpCliSock.send("Content-Type:text/html\r\n")

# Fill in start.

tcpCliSock.send(outputdata)

# Fill in end.

print 'Read from cache'

# Error handling for file not found in cache

except IOError:

if fileExist == False:

# Create a socket on the proxyserver

c = socket(AF\_INET, SOCK\_STREAM) # Fill in start. # Fill in end.

hostn = filename.replace("www.","",1).partition("/")[0]

try:

# Connect to the socket to port 80

# Fill in start.

ip = gethostbyname(hostn)

print ip

c.connect((ip, 80))

# Fill in end.

# Create a temporary file on this socket and ask port 80

# for the file requested by the client

c.sendall("GET "+ "/ HTTP/1.1\r\nHost: " + hostn + "\r\n\r\n")

# Read the response into buffer

# Fill in start.

# Fill in end.

# Create a new file in the cache for the requested file.

# Also send the response in the buffer to client socket

# and the corresponding file in the cache

# Fill in start.

tmpFile = open("./" + filename,"wb")

BUFF\_SIZE = 1024

data = b''

while True:

part = c.recv(BUFF\_SIZE)

data += part

if len(part) <= 0:

break

tmpFile.write(data)

tmpFile.close()

# Fill in end.

except:

print "Illegal request"

sys.exit()

else:

print 'else'

# HTTP response message for file not found

# Fill in start.

tcpCliSock.send('HTTP/1.1 404 NOT FOUND\r\n')

tcpCliSock.send("Content-Type: text/html\r\n\r\n")

tcpCliSock.send('<html> <body> File not found </body> </html>')

# Fill in end.

# Close the client and the server sockets

tcpCliSock.close()

# Fill in start.

tcpSerSock.close()

# Fill in end.

Output:











