#Sam Lee

#12/2/18

#CPE 138

#Professor Panzica

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)

tcpSerPort = 2345

tcpSerSock.bind(('', tcpSerPort))

tcpSerSock.listen(5)

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(1024)

print(message)

# Extract the filename from the given message

print(message.split()[1])

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

print(filename)

fileExist = "false"

filetouse = "/" + filename

print(filetouse)

try:

# Check wether the file exist in the cache

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

outputdata = f.readlines()

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")

for i in range(0, len(outputdata)) :

tcpCliSock.send(outputdata[i])

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)

print(hostn)

try:

# Connect to the socket to port 80

c.connect(hostn, 80)

print 'Socket connected to port 80 of the host'

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

# file requested by the client

fileobj = c.makefile('r', 0)

fileobj.write("GET "+"http://" + filename + " HTTP/1.0\n\n")

# Read the response into buffer

buff = fileobj.readlines()

# 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

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

for i in range(0, len(buff)) :

tmpFile.write(buff[i])

tcpCliSock.send(buff[i])

except:

print("Illegal request")

else:

print '404 Error file not found.'

#Close the client and server sockets

tcpCliSock.close()

if \_\_name\_\_ == '\_\_main\_\_' :

main()