这个作业需要写一个小的可以cache网页的代理服务器。 这个代理服务器只要能够处理简单的get request就好 但是需要能够处理不同种类的HTML页面和图片。我会给出一部分基础代码来帮助你们开始 但是你需要对我给出的代码做出部分修改

一般来说当一个客户（浏览器）产生一个请求，请求会被送到web服务器那里。Web服务器会处理这个请求然后返回一个response信息给客户。但是有时候客户需要使用代理服务器来发送请求。客户可以使用代理服务器来发送请求。 Web服务器也可以使用代理服务器来返回response。如果代理服务器找不到请求的对象的缓存资源（cached copy），代理服务器把请求传给web服务器，然后web服务器会产生一个response信息并且把它传到代理服务器那里（这个过程会产生一个缓存（cached copy）。代理服务器然后发送请求的对象给客户。

有时候请求的对象在web服务器里是找不到的 而你的代理服务器要能够正确的处理这种情况。 一般来说 web 服务器会发送一个 404 NOT FOUND 给你的代理服务器 然而这种response本身是没有response body 的。 关于这部分我提供给你们的代码是会去尝试读取response body（这是不存在的） 然后程序会产生一个exception。 所以 关于这部分代码你一定要修改一下才能用。所以你的代理服务器需要避免这种exception 然后发送一个“not found” 回复给客户（这个回复是有response body的， response body 是 404 not found“

当以下格式的请求输入进你的浏览器的时候 你的代理服务器要能够去执行它

http：//servername：serverport/webserver/filename\*

具体例子

http：//localhost：8888/www.google.com

http：//localhost：8888/www.google.com/index.html

http：//192.168.1.1：8888/www.google.com

http：//192.168.1.1：8888/www.google.com/index.html

你的代理服务器同时也需要能够正确的处理不正确的输入格式

你可以写和测试你的代理服务在同一个电脑（使用localhost作为IP地址）， 但是你的代理服务器需要能够处理浏览器和代理服务器在不同电脑上运行的情况（一个电脑发送请求 另一个电脑当代理服务器）。这种情况下 需要使用那台作为代理服务器电脑的IP地址作为服务器名字

你的代理服务器要通过下面的命令格式打开

Python ProxyServer.py proxyserverIP proxyserverPort

Fill in start 和end 是需要补充的地方 然后一些地方还要根据上面的要求做修改

Skeleton Python Code for the Proxy Server

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.

# 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 = # Fill in start. # Fill in end.

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")

# Fill in start.

# 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 = # Fill in start. # Fill in end.

hostn = filename.replace("www.","",1)