

#<https://github.com/patrickPeake/web-proxy>

#Using only what we know from Module 2 slides 29-34

#The main difference between a proxy server and a web server is that for requests that cannot be met from the locally stored files, a web server

#responds "404 file not found" while when a proxy server is unable to fill a request from a client it acts as a client to a web server to

#fetch the requested file and then serve it from its cache/local storage.

#the non 404 responses such as 403, 400, 411 etc remain the same along with 200, while 304 serves the cached file and

#a proxy server will only respond 404 when the origin server it acts as a client to responds 404.

#A minimal proxy server would ensure that the requests are well formed (valid http get request) and check its cache for the requested resource. If it is found and still

#has time left to live then 304 is returned along with the resource. Otherwise it acts as a client to fetch the requested resource from the origin

#server and sends it to the original client once the resource is recieved.

#Decide the test procedure to show it's working. Are client-side

#changes needed? If yes then describe them and find alternative

#ways to test the server-side functionality

#In order to test the proxy server one can enter "<http://127.0.0.1:8080/test.html>" into the address bar of a web browser. if a web page loads

#then the proxy server worked as the only way for it to fetch a page initially is to fetch the page from the web server. The proxy server

can be observed working by using print statements as well

#Bonus- Does your server have an HOL problem? Module 2 slide 36

#If yes, make changes in your server to avoid it

#and explain what you have done.

#If no, explain why your server does not have this problem.

#The server is susceptible to HOL blocking this could be fixed by either breaking the packets up into smaller constituent parts and using

a round robin to send them out roughly at the same time, or by implementing multithreading so each time a request comes in a process on a

thread is created to handle it. If this were done the smaller objects could be transmitted concurrently with the large one as long as

resources and clock/transmit time are shared

#HOL- Head of Line. (Big) Packet at the front blocks packets at the back from

#going through, even if they could have been easily processed