Topics This Section

• MP7 overview
• Coding Examples
MP7

• Proxy Server with cache

• You won’t test your program using a console!
  • To test: Open up a browser (Firefox, Chrome, IE, it doesn’t matter), change the proxy settings, connect to your proxy
• Basic Proxy service

1. HTTP Request
   www.google.com

2. HTTP Request
   www.google.com

3. HTTP Response
   www.google.com

4. HTTP Response
   www.google.com
• Proxy should be able to handle multiple requests
  • Each request should be handled by a different thread (connection handler)
MP7

• Multiple recvs and sends may be necessary

1. HTTP Request
   www.google.com

2. HTTP Request
   www.google.com

3. HTTP Response
   www.google.com

4. HTTP Response
   www.google.com

recv

send

recv

send

Proxy

Host of
www.google.com
• Cache the response if the page is cache-able
  • Use cacheHTTPResponse in aux_http.h to determine if the page is cache-able
  • Use priqueue.h to build the cache

1. HTTP Request
   www.google.com

2. HTTP Request
   www.google.com

3. HTTP Response
   www.google.com

4. HTTP Response
   www.google.com
• Serve the request if page is in cache

1. HTTP Request
   www.google.com

recv

send

4. HTTP Response
   www.google.com

Page found in cache

Host of
www.google.com

Proxy cache
MP7

• When inserting response in cache, if the cache is full, remove the oldest page in the cache

• A page is recent if the page was either inserted or fetched recently

• To implement cache replacement policy, write match and comparer function appropriately
MP7

• Proxy terminates on CTRL + C (a SIGINT signal)

• Clean up memory before terminating the proxy
Coding

• ds7/1.c
  • Step 1: Setup the proxy of your browser
  • Step 2: read a request from your browser
  • Step 3: print the request on your console
  • Step 4: get response back from server and send back to browser (if we have time)

• Don’t forget to change the proxy setting of your browser back to the original one!