CO324 Lab 1: HTTP Clients

Outcomes

At the end of this lab you should be able to

- Write a basic HTTP client that can make requests and display response data and metadata.
- Encode and decode URLs and explain its necessity.
- Define logic that runs in a separate thread and create new threads.
- Recognize when a resource shared among threads requires synchronization.

References for Java HTTP Clients

- How to set the proxy for Java programs.
- Working with URLs
- URL encoding and decoding

Exercises on HTTP clients

1. Write a class GetURL that fetches a document at a URL specified on the command-line and prints it to standard output. Note what happens when the user provides a non-existent host or path as part of the URL.
2. Extend GetUrl to print the metadata HTTP headers sent by the web server.
3. Write a class Google that sends a string given on the command line to Google and prints the search results. For example, to search for “java http” GET the URL “https://www.google.lk/search?q=java+http”. Make sure you encode the URL.

References for Java threads

- https://docs.oracle.com/javase/tutorial/essential/concurrency/
- http://www.vogella.com/tutorials/JavaConcurrency/article.html

Exercises on concurrency

1. Write a class GetMultiURL to take multiple URLs as command-line arguments. The contents of each URL should be stored in an array of Strings. Each URL should be concurrently fetched in a separate thread. The main thread must wait until all downloads have completed before printing all the downloaded contents.
2. Modify GetMultiURL so that the data that each thread downloads is stored in a single shared StringBuffer.
3. What can go wrong when multiple threads use a shared buffer to write data?