Assignment 2

Computer Science 441

Due: 23:55, Friday October 20, 2017

Instructor: Majid Ghaderi

1 Objective

The objective of this assignment is to practice server side network programming with TCP. Specifically, you will implement a multi-threaded web server to serve web objects to HTTP clients over the Internet.

2 Overview

In this assignment, you will implement a simple web server from scratch. The server should support non-persistent HTTP. This means that once an HTTP request is served, the server closes the TCP connection. To inform the client that the connection is closed, include the header line "`connection:close`" in the server response.

2.1 Requirements

Your server is only required to handle GET requests. It has to check if the requested object is available, and if so return a copy of the object to the client. As this simplified version of a web server deals with GET only, your program needs to return responses with the following status codes only:

- `200 OK`
- `400 Bad Request`
- `404 Not Found`

Your web server should be able to handle more than one connection simultaneously. To handle multiple connections, the server has to be multi-threaded. In the main thread, the server listens on a fixed port. When it receives a TCP connection request, it sets up a TCP connection through another socket and services the request in a separate worker thread. That is, once the server accepts a connection, it will spawn a new thread to parse the incoming HTTP request, transmit the object, etc. Algorithms 1 and 2 provide a high-level description of the main and worker thread's functionality. 5

Algorithm 1 Main Thread.

1: **while** not stopped **do**
2: Listen for connection requests
3: Accept a new connection request from a client
4: Spawn a new worker thread to handle the new connection
5: **end while**

Algorithm 2 Worker Thread.

1: Parse the HTTP request
2: Ensure well-formed request (return error otherwise)
3: Determine if requested object exists (return error otherwise)
4: Transmit the content of the object over the existing connection
5: Close the connection

All the requested web pages are in the current directory where your web server is running, *i.e.*, the current directory is the web server root directory.

2.2 Header Lines

To be compliant with what most web browsers expect from a well-behaving web server, include the following header lines in your server response whenever the response is `200 OK`:

- Date
- Server
- Last-Modified
- Content-Length
- Content-Type
- Connection : close

The `Server` field, is the name of your web server. For response codes 400 and 404, include the following header lines:

- Date
- Server
- Connection : close

2.3 Code Specifications

Skeleton code for a Java class named `WebServer` is provided to you. You are asked to write code to complete the implementation of this class. There are three public methods in this class that you need to implement. A description of these methods follows:

- `WebServer(int port)`
  This is the default constructor. The parameter `port` specifies the port at which the web server listens.

- `void run()`
  Starts the web server in listening mode and implements Algorithm 1.
- `void shutdown()`
  Informs the web server to shut down, clean up and exit.

Your implementation should include exception handling code to deal with all checked exceptions in your program. Print exception messages to standard system output. You should be able to use a web browser or your `UrlCache` tester from Assignment 1 to test your web server implementation.

3 Restrictions

- You are not allowed to change the signature of the methods provided to you. You can however define other methods or classes in order to complete the program.
- You are not allowed to use class `URL` or `URLConnection` or their subclasses for this assignment. Ask the instructor if you are in doubt about any specific Java classes that you want to use in your program.