import java.io.\*; //import io

import java.net.\*; //import net

import java.util.\*;// import util

class sWorker extends Thread{ //class definition

Socket sock; // class member

sWorker (Socket s) {sock = s;} // constructor

public void run(){

try{

// get I/O stream in/out from the socket

BufferedReader in = new BufferedReader(new InputStreamReader(sock.getInputStream()));

PrintStream out = new PrintStream(sock.getOutputStream());

String browserReq = in.readLine ();

String namedata;

String type = "";

//to isolate out the request and the type

StringTokenizer tokenizer = new StringTokenizer(browserReq, " ");

String reqType = tokenizer.nextToken();

// check if the request start with right command or not

if(reqType.equals("GET")){

namedata = tokenizer.nextToken();

if(namedata.contains("..")) throw new RuntimeException();

}else{

namedata = null;

System.out.println("Need GET method");

}

if(!namedata.contains(".")){

//System.out.println("working ok1 ");

handleDir(namedata, out);

//System.out.println("working ok2 ");

}else{

//check the extension type or the context of the request

if(namedata.endsWith(".txt") || namedata.endsWith(".java")){

type = "text/plain";

handleHttp(namedata, type, out);

}else if (namedata.endsWith(".html") || namedata.endsWith(".htm") || namedata.endsWith("/")){

type = "text/html";

handleHttp(namedata, type, out);

}else if(namedata.contains("/cgi/addnums.fake-cgi")){

type = "text/html";

addnums(namedata, type, out);

}else{

type = "text/html";

handleHttp(namedata, type, out);

}

}

}catch(IOException ioe){System.out.println("Listen again...");}

}

private void handleHttp(String name, String type, PrintStream out) throws IOException{

if(name.startsWith("/") ) { // if the file name has a leading slash

name = name.substring(1); // get rid of the leading slash

}

InputStream in = new FileInputStream(name);// read the input file

byte[] buffer = new byte[10000]; //create a byte array that is big enough to store all the data

File f = new File(name); //read the input file and create a constant

//because of the substring before, have to add 2 more to be able to fit all the context in

int actlen = (int) (f.length() + 2);

//use a string builder to build the string which displays the output this assignment want

StringBuilder http = new StringBuilder();

http.append("HTTP/1.1 200 OK\n");

http.append("Content-Type: " + type + "\r\n");

http.append("Content-Length: " + actlen + "\r\n");

http.append("\r\n\r\n");

out.println(http.toString()); //send the http info out

System.out.println("Handling HTTP Response: " + http.toString()); // print a line in the terminal

out.write(buffer, 0, in.read(buffer)); //write the bytes

out.flush();

in.close();

}

private void handleDir(String dir, PrintStream out) throws IOException{

//initiate 2 variables

File f1;

String upper;

//to find out if the directory is the root or sub directory, if it is sub dir, create a parent path

if(!dir.equals("")){

f1 = new File("./" + dir);

dir = "/" + dir;

upper = dir.substring(0, dir.lastIndexOf("/"));

if (upper.equals("") ) upper = "/";

}else{

f1 = new File(".");

upper = "/";

}

File[] file = f1.listFiles(); //to show all the files in directory

//Use a string builder to store the header of the http request

StringBuilder http = new StringBuilder();

//use a string builder to build the string which displays the output this assignment want

StringBuilder dirs = new StringBuilder();

////append all the information to string body

dirs.append("<h1> Index of " + dir + "</h1>");

dirs.append("<pre>");

// show the back to parent directory hyper link

dirs.append("<a href='" + upper + "'>Parent Directory</a><br><br>");

for(int i = 0; i < file.length; i++){

if(file[i].isFile()){ // check if that's a file

//if it is a file, append the following information to string

dirs.append("File: <a href=" + file[i].toString().substring(1) + ">" + file[i].toString().substring(2) + "</a>\n");

//and print out it is a file on terminal

System.out.println("file: " + file[i]);

}else if(file[i].isDirectory()){ // check if the file is directory

//if it is a directory, append the following information to string

dirs.append("Directory: <a href=" + file[i].toString().substring(1) + ">" + file[i].toString().substring(2) + "</a>\n");

//and print out it is a directory on terminal

System.out.println("directory: " + file[i]);

}

}

dirs.append("</pre>"); //close the html

String dirResponse = dirs.toString();

//append enough length to the http header

http.append("HTTP/1.1 200 OK \r\n");

http.append("Content-Type: text/html \r\n");

http.append("Content-Length: " + dirResponse.getBytes().length + "\r\n\r\n\r\n");

//out print the header and body to the browser for displaying

out.println(http.toString());

out.println(dirResponse); //send the http info out

System.out.println("Handling Directory Response: " + dirResponse); // print a line in the terminal for error checking

}

//call this function to display the user name, and the sum of two inputs n0.

private void addnums(String name, String type, PrintStream out) throws UnsupportedEncodingException{

//Use a string builder to store the header of the http request

StringBuilder http = new StringBuilder();

//Use a string builder to store the body of the http request

StringBuilder body = new StringBuilder();

//use this data structure to store a series of parameters

Map<String, String> inputs = new HashMap<String, String>();

//split the url to the position after question mark

String newName = name.substring(name.indexOf("?") + 1);

//create an array to store string obj, which is splited by the symbol"&"

String objs[] = newName.split("&");

//for each obj, use decoder to decode

for (String obj: objs){

int i = obj.indexOf("=");

inputs.put(URLDecoder.decode(obj.substring(0, i), "UTF-8"), URLDecoder.decode(obj.substring(i + 1), "UTF-8"));

}

//calculate the sum of the two integers and store them into a int variable.

int sum = Integer.parseInt(inputs.get("num1")) + Integer.parseInt(inputs.get("num2"));

//append all the information to string body

body.append("Dear "+ inputs.get("person") + ", the sum of ");

body.append(inputs.get("num1") + " and " + inputs.get("num2") + " is " + sum + ". \r\n\r\n\r\n");

String bodyAsString = body.toString();

//append enough length to the http header

http.append("HTTP/1.1 200 OK \r\n");

http.append("Content-Type: "+ type +"\r\n");

http.append("Content-Length: " + bodyAsString.getBytes().length + "\r\n\r\n\r\n");

//out print the header and body to the browser for displaying

out.println(http.toString());

out.println(bodyAsString); //send the http info out

System.out.println("The CGI result is: " + bodyAsString); // print a line in the terminal for error checking

}

}

public class MyWebServer{

public static void main(String a[]) throws IOException{

// always 6, never need to change

int q\_len = 6;

// for this assignment, professor specifically required to use this port no.

int port = 2540;

//TCP/ip implementation

Socket sock;

ServerSocket servsock = new ServerSocket(port, q\_len);

System.out.println("Lanny's Port Listener listening at port 2540.\n");

while (true) {

sock = servsock.accept(); // wait for the next client connection to come in

new sWorker(sock).start(); // call my web server worker to handle the request from browser

}

}

}

>> java MyListener

Lanny's Port Listener listening at port 2540.

GET /dog.txt HTTP/1.1

Host: localhost:2540

Connection: keep-alive

User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_13\_2) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/63.0.3239.132 Safari/537.36

Upgrade-Insecure-Requests: 1

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,\*/\*;q=0.8

Accept-Encoding: gzip, deflate, br

Accept-Language: zh-CN,zh;q=0.9,en;q=0.8,ja;q=0.7,zh-TW;q=0.6

GET /cat.html HTTP/1.1

Host: localhost:2540

Connection: keep-alive

User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_13\_2) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/63.0.3239.132 Safari/537.36

Upgrade-Insecure-Requests: 1

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,\*/\*;q=0.8

Accept-Encoding: gzip, deflate, br

Accept-Language: zh-CN,zh;q=0.9,en;q=0.8,ja;q=0.7,zh-TW;q=0.6

========================================

>>java MyTelnetClient condor.depaul.edu

Lanny's MyTelnet Client, 1.0.

Using server: condor.depaul.edu, Port: 80

Enter text to send to the server, <stop> to end: GET /elliott/dog.txt HTTP/1.1

Enter text to send to the server, <stop> to end: Host: condor.depaul.edu:80

Enter text to send to the server, <stop> to end:

Enter text to send to the server, <stop> to end:

Enter text to send to the server, <stop> to end: stop

HTTP/1.1 200 OK

Date: Wed, 31 Jan 2018 17:27:31 GMT

Server: Apache/2.2.3 (Red Hat)

Last-Modified: Wed, 07 Oct 2015 20:29:55 GMT

ETag: "8a1bfc-30-521899bff76c0"

Accept-Ranges: bytes

Content-Length: 48

Content-Type: text/plain

Connection: close

This is Elliott's dog file on condor. Good job!

>>java MyWebServer

Lanny's Port Listener listening at port 2540.

//local host:2540/dog.txt

Handling HTTP Response: HTTP/1.1 200 OK

Content-Type: text/plain

Content-Length: 50

Listen again...

//local host:2540/cat.html

Handling HTTP Response: HTTP/1.1 200 OK

Content-Type: text/html

Content-Length: 338

Listen again...

// localhost:2540

file: ./MyTelnetClient.class

file: ./.DS\_Store

file: ./AddNum.html

directory: ./subA

file: ./dog.txt

file: ./MyWebServer.class

file: ./MyTelnetClient.java

file: ./sWorker.class

file: ./lWorker.class

file: ./cat.html

file: ./WebServerWorker.class

file: ./MyWebServer.java

file: ./MyListener.class

file: ./MyListener.java

file: ./http-streams.txt

Handling Directory Response: <h1> Index of //</h1><pre><a href='/'>Parent Directory</a><br><br>File: <a href=/MyTelnetClient.class>MyTelnetClient.class</a>

File: <a href=/.DS\_Store>.DS\_Store</a>

File: <a href=/AddNum.html>AddNum.html</a>

Directory: <a href=/subA>subA</a>

File: <a href=/dog.txt>dog.txt</a>

File: <a href=/MyWebServer.class>MyWebServer.class</a>

File: <a href=/MyTelnetClient.java>MyTelnetClient.java</a>

File: <a href=/sWorker.class>sWorker.class</a>

File: <a href=/lWorker.class>lWorker.class</a>

File: <a href=/cat.html>cat.html</a>

File: <a href=/WebServerWorker.class>WebServerWorker.class</a>

File: <a href=/MyWebServer.java>MyWebServer.java</a>

File: <a href=/MyListener.class>MyListener.class</a>

File: <a href=/MyListener.java>MyListener.java</a>

File: <a href=/http-streams.txt>http-streams.txt</a>

</pre>

Listen again...

// select the directory of subA

file: ./subA/cat.html

Handling Directory Response: <h1> Index of //subA</h1><pre><a href='/'>Parent Directory</a><br><br>File: <a href=/subA/cat.html>subA/cat.html</a>

</pre>

Listen again...

//select and go back to the parent directory

file: ./MyTelnetClient.class

file: ./.DS\_Store

file: ./AddNum.html

directory: ./subA

file: ./dog.txt

file: ./MyWebServer.class

file: ./MyTelnetClient.java

file: ./sWorker.class

file: ./lWorker.class

file: ./cat.html

file: ./WebServerWorker.class

file: ./MyWebServer.java

file: ./MyListener.class

file: ./MyListener.java

file: ./http-streams.txt

Handling Directory Response: <h1> Index of //</h1><pre><a href='/'>Parent Directory</a><br><br>File: <a href=/MyTelnetClient.class>MyTelnetClient.class</a>

File: <a href=/.DS\_Store>.DS\_Store</a>

File: <a href=/AddNum.html>AddNum.html</a>

Directory: <a href=/subA>subA</a>

File: <a href=/dog.txt>dog.txt</a>

File: <a href=/MyWebServer.class>MyWebServer.class</a>

File: <a href=/MyTelnetClient.java>MyTelnetClient.java</a>

File: <a href=/sWorker.class>sWorker.class</a>

File: <a href=/lWorker.class>lWorker.class</a>

File: <a href=/cat.html>cat.html</a>

File: <a href=/WebServerWorker.class>WebServerWorker.class</a>

File: <a href=/MyWebServer.java>MyWebServer.java</a>

File: <a href=/MyListener.class>MyListener.class</a>

File: <a href=/MyListener.java>MyListener.java</a>

File: <a href=/http-streams.txt>http-streams.txt</a>

</pre>

Listen again...

//select the addnum.html

Handling HTTP Response: HTTP/1.1 200 OK

Content-Type: text/html

Content-Length: 687

//type in my name and 2 numbers, show the correct message

Listen again...

The CGI result is: Dear Lanny, the sum of 4 and 6 is 10.

Listen again...

^Z

[1]+ Stopped java MyWebServer