(1) MyWebServer.java

import java.io.\*; //import everything in package java.io

import java.net.\*; //import everything in package java.net

import java.util.\*;

public class MyWebServer {

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

int q\_len=6; //the number of queues can get to wait in operating system

int port=2540; //the port number is 2540

Socket sock;

ServerSocket servsock=new ServerSocket(port, q\_len); //create a socket conneting with the port number

System.out.println("Jing Li's server starts to listen at port 2540.\n");

while (true){

sock=servsock.accept(); //socket is waiting to be connected with the client

new Worker(sock).start(); //use the Worker thread to do the work

}

}

}

/\*

\* The Worker thread would get http protocol message from the browser

\* and return information back to the browser accordingly

\* -the static file, which ends with .txt or .html, also works when the requst is asking for a directory

\* -send a dynamic html file back to the browser including file names on the same directory with the server

\* -would accept FORM input from the user and send relative messages back

\*/

class Worker extends Thread{

Socket sock;

Worker (Socket s){sock=s;}

public void run(){

PrintStream out=null;

BufferedReader in=null;

try{

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

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

String name; //the message coming from the socket

String[] nameList;

String fileName;

name=in.readLine(); //read the first line of the message coming from the browser

String fileType; //define the filetype

//If the message coming from the browser is starts with GET methos for the HTTP protocol, would do the following

//otherwise send back an error message.

if (name!=null&&!name.isEmpty()&&name.startsWith("GET")){

nameList=name.split(" "); //get a string array separated by " "

String addr=nameList[1]; //get the middle part which contains the path name

//If the browser is asking for the current directory which the server is working on, send a dynamic html page back to the browser

//which lists all the file name on the browser by hot link references to their contents

if (addr.equals("/")){

fileType="text/html"; //set up the type to be html file

out.print("HTTP/1.1 200 OK\r\n" +"Content-Type: " +fileType +"\r\n"+"Content-Length: " +"2000"+"\r\n"+"Date: " + new Date() + "\r\n\r\n" ); //send http header back to the browser

out.print("<html><head>\r\n</head>\r\n<body>\r\n"); //add head for the html

out.print("<h1>Index of my directory</h1>\r\n"); //add head of the body for the html

File dir=new File("./"); //create an abstract file using it's path

File[] listOfFiles=dir.listFiles(); //create an array of File which contains all the File or directory under the current directory

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

String linkName;

if ( listOfFiles[i].isDirectory ( ) ) { //if the file is a directory, get the path and add a slash after it

linkName=listOfFiles[i].getPath()+"/";

out.print ( "<a href= "+linkName+">"+listOfFiles[i].getPath().substring(2)+"</a><br>\r\n") ; //send back the name of the directory using hot link reference

}else if ( listOfFiles[i].isFile ( ) ){

linkName=listOfFiles[i].getPath(); //if it's a file, get the path

out.print ( "<a href="+linkName+">"+listOfFiles[i].getPath().substring(2)+"</a><br>\r\n") ; //send back the name of the file using hot link reference

}

}

out.print("</body></html>"); //end up the html file

//if the browser is asking for a directory, send a dynamic html page back to the browser

//which lists all the file name under that directory by hot link references to their contents

}else if(addr.endsWith("/")){

addr=addr.substring(1);

fileType="text/html"; //set up the type to be html file

out.print("HTTP/1.1 200 OK\r\n" +"Content-Type: " +fileType +"\r\n"+"Content-Length: " +"2000"+"\r\n"+"Date: " + new Date() + "\r\n\r\n" ); //send http header back to the browser

out.print("<html><head>\r\n</head>\r\n<body>\r\n"); //add head for the html

File dir=new File(addr); //create an abstract file using it's path

int len=addr.length(); //calculate the path's length

out.print("<h1>Index of my "+addr+" directory</h1>\r\n"); //add head of the body for the html

File[] listOfFiles=dir.listFiles(); //create an array of File which contains all the File or directory under the current directory

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

String linkName;

if ( listOfFiles[i].isDirectory ( ) ) {

linkName=listOfFiles[i].getPath().substring(len)+"/"; //if it is a directory, get the path, deleting the current name and add a slash after it

out.print ( "<a href= "+linkName+">"+listOfFiles[i].getPath().substring(len)+"</a><br>\r\n") ; //send back the name of the directory using hot link reference

}else if ( listOfFiles[i].isFile ( ) ){

linkName=listOfFiles[i].getPath().substring(len); //if it's a file, get the path , delete the current name from it

out.print ( "<a href="+linkName+">"+listOfFiles[i].getPath().substring(len)+"</a><br>\r\n") ; //send back the name of the file using hot link reference

}

}

out.print("</body></html>"); //end up the html file

//if the browser is asking for a txt or html file, get the content of the file,

//and send it back to the browser.

}else if (addr.endsWith("txt")||addr.endsWith("html")){

fileName=addr.substring(1); //extract "/" from the path name to get the file name

if (fileName.endsWith(".html")){

fileType="text/html"; //if the file is a html file, set up fileType to be html

}else{

fileType="text/plain"; //if the file is a txt file, set up fileType to be plain

}

//try to find the file with the same name as asked for,

//if can find such file in the current directory, send to to the browser,

//otherwise give an error message

try{

File f=new File(fileName); //create an abstract file using it's path

InputStream file=new FileInputStream(f); //create InputStream

out.print("HTTP/1.1 200 OK\r\n" +"Content-Type: " +fileType + "\r\n"+"Content-Length: " +f.length()+"\r\n"+"Date: " + new Date() + "\r\n\r\n" ); //send http header back to the browser

try {

byte[] buffer = new byte[1000]; //set the buffer size

while (file.available()>0)

out.write(buffer, 0, file.read(buffer)); //write the message to the browser from the buffer read

} catch (IOException e) { System.out.println(e); }

out.print("</body></html>"); //end up the html file

}catch(FileNotFoundException e){

out.println("File not found error!");

}

//if we get a FORM message from the browser, extract the message we want,

//and do some calculation, send the result back

}else if (addr.contains("fake-cgi")){

String[] parameters=addr.split("\\?")[1].split("&"); //try to extract the stream coming in by ? and &

String person=parameters[0].split("=")[1]; //get the information for person

int num1=Integer.parseInt(parameters[1].split("=")[1]); //get the num1

int num2=Integer.parseInt(parameters[2].split("=")[1]); //get the num2

int sum=num1+num2; //add num1 and num2, get the sum of them

fileType="text/html"; //set up the type to be html file

out.print("HTTP/1.1 200 OK\r\n" +"Content-Type: " +fileType +"\r\n"+"Content-Length: " +"100"+"\r\n"+"Date: " + new Date() + "\r\n\r\n" );//send http header back to the browser

out.print("<html><head>\r\n</head>\r\n<body>\r\n");

out.print("<h1>Dear "+person+", the sum of "+num1+" and "+ num2+ " is "+sum+". </h1><br>\r\n"); //send back the infromation we get after calculation

out.print("</body></html>"); //end up the html file

}else{

out.println("Request can't be resolved!");

}

}else{

out.println("Bad request, the browser sent a request this web server doesn't understand!");

}

sock.close();

}catch(IOException ioe){

System.out.println(ioe);

}

}

}

(2) http-streams

1)From MyListener:

LiJings-MacBook-Pro:MyWebServer lijing$ java MyListener

Jing Li's server starts to listen at port 2540.

GET /jane/dog.txt HTTP/1.1

Host: localhost:2540

User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.12; rv:51.0) Gecko/20100101 Firefox/51.0

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

Accept-Language: en-US,en;q=0.5

Accept-Encoding: gzip, deflate

Connection: keep-alive

Upgrade-Insecure-Requests: 1

GET /jane/cat.html HTTP/1.1

Host: localhost:2540

User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.12; rv:51.0) Gecko/20100101 Firefox/51.0

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

Accept-Language: en-US,en;q=0.5

Accept-Encoding: gzip, deflate

Connection: keep-alive

Upgrade-Insecure-Requests: 1

2)From MyTelnet:

LiJings-MacBook-Pro:MyWebServer lijing$ java MyTelnet condor.depaul.edu

Jing Li's MyTelnet, 1.8.

Using server: condor.depaul.edu, port 80

GET /elliott/dog.txt HTTP/1.1

Host: condor.depaul.edu:80

HTTP/1.1 400 Bad Request

Date: Sat, 04 Feb 2017 23:02:40 GMT

Server: Apache/2.2.3 (Red Hat)

Content-Length: 226

Connection: close

Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">

<html><head>

<title>400 Bad Request</title>

</head><body>

<h1>Bad Request</h1>

<p>Your browser sent a request that this server could not understand.<br />

</p>

</body></html>

GET /elliott/cat.html HTTP/1.1

Host: condor.depaul.edu:80

HTTP/1.1 400 Bad Request

Date: Sat, 04 Feb 2017 23:13:48 GMT

Server: Apache/2.2.3 (Red Hat)

Content-Length: 226

Connection: close

Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">

<html><head>

<title>400 Bad Request</title>

</head><body>

<h1>Bad Request</h1>

<p>Your browser sent a request that this server could not understand.<br />

</p>

</body></html>

//MyTelnet has some problems when trying to connect with the condor server. The message sent back from the web server is not what I expected, I checked the HTTP stream which I put into the console window, it should be the right one.

//So I tried to use the firefox network to get the response headers, and it gave me the correct and reasonable message which is as followed:

HTTP/1.1 200 OK

Accept-Ranges:"bytes"

Content-Length:"48"

Content-Type:"text/plain"

Date:"Sat, 04 Feb 2017 06:47:29 GMT"

Etag:""30c78-30-521899bff76c0""

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

Server:"Apache/2.2.3 (Red Hat)"

HTTP/1.1 200 OK

Accept-Ranges:"bytes"

Connection:"close"

Content-Length:"120"

Content-Type:"text/html"

Date:"Sat, 04 Feb 2017 23:11:24 GMT"

Etag:""2c806-78-475360c5dcbc0""

Last-Modified:"Mon, 05 Oct 2009 20:35:03 GMT"

Server:"Apache/2.2.3 (Red Hat)"

(3)serverlog

1) http:localhost:2540/dog.txt

The file dog.txt has been returned:

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

2) http://localhost:2540/lion.html

The lion.html file has been returned:

This is Elliott's

LION

html file

3) http://localhost:2540/sub-a/sub-b/lion.html

The lion.html file which is in directory ./a/b has been returned:

This is Elliott's

LION

html file

4)http:localhost:2540 AND http:localhost:2540/

a dynamic html page has shown on browser with hot link references:

Index of my directory

.DS\_Store

.git

.gitignore

addnums.html

dog.txt

http-streams.txt

lion.html

MyWebServer-Checklist.html

MyWebServer.class

MyWebServer.java

serverlog.txt

sub-a

Worker.class

5)http://localhost:2540/sub-a/

a dynamic html page has shown on browser with hot link references:

Index of my sub-a/ directory

.DS\_Store

sub-b

6) http://localhost:2540/sub-a/sub-b/

a dynamic html page has shown on browser with hot link references:

Index of my sub-a/sub-b/ directory

lion.html

7)http://localhost:2540/addnums.html

I entered my name and two numbers to be Jane, 34 and 65. After I submitted it, it returned to me a html file with the following content:

Dear Jane, the sum of 34 and 65 is 99.