import java.io.*;
import java.net.*;

public class Httpd implements Runnable {
    protected Socket client;

    public Httpd (Socket client) {
        this.client = client;
    }

    public void run () {
        try {
            InputStream in = client.getInputStream();
            HttpInputStream httpIn = new HttpInputStream(in);
            HttpProcessor processor = getProcessor(httpIn);
            OutputStream out = client.getOutputStream();
            HttpOutputStream httpOut = new HttpOutputStream(out, httpIn);
            processor.processRequest(httpOut);
            httpOut.flush();
        } catch (IOException ex) {
            ex.printStackTrace();
        } finally {
            try {
                client.close();
            } catch (IOException ignored) {
            }
        }
    }

    protected HttpProcessor getProcessor (HttpInputStream httpIn) {
        try {
            httpIn.readRequest();
            if (httpIn.getPath().startsWith(HTTP.CGI_BIN))
                return new HttpCGI(httpIn, client.getInetAddress());
            else if (httpIn.getPath().startsWith(HTTP.CLASS_BIN))
                return new HttpClass(httpIn);
            else
                return new HttpFile(httpIn);
        } catch (HttpException ex) {
            return ex;
        } catch (Exception ex) {
            StringWriter trace = new StringWriter();
            ex.printStackTrace(new PrintWriter(trace, true));
            return new HttpException(HTTP.STATUS_INTERNAL_ERROR,
                                       "<PRE>" + trace + "</PRE>");
        }
    }

    public static void main (String[] args) throws IOException {
        ServerSocket server = new ServerSocket(HTTP.PORT);
        while (true) {
            Socket client = server.accept();
            Httpd httpd = new Httpd(client);
            Thread thread = new Thread(httpd);
            thread.start();
        }
    }
}