

### 3

- 
- - 
  - 
  - URL
  - 
  -
- 

가 , C C++  
1 URL(Uniform Resource Locator) : URL AppletContext showDocument

1  
가 ,  
CGI(Common gateway Interface)  
가  
TCP/IP  
가

, 가

NT , (telnet) . ( 95  
가 . )  
 . \windows  
TELNET.EXE . Setup . )

1. .
2. time-A.timefreq.bldrdoc.gov .
3. 13 .(  
가 . )  
가 13-1

telnet time-A.timefreq.bldrdoc.gov 13

### 3-1 :

3-2 , .

50692 97-09-01 21:43:15 50 0 0 50.0 UTC(NIST) \*

### 3-2 : “Time of day”

“time of day”  
?  
 ,  
 . (  
 .) “time of day”  
13 “ ” .

---

: , 가  
.( 3-3 )

---

### 3-3 :

가 13 .

13  
(Listening)

time-A.timefreq.bldrdoc.gov 13 ,  
time-A.timefreq.bldrdoc.gov

132.163.135.130 . ( :  
time-A.timefreq.bldrdoc.gov

hosts

. ) 13

.( 95  
-> . )

1. java.sun.com 80 .  
2. .  
GET / HTTP/1.0  
3. Enter .

3-4 가 . -  
HTML .

---

가

### 3-4: Telnet HTTP

3-1

가

3-1 : SocketTest.java

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

public class SocketTest
{ public static void main(String[] args)
  { try
    { Socket s = new Socket("time-A.timefreq.bldrdoc.gov",13);

      BufferedReader in = new BufferedReader
        (new InputStreamReader(s.getInputStream()));
      boolean more = true;
      while (more)
      { String line = in.readLine();
        if (line == null) more = false;
        else
          System.out.println(line);
      }
    }
    catch (IOException e)
    { System.out.println("Error" + e);
    }
  }
}
```

가 가

가 java.net (import) . (

```

        . )
        /
        . (
        I/O
        가
        . )

```

---

, 가 ::

```

Socket s = new Socket("time-A.timefreq.bldrdoc.gov",13);
BufferedReader in = new BufferedReader
    (new InputStreamReader(s.getInputStream()));

```

```

        ,
        가
        .
        UnknownHostException
        가 , IOException
        UnknownHostException
        IOException
        (IOException)
        , java.net.Socket
        InputStream
        . ( 1 12
        )
        ,
        :

```

1. readLine
- 2.

---

```

        readLine
        null
        ,
        가
        java.net

```

가 가

```

,
, setSoTimeout
.)
.

```

```

Socket s = new Socket( . . );
s.setTimeout(10000); // 10

```

가

InterruptedException

```

try
{
    String line
    while((String line = in.readLine() ) != null)
    {
        process line
    }
}
catch(InterruptedException exception)
{
    react to timeout
}

```

가 . setSoTimeOut  
가 . Socket

가 . 가 .

SocketOpener.openSocket ”

```

Socket s = SocketOpener.openSocket( host , port , timeout);

```

run

```

public void run()
{
    try
    {
        socket = new Socket( host , port );
    }
    catch( IOException exception)
    {
    }
}

```

```

openSocket
    . Join
        가
        가
        join
        .(
        .)

```

```

t.start();
try
{
    t.join(timeout);
}
catch(InterruptedException exception)
{
}

```

3-2

SocketOpener

### 3-2 : SocketOpenerTest.java

```

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

public class SocketOpenerTest
{
    public static void main(String[] args)
    {
        String host;
        if (args.length > 0) host = args[0];
        else host = "www.yourcompany.com";

        int port;
    }
}

```

```

    if (args.length > 1) port = Integer.parseInt(args[1]);
    else port = 80;

    int timeout = 5000;
    Socket s = SocketOpener.openSocket(host, port, timeout);

    if (s == null)
        System.out.println("The socket could not be opened.");
    else
        System.out.println(s);
    }
}

```

```

class SocketOpener implements Runnable
{
    public static Socket openSocket(String aHost, int aPort,
        int timeout)
    {
        SocketOpener opener = new SocketOpener(aHost, aPort);
        Thread t = new Thread(opener);
        t.start();
        try
        {
            t.join(timeout);
        }
        catch (InterruptedException exception)
        {
        }
        return opener.getSocket();
    }
}

```

```

public SocketOpener(String aHost, int aPort)
{
    socket = null;
    host = aHost;
    port = aPort;
}

```

```

public void run()
{
    try

```



```

        { socket = new Socket(host, port);
        }
        catch (IOException exception)
        {
        }
    }

    public Socket getSocket()
    { return socket;
    }

    private String host;
    private int port;
    private Socket socket;
};

```

```

        .
        132.163.135.130      4      (      6      가      )

        InetAddress

        GetByName      InetAddress      ,

        InetAddress address
            = InetAddress.getByName("time-A.timefreq.bldrdoc.gov");

        4      132.163.135.130      InetAddress      .
getBytes

        byte[] addressBytes = address.getBytes();

        (      )
        . 가 ,      java.sun.com      3

        getAllByNama

```

호스트명	IP주소	호출된 메서드
localhost	127.0.0.1	getLocalHost

3-3

### 3-3 : InetAddressTest.java

```
public class InetAddressTest
{
    public static void main(String[] args)
    {
        try
        {
            if (args.length > 0)
            {
                String host = args[0];

                InetAddress[] addresses
                    = InetAddress.getAllByName(host);

                for (int i = 0; i < addresses.length; i++)
                    System.out.println(addresses[i]);
            }
            else
            {
                InetAddress localHostAddress
                    = InetAddress.getLocalHost();

                System.out.println(localHostAddress);
            }
        }
    }
}

catch (Exception e)
```

```

        { System.out.println("Error: " + e);
        }
    }
}

```

---

: TCP(Transmission Control Protocol) .  
 TCP . UDP(User Datagram Protocol) . TCP  
 . UDP  
 . UDP  
 가 .  
 UDP .  
 Java Network Programming , Elliotte Harold( O' Reilly , 1977 )

---

#### java.net.Socket

- Socket( String host , int port) throws UnknowHostException , IOException

: host  
 port

- void close()
- InputStream getInputStream
- OutputStream getOutputStream
- Void setSoTimeout(int timeout)

InterruptedException

: timeout (0 )

java.net.InetAddress

- static InetAddress getByNama(String host)
- static InetAddress[] getAllByName(String host)
- InetAddress
- static InetAddress getLocalHost()
- InetAddress
- byte[] getAddress()
- String getHostAddress()
- “132.163.135.130”
- String getHostName()

가  
8189

ServerSocket

8189

ServerSocket s = new ServerSocket(8189);

가

Socket incoming = s.accept();

가가

Socket

input reader    output writer

---

```

BufferedReader in = new BufferedReader
    (new InputStreamReader(incoming.getInputStream()));
PrintWriter out = new PrintWriter
    (incoming.getOutputStream(), true /* autoFlush */);

```

가

(reader)	(Writer)	(InputStream	BufferedReader
) readLine	(OutputStream	PrintWriter	) print
.	(binary)		DataInputStream
DataOutputStream		(Serialized Object)	
ObjectInputStream	ObjectOutputStream	.	

```

out.println("Hello! Enter BYE to exit.");

```

8189

```

String line = in.readLine();
if(line != null)
{
    out.println("Echo: " + line);
    if(line.trim().equals("BYE") done = true;
}
else done = true;

```

incoming

```

incoming.close();

```

---

http

∴

1.

2.

3.

3-4

3-4 : EchoServer.java

---

```
import java.io.*;
```

```
import java.net.*;
```

```
public class EchoServer
```

```
{ public static void main(String[] args )
```

```
{ try
```

```
{ ServerSocket s = new ServerSocket(8189);
```

```
Socket incoming = s.accept( );
```

```
BufferedReader in = new BufferedReader
```

```
(new InputStreamReader(incoming.getInputStream()));
```

```
PrintWriter out = new PrintWriter
```

```
(incoming.getOutputStream(), true /* autoFlush */);
```

```
out.println( "Hello! Enter BYE to exit." );
```

```
boolean done = false;
```

```
while (!done)
```

```
{ String line = in.readLine();
```

```
if (line == null) done = true;
```

```
else
```

```
{ out.println("Echo: " + line);
```

```
if (line.trim().equals("BYE"))
```

```

        done = true;
    }
}
incoming.close();
}
catch (Exception e)
{
    System.out.println(e);
}
}
}

```

```

: 127.0.0.1
: 8189

```

IP 127.0.0.1 (Local Loopback) 가

---

: -

. ( : TCP/IP .)

---

, 가 .

IP .

.

Hello! Enter BYE to exit.

BYE( ) .

### 3-5 :

가 가 .  
가 .

가 , accept

가 .  
..

```
while(true)
{
    Socket incoming = s.accept();
    Thread t = new ThreadedEchoHandler(incoming);
    t.start();
}
```

ThreadedEchoHandler Thread . run

class ThreadedEchoHandler extends Thread

```
{
    ...
    public void run()
    {
        try
        {
            BufferedReader in = new BufferedReader
                (new InputStreamReader(incoming.getInputStream()));
            PrintWriter out = new PrintWriter
                (incoming.getOutputStream(), true /* autoFlush */);
```



```

        String line;
        while ( ( line = in.readLine()) != null )
        { // process line
        }
        incoming.close();
    }
    catch (Exception e)
    { //
    }
}

```

( 3-5). 3-6

CTRL+C

**13-6 :**

### **3-5 : ThreadedEchoServer.java**

```

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

public class ThreadedEchoServer
{ public static void main(String[] args )
    { int i = 1;
      try
      { ServerSocket s = new ServerSocket(8189);

        for (;;)
        { Socket incoming = s.accept( );

```

```

        System.out.println("Spawning " + i);
        new ThreadedEchoHandler(incoming, i).start();
        i++;
    }
}
catch (Exception e)
{
    System.out.println(e);
}
}
}

class ThreadedEchoHandler extends Thread
{
    public ThreadedEchoHandler(Socket i, int c)
    {
        incoming = i; counter = c;
    }

    public void run()
    {
        try
        {
            BufferedReader in = new BufferedReader
                (new InputStreamReader(incoming.getInputStream()));
            PrintWriter out = new PrintWriter
                (incoming.getOutputStream(), true /* autoFlush */);

            out.println( "Hello! Enter BYE to exit." );

            boolean done = false;
            while (!done)
            {
                String str = in.readLine();
                if (str == null) done = true;
                else
                {
                    out.println("Echo (" + counter + "): " + str);

                    if (str.trim().equals("BYE"))
                        done = true;
                }
            }
            incoming.close();

```

```

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

```

```

private Socket incoming;
private int counter;
}

```

### java.net.ServerSocket

- ServerSocket( int port ) throws IOException

:  
 port

- Socket accept() throws IOException

. (            가

. )  
 Socket

- void close() throws IOException

:

SMTP            25  
 Transport Protocol)

. SMTP(Simple Mail  
 SMTP            가  
 SMTP

sendmail  
 Sendmail

, IP .

, ( SMTP )

.

.

1. .

```
Socket s = new Socket(" mail.yourserver.com", 25); // 25 is SMTP
```

```
PrintWriter out = new PrintWriter( s.getOutputStream());
```

2. print .

HELO

MAIL FROM:

RCPT TO :

DATA

( )

.

QUIT

SMTP

-

sender

. ( [president@whitehouse.gov](mailto:president@whitehouse.gov)

.

SMTP

.)

3-6 .

3-7

, , SMTP "Send"

가 .

SMTP

.

.

. (

1

.)

**3-7 : MailTest**

### 3-6: MailTest.java

```
import java.awt.*;
import java.awt.event.*;
import java.util.*;
import java.net.*;
import java.io.*;
import javax.swing.*;

public class MailTest
{
    public static void main(String[] args)
    {
        JFrame frame = new MailTestFrame();
        frame.show();
    }
}

class MailTestFrame extends JFrame
    implements ActionListener
{
    public MailTestFrame()
    {
        setTitle("MailTest");
        setSize(300, 300);
        addWindowListener(new WindowAdapter()
        {
            public void windowClosing(WindowEvent e)
            {
                System.exit(0);
            }
        });

        getContentPane().setLayout(new GridBagLayout());

        GridBagConstraints gbc = new GridBagConstraints();
        gbc.fill = GridBagConstraints.HORIZONTAL;
        gbc.weightx = 0;
        gbc.weighty = 0;

        gbc.weightx = 0;
        add(new JLabel("From:"), gbc, 0, 0, 1, 1);
    }
}
```

```

        gbc.weightx = 100;
        from = new JTextField(20);
        add(from, gbc, 1, 0, 1, 1);

        gbc.weightx = 0;
        add(new JLabel("To:"), gbc, 0, 1, 1, 1);
        gbc.weightx = 100;
        to = new JTextField(20);
        add(to, gbc, 1, 1, 1, 1);

        gbc.weightx = 0;
        add(new JLabel("SMTP server:"), gbc, 0, 2, 1, 1);
        gbc.weightx = 100;
        smtpServer = new JTextField(20);
        add(smtpServer, gbc, 1, 2, 1, 1);

        gbc.fill = GridBagConstraints.BOTH;
        gbc.weighty = 100;
        message = new JTextArea();
        add(new JScrollPane(message), gbc, 0, 3, 2, 1);

        response = new JTextArea();
        add(new JScrollPane(response), gbc, 0, 4, 2, 1);

        gbc.weighty = 0;
        JButton sendButton = new JButton("Send");
        sendButton.addActionListener(this);
        JPanel buttonPanel = new JPanel();
        buttonPanel.add(sendButton);
        add(buttonPanel, gbc, 0, 5, 2, 1);
    }

    private void add(Component c, GridBagConstraints gbc,
        int x, int y, int w, int h)
    {
        gbc.gridx = x;
        gbc.gridy = y;

```

```

        gbc.gridwidth = w;
        gbc.gridheight = h;
        getContentPane().add(c, gbc);
    }

```

```

public void actionPerformed(ActionEvent evt)
{
    SwingUtilities.invokeLater(new Runnable()
    {
        public void run()
        {
            sendMail();
        }
    });
}

```

```

public void sendMail()
{
    try
    {
        Socket s = new Socket(smtpServer.getText(), 25);

        out = new PrintWriter(s.getOutputStream());
        in = new BufferedReader(new
            InputStreamReader(s.getInputStream()));

        String hostName
            = InetAddress.getLocalHost().getHostName();

        send(null);
        send("HELO " + hostName);
        send("MAIL FROM: " + from.getText());
        send("RCPT TO: " + to.getText());
        send("DATA");
        out.println(message.getText());
        send(".");
        s.close();
    }
    catch (IOException exception)
    {
        response.append("Error: " + exception);
    }
}

```

```
}
```

```
public void send(String s) throws IOException
```

```
{ if (s != null)
    { response.append(s + "\n");
      out.println(s);
      out.flush();
    }
  String line;
  if ((line = in.readLine()) != null)
    response.append(line + "\n");
}
```

```
private BufferedReader in;
private PrintWriter out;
private JTextField from;
private JTextField to;
private JTextField smtpServer;
private JTextArea message;
private JTextArea response;
}
```

## URL

SMTP

가

API

API

:

```
Transport.send(message);
```



가

URL URLConnection  
가 URL

```
URL url = new URL(protocol:resource);
```

2 HTTP FTP

(resource) URL OpenStream  
InputStream

```
InputStream Uin = url.openStream();  
BufferedReader in = new BufferedReader(new InputStreamReader(uin));  
String line;  
While ( (line = in.readLine() )!= null)  
{  
    process line;  
}
```

가 URLConnection

URLConnection

1. URLConnection URL OpenConnection

:

```
URLConnection Connection = url.openConnection();
```

2.

setDoInput

setDoOutput

setIfModifiedSince

setUseCaches  
setAllowUserInteraction  
setRequestProperty..

API

3. connect

connection.connect();

4.

. getHeaderFieldKey

getHeaderField

가

getContentType

getContentLength

getContentEncoding

getData

getExpiration

getLastModified

5.

getInputStream

.( URL

openStream 가

.)

getObject

. text/plain image/gif

com.sun

가

가

. 가

setDoInput setDoOutput

( ,

)

connection.setDoOutput(true);

setIfModifiedSince

```

        . setUseCaches      setAllowUserInteraction
setUseCaches      . setAllowUserInteraction

.(      3-8)      가      .

```

### 3-8:

```

setReguestproperty      /

.

:

1.      ,      .
String input = username + “:” + password;
2.      64      .(      64
가      II      .)
String encoding = base64Encode(input);
3. “Authorization”      “Basic “ + encoding”      setReguestproperty

.

connection. setReguestproperty(“Authorization” , “Basic “ + encoding );

```

---

```

:      . FTP

URL
ftp://username:password@ftp.yourserver.com/pub/file.txt

sun.net.www.proto-
col.FtpURLConnection      가      rt.jar      javap

```

---

```

connect      . ,

```

```
        )
        String key = connection.getHeaderFieldKey(n);
        if (key != null) {
            String value = connection.getHeaderField(n);
        }
    }
}
```

HTTP

Date: Sun, 29 Aug 1999 00:15:48 GMT

Server:Apache/1.3.3(Unix)

Last-Modified: Thu , 24 Jun 1999 20:53:38 GMT

Accept-Range: bytes

Content-Length : 4813

Connection:close

Content-Type:text/html

, 6 가 가

. Long GMT 1970 1 1

3-1: connection

Date	getDate	long
Expires	getExpire	long
Last-Modified	getLastModified	long
Content-Length	getContentLength	int
Content-Type	getContentType	String
Content-Encoding	getContentEncoding	String

3-7 URL 가 ,

URL 가 ,

```
java URLConnectionTest http://www.yourserver.com user pw
```

- 
- 3-1 6
- 10

64

..

```
sun.misc.Base64Encoder
```

Base64Encoder

```
String encoding = new sun.misc.Base64Encoder().encode(input.getBytes());
```

```
sun com.sun
```

---

```
: javax.internet.MimeUtility
64
```

---

3-7 : URLConnectionTest.java

```
import java.io.*;
import java.net.*;
import java.util.*;

public class URLConnectionTest
{ public static void main(String[] args)
{ try
{ String urlName;
if (args.length > 0)
urlName = args[0];
else
urlName = "http://java.sun.com";

URL url = new URL(urlName);
URLConnection connection = url.openConnection();
```

```

// set username, password if specified on command line

if (args.length > 2)
{
    String username = args[1];
    String password = args[2];
    String input = username + ":" + password;
    String encoding = base64Encode(input);
    connection.setRequestProperty("Authorization",
        "Basic " + encoding);
}

connection.connect();

// print header fields

int n = 1;
String key;
while ((key = connection.getHeaderFieldKey(n)) != null)
{
    String value = connection.getHeaderField(n);
    System.out.println(key + ": " + value);
    n++;
}

// print convenience functions

System.out.println("-----");
System.out.println("getContentType: "
    + connection.getContentType());
System.out.println("getContentLength: "
    + connection.getContentLength());
System.out.println("getContentEncoding: "
    + connection.getContentEncoding());
System.out.println("getDate: "
    + connection.getDate());
System.out.println("getExpiration: "
    + connection.getExpiration());
System.out.println("getLastModified: "
    + connection.getLastModified());
System.out.println("-----");

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

// print first ten lines of contents

String line;
n = 1;
while ((line = in.readLine()) != null && n <= 10)
{
    System.out.println(line);
    n++;
}
if (line != null) System.out.println(". . .");
}
catch (IOException exception)
{
    System.out.println("Error: " + exception);
}

```

```

    }
}

public static String base64Encode(String s)
{
    ByteArrayOutputStream bOut
        = new ByteArrayOutputStream();
    Base64OutputStream out = new Base64OutputStream(bOut);
    try
    {
        out.write(s.getBytes());
        out.flush();
    }
    catch (IOException exception)
    {
    }
    return bOut.toString();
}
}

/* BASE64 encoding encodes 3 bytes into 4 characters.
|11111122|22223333|33444444|
Each set of 6 bits is encoded according to the
toBase64 map. If the number of input bytes is not
a multiple of 3, then the last group of 4 characters
is padded with one or two = signs. Each output line
is at most 76 characters.
*/

class Base64OutputStream extends FilterOutputStream
{
    public Base64OutputStream(OutputStream out)
    {
        super(out);
    }

    public void write(int c) throws IOException
    {
        inbuf[i] = c;
        i++;
        if (i == 3)
        {
            super.write(toBase64[(inbuf[0] & 0xFC) >> 2]);
            super.write(toBase64[((inbuf[0] & 0x03) << 4) |
                ((inbuf[1] & 0xF0) >> 4)]);
            super.write(toBase64[((inbuf[1] & 0x0F) << 2) |
                ((inbuf[2] & 0xC0) >> 6)]);
            super.write(toBase64[inbuf[2] & 0x3F]);
            col += 4;
            i = 0;
            if (col >= 76)
            {
                super.write('\n');
                col = 0;
            }
        }
    }

    public void flush() throws IOException
    {
        if (i == 1)
        {
            super.write(toBase64[(inbuf[0] & 0xFC) >> 2]);
            super.write(toBase64[(inbuf[0] & 0x03) << 4]);
            super.write('=');
            super.write('=');
        }
    }
}

```

```

        else if (i == 2)
        {
            super.write(toBase64[(inbuf[0] & 0xFC) >> 2]);
            super.write(toBase64[((inbuf[0] & 0x03) << 4) |
                ((inbuf[1] & 0xF0) >> 4)]);
            super.write(toBase64[(inbuf[1] & 0x0F) << 2]);
            super.write('=');
        }
    }

    private static char[] toBase64 =
    {
        'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H',
        'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P',
        'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X',
        'Y', 'Z', 'a', 'b', 'c', 'd', 'e', 'f',
        'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n',
        'o', 'p', 'q', 'r', 's', 't', 'u', 'v',
        'w', 'x', 'y', 'z', '0', '1', '2', '3',
        '4', '5', '6', '7', '8', '9', '+', '/'
    };

    private int col = 0;
    private int i = 0;
    private int[] inbuf = new int[3];
}

```

---

가 :  
 . https  
 SSL  
 . https URL  
 SSL  
 RSA RSA  
 SSL  
 가  
 RSA  
 . RSA  
 . ( RSA 2000 11 )  
 가  
 SSL  
<http://java.sun.com/products/plugin/1.2/docs/https.html>

---

#### java.net.URL

- InputStream openStream()
- URLConnection openConnection()  
 URLConnection

#### java.net.URLConnection

- void setDoInput(boolean doInput)  
 doInput true URLConnection



- void setDoOutput(boolean doOutput)  
doInput true URLConnection .
- void setIfModifiedSince(long time)  
URLConnection 가  
GMT 1970 1 1 .
- void setUseCaches(boolean useCaches)  
useCaches 가 true . URLConnection
- void setAllowUserInteraction(boolean allowUserInteraction)  
allowUserInteraction true  
URLConnection
- void setRequestProperty.(String key, String value)  
Key/value 가
- void connect()
- String getHeaderFieldKey(int n)  
N 가 n 0 ,  
null 가 .
- String getHeaderField(int n)  
N 가 n 0 ,  
null 가 .
- int getLength()  
가 -1 .
- String getContentType()  
text/plain image/gif .
- String getEncoding()  
gzip . identity  
Content-Encoding .
- long getData()
- long getExpiration()
- long getLastModified()

GMT 1970 1 1

- InputStream openInputStream()
- OutputStream openOutputStream()
- Object getObject()

가

text/plain image/gif

CGI

**CGI**

가

3-9

### 3-9 : HTML

가 Submit

CGI

(CGI Common gateway interface .) CGI

FORM

ACTION

CGI

CGI

cgi-bin

CGI

가 . CGI

HTML

3-10

HTML

(

)

### 3-10 : CGI 가

CGI Perl .

---

: CGI HTML  
. HTML 1997 “Definitive Guide 2<sup>nd</sup> edition ,  
Musciano Kennedy”가 . HTML CGI

---

CGI .  
가 .

Core Java Web Server , Chris Taylor and Tim Kinnear , Prentice-Hall 1998

가 CGI  
가 .

가 : GET POST .

GET , URL . URL

GET ?

, maps.yahoo.com  
py/maps.py 가 . addr csz  
& 가 .

```

        +
        .
        %
        16
        , Mastering C++
        Mastering+C%2b%2b
        16 2b ( 10 43 )
        “+”
        가
        .
        URL
        .
        , 1 Infinite Loop, Cupertino, CA
        URL
        .
        http://maps.yahoo.com/py/maps.py?addr=1+Infinite+Loop&csz=Cupertino+CA
GET
        GET
        가
        .
POST
        URL
        URLConnection
        name/value
        &
        .
        URLConnection
        .
        URL url = new URL(http://host/script);
        URLConnection connection = url.openConnection();
        ,
        setDoOutput
        .
        Connection.setDoOutput(true);
        ,
        getOutputStream
        .
        PrintWriter
        .
        .
        PrintWriter out
        = new PrintWriter(connection.getOutputStream() );

```

```

        out.print(nama1 + "=" + URLEncoder.encode(value1) + "&");
        out.print(nama2 + "=" + URLEncoder.encode(value1) + "\n");

        out.close();

```

```

BufferedReader in = new BufferedReader( new
        InputStreamReader(connection.getInputStream() ) );
String line;
while( (line=in.readLine())!=null)
{
    // process line
}

```

<http://www.census.gov/ipc/www/idbprint.html>  
 가 가 .( 3-11 ) HTML  
 HTML

```

<FORM METHID=POST ACTION="/cgi-bin/ipc/idbsprd">

```

```

        가 "Summit:
        POST 가
, 가 가
        NAME 가

```

```

<select name="tbl" size=8 >
<option value="001">001 Total Midyear Population
        more options....
</select>

```

tbl

001

Unites State US  
cty 가 . ( 가  
ISO-3166 .)

optyr 가 .  
latest checked .  
가 가 .

tbl=1&cty=CH&optyr=latest+checked

URL

<http://www.census.gov/cgi-bin/ipc/idbsprd:>

3-11 :

<PRE>

U.S. Bureau of the Census, International Data Base

Table 001. Total Midyear Population

| CtYear | Population |
|--------|------------|
|--------|------------|

|       |  |
|-------|--|
| China |  |
|-------|--|

|        |               |
|--------|---------------|
| CH2000 | 1,256,167,701 |
|--------|---------------|

Source: U.S. Bureau of the Census, International  
Data Base.

</PRE>

<PRE> </PRE>

HTML

가

—

. ( 가

HTML

.)

3-8

URL POST

URL

URL=http://www.census.gov/cgi-bin/ipc/idbsprd

tbl=001

cty=CH

optyr=latest checked

가

Properties props = new Properties();

FileInputStream in = new FileInputStream(fileName);

Props.load(in);

, URL

URL

URL url = new URL(props.getProperty("URL"));

props.remove("URL");

doPost

doOutput(true)

, Properties

name + "=" + URLEncoder.encode(value) + ch

/

ch ' & '

' \n '

```

        connection.getInputStream()    FileNotFoundException
                                   .( Error 404-page not found
        .)
                                   URLConnection
URLConnection                       getErrorStream
                                   .

        InputStream err = ( (URLConnection) connection).getErrorStream();

        , err

        zip.properties

        가
        가
        (
        HTML
        .)

        HTML

```

### 3-8: PostTest.java

```

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

public class PostTest
{
    public static void main(String[] args)
    {
        try
        {
            String fileName;
            if (args.length > 0)
                fileName = args[0];
            else
                fileName = "PostTest.properties";
            Properties props = new Properties();
            FileInputStream in = new FileInputStream(fileName);
            props.load(in);

            URL url = new URL(props.getProperty("URL"));
            props.remove("URL");
            String r = doPost(url, props);
        }
        catch (Exception e)
        {
            e.printStackTrace();
        }
    }
}

```



```

        System.out.println(r);
    }
    catch (IOException exception)
    { System.out.println("Error: " + exception);
    }
}

public static String doPost(URL url,
    Properties nameValuePairs) throws IOException
{ URLConnection connection = url.openConnection();
    connection.setDoOutput(true);

    PrintWriter out
        = new PrintWriter(connection.getOutputStream());

    Enumeration enum = nameValuePairs.keys();

    while (enum.hasMoreElements())
    { String name = (String)enum.nextElement();
        String value = nameValuePairs.getProperty(name);
        char ch;
        if (enum.hasMoreElements()) ch = '&'; else ch = '\n';
        out.print(name + "="
            + URLEncoder.encode(value) + ch);
    }

    out.close();

    BufferedReader in;
    try
    { in = new BufferedReader(new
        InputStreamReader(connection.getInputStream()));
    }
    catch (FileNotFoundException exception)
    { InputStream err
        = ((URLConnection)connection).getErrorStream();
        if (err == null) throw exception;
        in = new BufferedReader(new InputStreamReader(err));
    }
    StringBuffer response = new StringBuffer();
    String line;

    while ((line = in.readLine()) != null)
        response.append(line + "\n");

    in.close();
    return response.toString();
}
}

```

URLConnection

URLConnection

URLConnection

Content-type :

text/plain

text/html

application/octet-stream

application/x-www-form-urlencoded

Content-length: length

Content-type: 1024

URLConnection

( )

java.net.HttpURLConnection

- inputStream getErrorStream()

java.net.URLEncoder

- static String encode(String s)

s URL . URL 'A' - 'Z' , 'a' - 'z' , '0'  
- '9' , ' ' , '-' , '\_' , '.' '\*' '+'  
"%UV" 0xUV  
(low order)

java.net.URLDecoder

- static string decode(String s)

URL s .

가  
.: 가  
. 가 .:

가 .

가 . 가 .

- 가 .
- 가 .
- ,
- FAQ, , .

. 가

<http://iwin.nws.noaa.gov>

<http://iwin.nws.noaa.gov/iwin/ca/hourly.html>

. “Get Report” 가  
( 3-12 ). 3-9

가 getWeather

queryBase URL(<http://iwin.nws.noaa.gov/iwin/>)  
.html 가 :

```
String queryBase = getParameter("queryBase");
String query = queryBase + state + "/" + report + ".html";
```

```
URL url = new URL(query);
InputStream in = url.openStream();
```

```
URL url = new URL(query);
BufferedReader in = new BufferedReader( new InputStreamReader(url.openStream() ));
```

```
String line;
```

```
String line;
while( (line = in.readLine() ) != null )
{
    weather.append(removeTags(line) + "\n");
}
```

### 3-12: WeatherReport

#### 3-9: WeatherApplet.java

```
import java.net.*;
import java.io.*;
import java.util.*;
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
import javax.swing.*;

public class WeatherApplet extends JApplet
    implements ActionListener
{
    public void init()
    {
        Container contentPane = getContentPane();
        contentPane.setLayout(new BorderLayout());

        // Set up the lists of choices for states and reports
        JPanel listPanel = new JPanel();
        state = makeList(states, 6, listPanel);
        report = makeList(reports, 6, listPanel);
        contentPane.add(listPanel, "North");

        // Add the text area
        weather = new JTextArea(20, 80);
        weather.setFont(new Font("Courier", Font.PLAIN, 12));

        // Add the report button
        contentPane.add(new JScrollPane(weather), "Center");
    }
}
```

```

        JPanel buttonPanel = new JPanel();
        JButton reportButton = new JButton("Get report");
        reportButton.addActionListener(this);
        buttonPanel.add(reportButton);
        contentPane.add(buttonPanel, "South");
    }

    public JList makeList(final String[][] items, int visibleRows,
        Container parent)
    {
        JList list = new JList(new AbstractListModel()
        {
            public Object getElementAt(int i)
            {
                return items[i][0];
            }

            public int getSize()
            {
                return items.length;
            }
        });
        list.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);
        list.setVisibleRowCount(visibleRows);
        parent.add(new JScrollPane(list));
        return list;
    }

    public String getItem(JList list, String[][] items)
    {
        return items[list.getSelectedIndex()][1];
    }

    public void actionPerformed(ActionEvent evt)
    {
        weather.setText("");
        getWeather(getItem(state, states), getItem(report, reports));
    }

    // Put together the URL query and go get the data from it
    public void getWeather(String state, String report)
    {
        String r = new String();
        try
        {
            String queryBase = getParameter("queryBase");
            String query
                = queryBase + state + "/" + report + ".html";
            URL url = new URL(query);
            BufferedReader in = new BufferedReader(new
                InputStreamReader(url.openStream()));

            String line;
            while ((line = in.readLine()) != null)
                weather.append(removeTags(line) + "\n");
        }
        catch (IOException e)
        {
            showStatus("Error " + e);
        }
    }

    public static String removeTags(String s)
    {
        while (true)
        {
            int lb = s.indexOf('<');
            if (lb < 0) return s;
        }
    }

```

```

        int rb = s.indexOf('>', lb);
        if (rb < 0) return s;
        s = s.substring(0, lb) + " " + s.substring(rb + 1);
    }
}

```

```

private JTextArea weather;
private JList state;
private JList report;

```

```

private String[][] states =
{
    { "Alabama", "al" },
    { "Alaska", "ak" },
    { "Arizona", "az" },
    { "Arkansas", "ar" },
    { "California", "ca" },
    { "Colorado", "co" },
    { "Connecticut", "ct" },
    { "Delaware", "de" },
    { "Florida", "fl" },
    { "Georgia", "ga" },
    { "Hawaii", "hi" },
    { "Idaho", "id" },
    { "Illinois", "il" },
    { "Indiana", "in" },
    { "Iowa", "ia" },
    { "Kansas", "ks" },
    { "Kentucky", "ky" },
    { "Louisiana", "la" },
    { "Maine", "me" },
    { "Maryland", "md" },
    { "Massachusetts", "ma" },
    { "Michigan", "mi" },
    { "Minnesota", "mn" },
    { "Mississippi", "ms" },
    { "Missouri", "mo" },
    { "Montana", "mt" },
    { "Nebraska", "ne" },
    { "Nevada", "nv" },
    { "New Hampshire", "nh" },
    { "New Jersey", "nj" },
    { "New Mexico", "nm" },
    { "New York", "ny" },
    { "North Carolina", "nc" },
    { "North Dakota", "nd" },
    { "Ohio", "oh" },
    { "Oklahoma", "ok" },
    { "Oregon", "or" },
    { "Pennsylvania", "pa" },
    { "Rhode Island", "ri" },
    { "South Carolina", "sc" },
    { "South Dakota", "sd" },
    { "Tennessee", "tn" },
    { "Texas", "tx" },
    { "Utah", "ut" },
    { "Vermont", "vt" },
    { "Virginia", "va" },
    { "Washington", "wa" },
}

```

```

        { "West Virginia", "wy" },
        { "Wisconsin", "wi" },
        { "Wyoming", "wy" }
    };

    private String[][] reports =
    { { "Hourly (State Weather Roundup)", "hourly" },
      { "State Forecast", "state" },
      { "Zone Forecast", "zone" },
      { "Short Term (NOWCASTS)", "shortterm" },
      { "Forecast Discussion", "discussion" },
      { "Weather Summary", "summary" },
      { "Public Information", "public" },
      { "Climate Data", "climate" },
      { "Hydrological Data", "hydro" },
      { "Watches", "watches" },
      { "Special Weather Statements", "special" },
      { "Warnings and Advisories", "allwarnings" }
    };
}

```

“Get Report”

가

grant

```

{
    permission java.net.SocketPermission
        "iwin.nws.noaa.gov:80", "connect";
};

```

WeatherApplet.policy 9

80 HTTP iwin.nws.noaa.gov  
가

Appletviewer -J -Djava.security.policy=WeatherApplet.policy WeatherApplet.html

-J

-D

java.security.policy

.

.

:

●

.( 가 )

●

.

9

.

80

.

가

.

.

“

”

.

.

.

가

가

?

3-13

가

.

●

-

●

-

●

3

3-13 :

.

.

.

가

.

,



⇒  
( :

: <http://www.unitedmedia.com/comics/dilbert> )

가? 가?

(. (firewall)

가  
FTP ( anonymous FTP, FTP)

가  
FTP FTP

. ( 3-14 )

**3-14 :**

( Steven M. Bellovin ) 가 ( William K. Cheswick )  
Firewalls and Internet Security [Addison-Wesley, 1994]

가

가

가

가

(write)

?

가

URL

. URL

<http://www.rouge.com/cgi-bin/cgi-bin/cracker.pl?Garys+password+is+Sicily>

CGI

. CGI

. ( . ) CGI

.( (Convert Channel) . ) ( : CGI

.)

, 가 CGI  
가? . 80  
가 http  
HTML .: “  
.”  
URL .

<http://www.rogue.com/Garys/password/is/Sicily>

:

“ ”

?

. ( (Stock Ticker)  
) 가

가 .

..

HTML 가 :  
가 .

GET

가 .

`http://www.yourserver.com/proxysvr?URL=http://iwin.nws.noaa.gov/iwin/CA/hourly.html`

GET

.( 3-15 )

3-10

1998

Core

Java Web Server

3-10

가

가

GET

doGet

HttpServletRequest

URL

getParameter

HttpServletResponse

getWriter

가

PrintStream

URL

sendError

. C Perl

### 3-10 : ProxySvr.java

```
import java.io.*;
import java.net.*;
import javax.servlet.*;
import javax.servlet.http.*;
```

```

public class ProxySvr extends HttpServlet
{
    public void doGet(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException
    {
        String query = null;

        response.setContentType("text/html");
        PrintWriter out = response.getWriter();

        query = request.getParameter("URL");
        if (query == null)
        {
            response.sendError(HttpServletResponse.SC_BAD_REQUEST,
                "Missing URL parameter");
            return;
        }

        try
        {
            query = URLDecoder.decode(query);
        }
        catch (Exception exception)
        {
            response.sendError(HttpServletResponse.SC_BAD_REQUEST,
                "URL decode error " + exception);
            return;
        }

        try
        {
            URL url = new URL(query);
            BufferedReader in = new BufferedReader(new
                InputStreamReader(url.openStream()));

            String line;
            while ((line = in.readLine()) != null)
                out.println(line);
            out.flush();
        }
        catch (IOException exception)
        {
            response.sendError(HttpServletResponse.SC_NOT_FOUND,
                "Exception: " + exception);
        }
    }
}

```

**3-15:**

CGI

. CGI

C

Perl

가

http

가

Perl C

3-11

3-12

C

(

,

)

C

“Unix Networking Programming( Printice-hall,

1990)”

java.net

Perl

Perl

\$1

\$|

\$url = ~ s/%([a-fA-F0-9][a-fA-F0-9]) /pack(“C” , hex(\$1) ) /eg;

“

16

가

‘ % ’

16

”

. (

. )

Larry Wall

Richard L. Schwarz “programming Perl(O’ Reilly & Assoc., 1991)”

CGI

C

Perl

---

### 3-11 : proxysvr.c

```
#include <netdb.h>
#include <sys/types.h>
```

```

#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

#define MAXLINE 512
#define MAXNAME 128
#define HTTP 80

unsigned writen(fd, vptr, n)
int fd;
char* vptr;
unsigned n;
{ unsigned nleft;
  unsigned nwritten;
  char* ptr;

  ptr = (char*)vptr;
  nleft = n;
  while (nleft > 0)
  { if ((nwritten = write(fd, ptr, nleft)) <= 0)
    return nwritten;
    nleft -= nwritten;
    ptr += nwritten;
  }
  return n - nleft;
}

unsigned readline(fd, vptr, maxlen)
int fd;
char* vptr;
int maxlen;
{ unsigned n;
  unsigned rc;
  char* ptr;
  char c;

  ptr = vptr;
  for (n = 1; n < maxlen; n++)
  { if ((rc = read(fd, &c, 1)) == 1)
    { *ptr++ = c;
      if (c == '\n')
      { *ptr = 0;
        return n;
      }
    }
    else if (rc == 0)
    { if (n == 1) return 0;
      else
      { *ptr = 0;
        return n;
      }
    }
    else
    return -1;
  }
}

```

```

    *ptr = 0;
    return n;
}

void error(msg)
char* msg;
{
    fputs(msg, stderr);
    fputc('\n', stderr);
    exit(1);
}

void url_decode(in, out, outlen)
char* in;
char* out;
int outlen;
{
    int i = 0;
    int j = 0;
    while (in[i] != '\0' && j < outlen - 1)
    {
        if (in[i] == '+') out[j] = ' ';
        else if (in[i] == '%')
        {
            int ch;
            sscanf(in + i + 1, "%x", &ch);
            out[j] = ch;
            i += 2;
        }
        else out[j] = in[i];
        i++;
        j++;
    }
    out[j] = 0;
}

int main(argc, argv)
int argc;
char** argv;
{
    int sockfd;
    struct sockaddr_in serv_addr;
    int n;
    char* name;
    struct hostent* hostptr;
    char url[MAXLINE + 1];
    char sendline[MAXLINE + 1];
    char recvline[MAXLINE + 1];
    char server_name[MAXNAME];
    char file_name[MAXLINE];
    int port;
    int service = 0;
    char* p;
    char* q;

    url_decode(argv[1], url, sizeof(url));

    p = strstr(url, "URL=http://");
    if (p != url)
        error("Sorry--can only recognize URL=service://server/file");
    service = HTTP;
    p += strlen("URL=http://");
    q = strchr(p, '/');

```



```

if (q == NULL)
    error("Sorry--can only recognize //server/file");
strncpy(server_name, p, q - p);
server_name[q - p] = '\0';
strncpy(file_name, q, sizeof(file_name) - 1);
file_name[sizeof(file_name) - 1] = '\0';
port = service;

if ((sockfd = socket(PF_INET, SOCK_STREAM, 0)) < 0)
    error("Can't open stream socket");

bzero((char*)&serv_addr, sizeof(serv_addr));
serv_addr.sin_family = AF_INET;
hostptr = gethostbyname(server_name);
if (hostptr == 0) error("Can't find host");
name = inet_ntoa(*(struct in_addr*)*hostptr->h_addr_list);
serv_addr.sin_addr.s_addr = inet_addr(name);
serv_addr.sin_port = htons(port);

if (connect(sockfd, (struct sockaddr*)&serv_addr, sizeof(serv_addr)) < 0)
    error("Can't connect to server");

sendline[sizeof(sendline) - 1] = 0;
if (service == HTTP)
{
    strcpy(sendline, "GET ");
    strncat(sendline, file_name, sizeof(sendline) - 1
        - strlen(sendline));
}
strncat(sendline, "\r\n", sizeof(sendline) - 1
    - strlen(sendline));

n = strlen(sendline);
if (writen(sockfd, sendline, n) != n)
    error("Write error on socket");

fputs("Content-type: text/html\n\n", stdout);

do
{
    n = readline(sockfd, recvline, MAXLINE);
    if (n < 0)
        error("Read error on socket");
    else if (n > 0)
    {
        recvline[n] = 0;
        fputs(recvline, stdout);
    }
} while (n > 0);

return 0;
}

```

### 3-12: proxysvr.c

```

($url) = @ARGV;

$url =~ tr/+/ /;

```

```

$url =~ s/%([a-fA-F0-9][a-fA-F0-9])/pack("C", hex($1))/eg;

$pos = index($url, "URL=http://");

if ($pos != 0)
{ die "Sorry--can only recognize URL=http://server/file";
}

$port = 80;

$pos = 11;
$pos2 = index($url, "/", $pos);
if ($pos2 < 0)
{ die "Sorry--can only recognize //server/file";
}

$server_name = substr($url, $pos, $pos2 - $pos);
$file_name = substr($url, $pos2);
$AF_INET = 2;
$SOCK_STREAM = 1;

$sockaddr = 'S n a4 x8';

($name, $aliases, $proto) = getprotobyname ('tcp');
($name, $aliases, $type, $len, $thataddr)
    = gethostbyname($server_name);
$that = pack($sockaddr, $AF_INET, $port, $thataddr);

if (!socket (S, $AF_INET, $SOCK_STREAM, $proto))
{ die $!;
}

if (!connect (S, $that))
{ die $!;
}

select(S); $|=1; select(STDOUT);

$command = "GET ".$file_name;

print S $command."\r\n";

print "Content-type: text/html\n\n";
while (<S>)
{ print;
}

```

## WeatherReport

CGI

가

가

1. <http://java.sun.com/products/jsp/download.html>
2. `\jswdk`  
가  
(Path Seperator)
3. `8080 80 \jswdk\webserver.`  
port port NMToken "80"
4. `javac -classpath \jswdk\lib\servlet.jar;. proxySvr.java`
5. `ProxySvr.class \jswdk\webpages\WEB-INF\servlets`
6. `\jswdk\webpages\WEB-INF\mappings.properties` 가  
`Proxysvr.code=ProxySvr`
7. `\jswdk\webpages`  
`WeatherApplet.html`  
`WeatherApplet.class`  
`WeatherApplet$1.class`
8. `WeatherApplet.html PARAM`  
`<PARAM NAME="queryBase"`  
`VALUE=http://localhost/proxysvr?URL=http://iwin.nws.noaa.gov/iwin/>`
9. `\jswdk\startserver`
10. Appletviewer <http://localhost/WeatherApplet.html>  
URL  
2 -  
!  
가  
가  
가  
가

가

가

?

가

가

3 (

n )