

# 第6章 网络通信

2024年5月

## 6.1 Web视图

## 6.1.1 浏览器引擎WebKit

- WebKit是一个开源的浏览器引擎。Webkit内核具有非常好的网页解析机制，很多应用系统都使用WebKit作浏览器的内核。
- Android对 Webkit作了进一步的封装，并提供了丰富的API。Android平台的WebKit模块由Java层和WebKit库两个部分组成，Java层负责与Android应用程序进行通信，而WebKit类库负责实际的网页排版处理。

## WebKit包中的几个重要类

类 名	说 明
WebSettings	用于设置WebView的特征、属性等。
WebView	显示Web页面的视图对象，用于网页数据载入、显示等操作
WebViewClient	在Web视图中帮助处理各种通知、请求事件。
WebChromeClient	Google浏览器Chrome的基类，辅助WebView处理JavaScript对话框、网站的标题、网站的图标、加载进度条等。

## 6.1.2 Web视图对象

### 1、WebView类

- **WebView**类是**WebKit**模块**Java**层的视图类，所有需要使用**Web**浏览功能的**Android**应用程序都要创建该视图对象，用于显示和处理请求的网络资源。
- **WebKit**模块支持**HTTP**、**HTTPS**、**FTP**以及**javascript**请求。**WebView**作为应用程序的**UI**接口，为用户提供了一系列的网页浏览、用户交互接口，客户程序通过这些接口访问**WebKit**核心代码。

## WebView类的常用方法

方 法	说 明
WebView(Context context)	构造方法
loadUrl(String url)	加载URL网站页面
loadData(String data, String mimeType, String encod)	显示HTML格式的web视图
reload()	重新加载网页
getSettings()	获取 WebSettings对象
goBack()	返回上一页面
goForward()	向前一页面
clearHistory()	清除历史记录
addJavascriptInterface (Object obj, String interfaceName)	将对象绑定到Javascript, 允许从网页控制 Android程序, 从网页调用该方法。

## 2、使用WebView的说明

### (1) 设置WebView基本信息:

- 如果访问的页面中有Javascript, 则webview必须设置支持Javascript。

`webview.getSettings().setJavaScriptEnabled(true);`

- 触摸焦点起作用

`requestFocus();`

- 取消滚动条

`this.setScrollBarStyle(SCROLLBARS_OUTSIDE_OVERLAY);`

## (2) 设置WebView要显示的网页:

- 互联网用: `webView.loadUrl("http://www.google.com");`
- 本地文件用:  
`webView.loadUrl("file:///android_asset/XX.html");` 本地文件要存放在项目的assets目录中



(3) 用webview点击链接看了很多页面以后，如果不做任何处理，点击系统“**Back**”键，浏览器会调用finish()而结束自身的运行；如果希望浏览的网页回退而不是退出浏览器，需要在当前Activity中覆盖Activity类的onKeyDown(int keyCode,KeyEvent event)方法处理该Back事件。

# 【例6-1】应用WebView对象浏览网页。



```
public class WebViewActivity extends AppCompatActivity {  
    @Override    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_web_view);  
        //获得控件  
        WebView webView = (WebView) findViewById(R.id.wv_webview);  
        //访问网页  
        webView.loadUrl("http://www.baidu.com");  
        //系统默认会通过手机浏览器打开网页，为了能够直接通过WebView显示网页，则必须设置  
        webView.setWebViewClient(new WebViewClient(){  
            @Override  
            public boolean shouldOverrideUrlLoading(WebView view, String url) {  
                //使用WebView加载显示url  
                view.loadUrl(url);  
                //返回true  
                return true;  
            }  
        });  
    }  
}
```

<!-- 添加网络权限 -->

<uses-permission android:name="android.permission.INTERNET" />

# 6.1.3 调用JavaScript

## 几个辅助类

### (1) WebSettings类

方 法	说 明
<code>setAllowFileAccess(boolean flag)</code>	设置是否允许访问文件数据
<code>setJavaScriptEnabled(boolean flag)</code>	设置是否支持JavaScript脚本
<code>setBuiltInZoomControls(boolean flag)</code>	设置是否支持缩放
<code>setBlockNetworkImage (boolean flag)</code>	设置是否禁止显示图片， true为禁止显示
<code>setDefaultFontSize (int size)</code>	设置默认字体大小， 在1 ~ 72之间取值
<code>setTextZoom (int textZoom)</code>	设置页面文字缩放的百分比， 默认为100

## (2) WebViewClient类

**WebViewClient** 类用于对**WebView**对象中各种事件的处理，通过重写这些提供的事件方法，可以对**WebView**对象在页面载入、资源载入、页面访问错误等情况发生时进行各种操作。

方 法	说 明
onLoadResource(WebView view, String url)	通知webView加载url指定的资源时触发
onPageStarted(WebView view, String url, Bitmap favicon)	页面开始加载时触发
onPageFinished(WebView view, String url)	页面加载完毕时触发

### (3) WebChromeClient类

**WebChromeClient**是辅助**WebView**处理**JavaScript**对话框、网站的标题、网站的图标、加载进度条等操作的类。

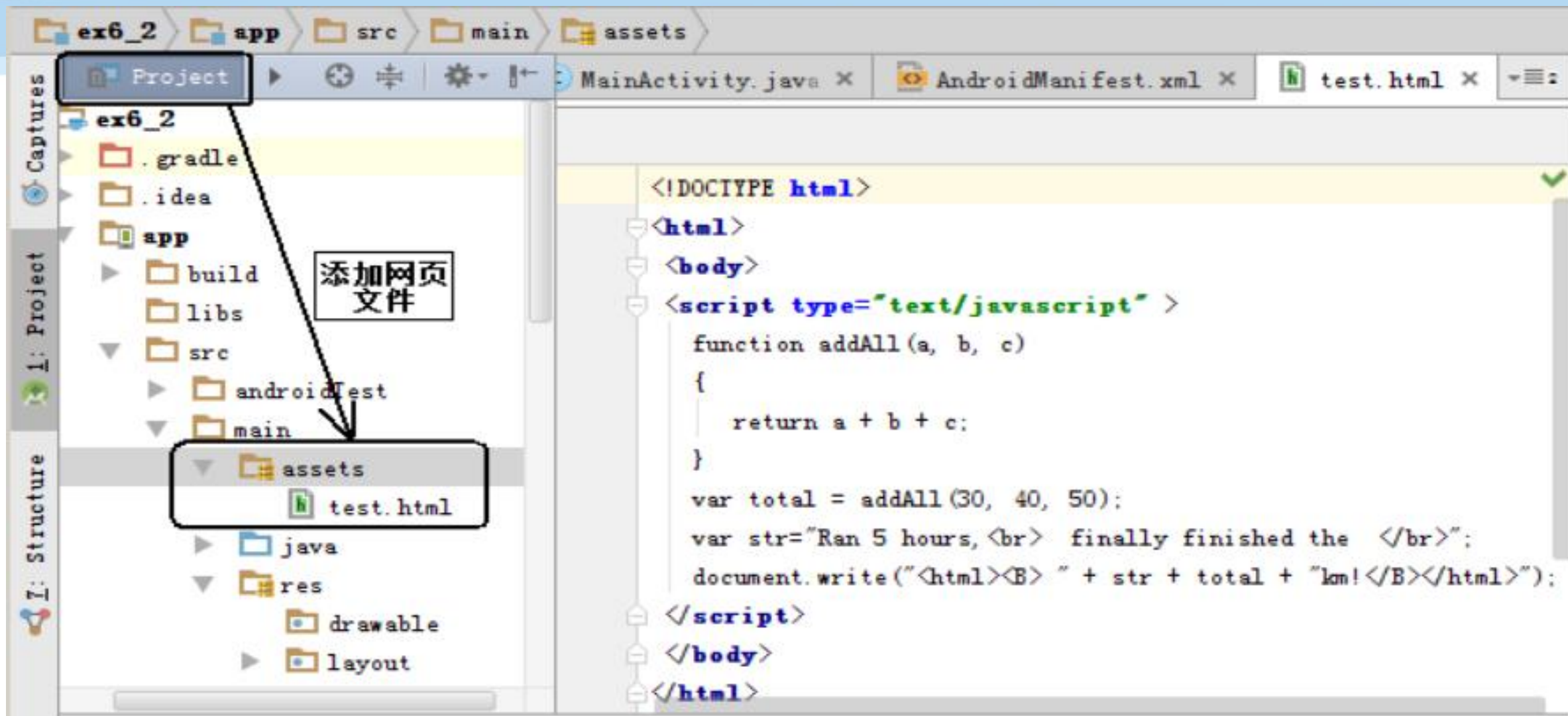
方 法	说 明
onJsAlert (WebView view, String url, String message, JsResult result)	处理JavaScript的alert对话框
onJsPrompt (WebView view, String url, String message, String defaultValue, JsPromptResult result)	处理JavaScript的Prompt提示对话框
onCloseWindow (WebView window)	关闭WebView

## 2. 调用本地HTML网页文件的JavaScript

可以在Android程序中调用本地的HTML网页文件的JavaScript, 如下面例子所示。

**【例6-2】** 在Android程序中调用本地的HTML程序示例。

(1) 在Android Studio编辑器中, 首先调整成“project”模式, 再在“main”目录下新建“assets”目录, 在“assets”目录下新建一个HTML程序test.html。





(2) 界面布局文件 (详见教材)

(3) 控制文件 (详见教材)



## 【例6-3】 Android程序操纵Javascript对话框。

- (1) 在项目的“assets”目录下，新建一个JavaScript对话框程序test1.html。
- (2) 在Android主控程序中操纵JavaScript对话框。

(代码详见教材)

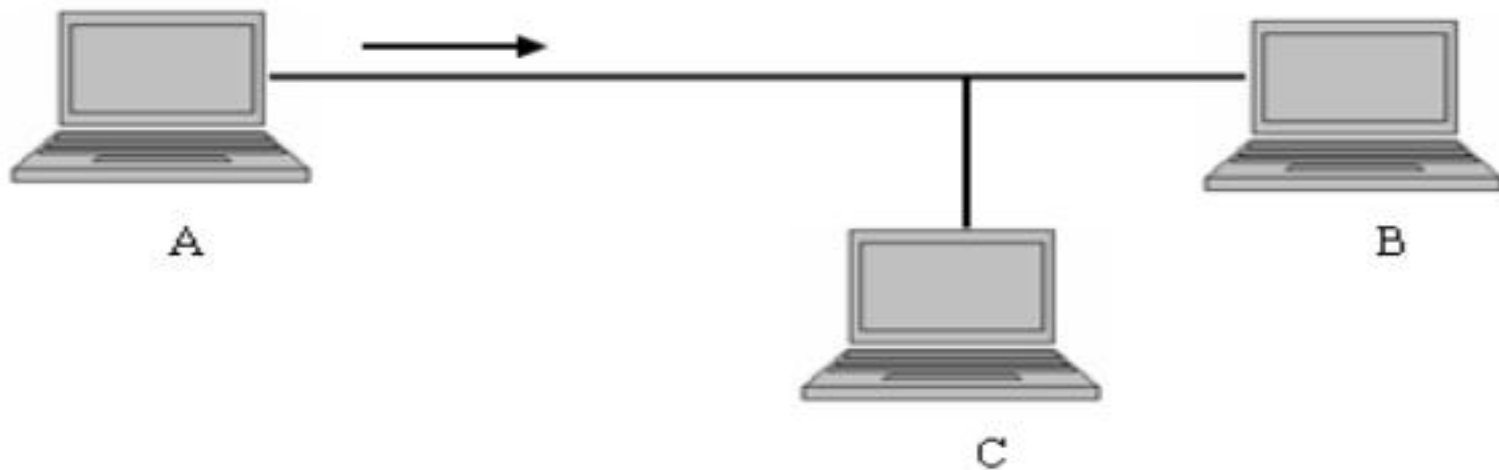


## 6.2 基于TCP协议网络程序设计

## 6.2.1 网络编程的基础知识

### 1、IP地址

网络中的每台计算机都必须有一个惟一的IP地址作为标识,这个数通常写作一组由“.”号分隔的十进制数。



在Java.net包中，IP地址由一个称作InetAddress的特殊类来描述。这个类提供了三个用来获得一个InetAddress类的实例的静态方法

- `getLocalHost()` 返回一个本地主机的IP地址。
- `getByName(String host)` 返回对应于指定主机的IP地址。
- `getAllByName(String host)` 对于某个主机有多个IP地址（多宿主主机）可用于得到一个IP地址数组。
- 此外，对一个InetAddress的实例可以使用
- `getAddress()` 获得一个用字节数组形式表示的IP地址。

# 【例6-4】通过域名查找IP地址。

核心语句:

```
23 class mClick implements OnClickListener
24 {
25     @Override
26     public void onClick(View arg0)
27     {
28         String str;
29         try{
30             InetAddress zsm_address=InetAddress.getByName("www.zsm8.com");
31             str="思维论坛的IP地址为: \n"+zsm_address.toString();
32         }
33         catch(UnknownHostException e)
34         {
35             str="无法找到思维论坛";
36         }
37         Toast.makeText(MainActivity.this, str, Toast.LENGTH_LONG).show();
38     }
39 }
```



## 2、端口

- 由于一台计算机上可同时运行多个网络程序，IP地址只能保证把数据信息送到该计算机，但无法知道要把这些数据交给该主机上的哪个网络程序，因此，我们用“端口号”来标识正在计算机上运行的进程（程序）。
- 每个被发送的网络数据包也都包含有“端口号”，用于将该数据帧交给具有相同端口号的应用程序来处理。

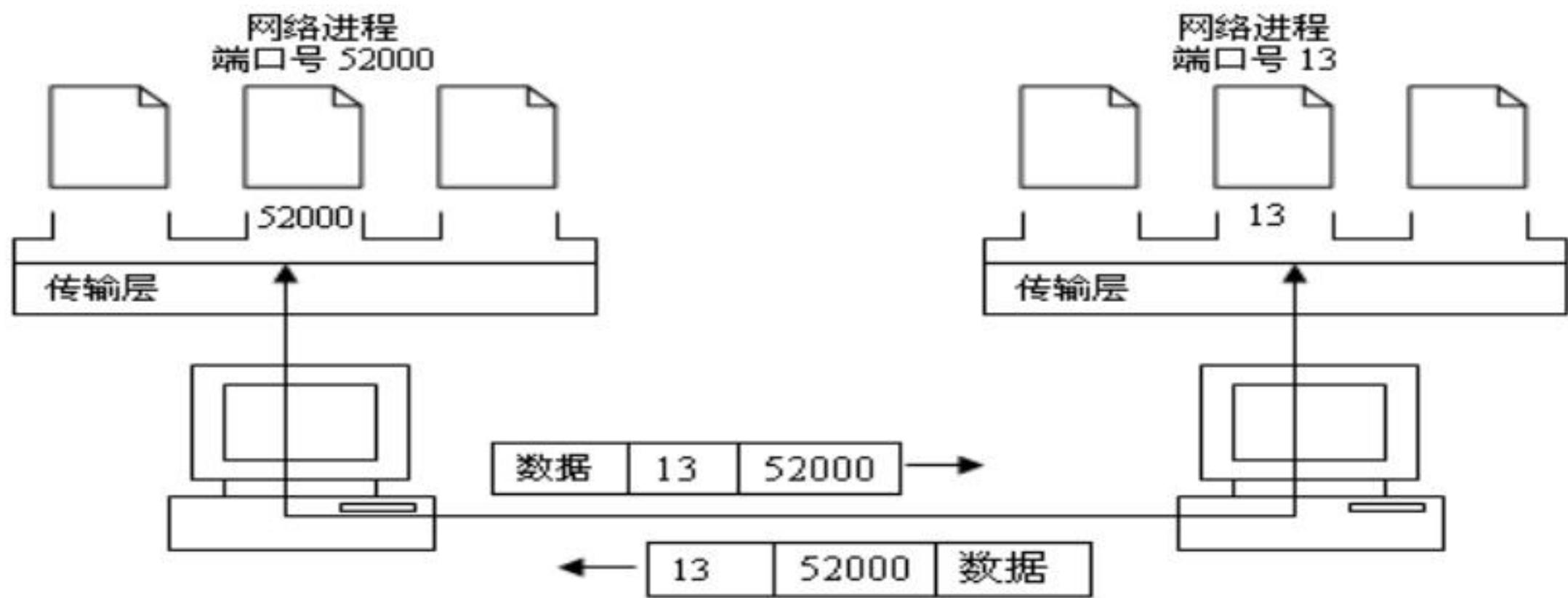


图6.3 用“端口号”来标识进程



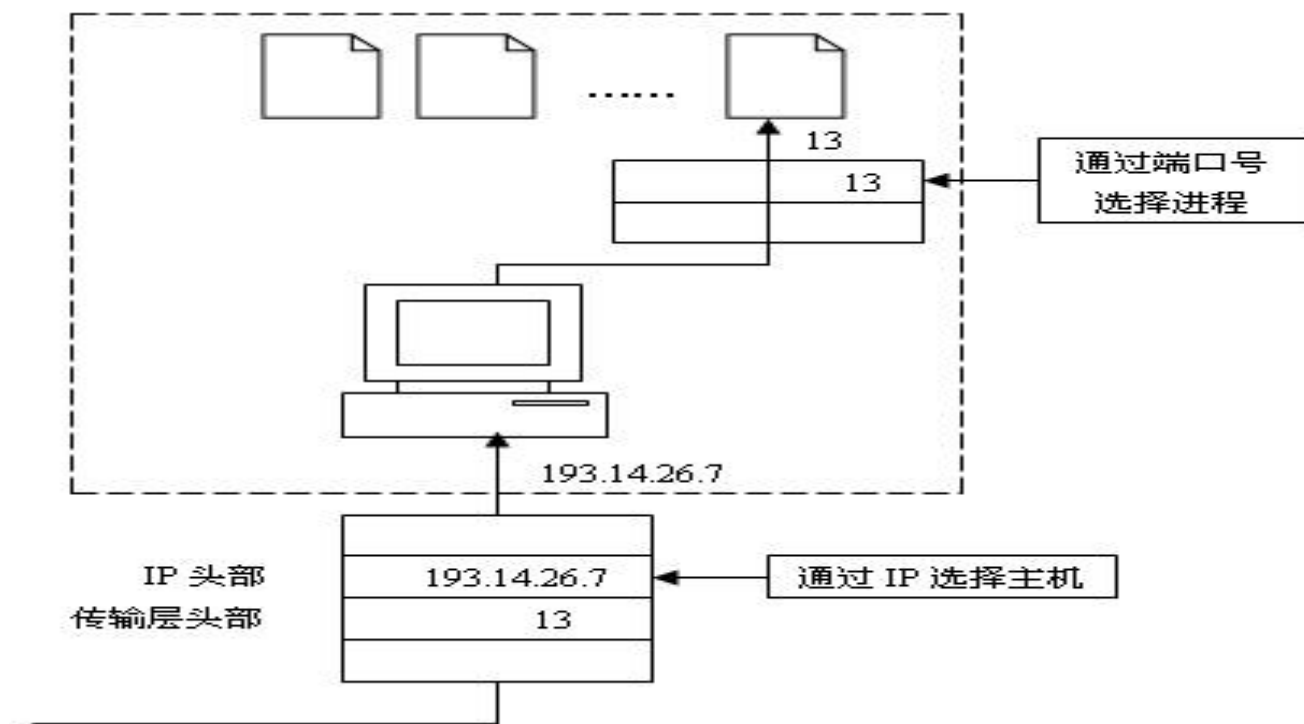
### 3、TCP与UDP协议

- TCP是面向连接的通信协议，TCP提供两台计算机之间的可靠无差错的数据传输。
- UDP是无连接通信协议，UDP不保证可靠数据的传输。

## 4. 套接字

什么是套接字？

在TCP/IP通信协议中，套接字（Socket）就是IP地址与端口号的组合。



## 5、客户端建立套接字Socket对象

- 在客户端使用`socket`类，建立向指定服务器IP和端口号连接的套接字,其构造方法是:
- `Socket(host_IP, prot);`
- 其中`host_IP`是服务器的IP地址，`prot`是一个端口号。
- 由于建立`Socket`对象可能发生`IOException`异常，因此，在建立`Socket`对象时要使用`try-catch`结构处理异常事件。

- Socket主要方法有:
- `getInputStream()`: 获得一个输入流, 读取从网络线路上传送来的数据信息。
- `getOutputStream()`: 获得一个输出流, 用这个输出流将数据信息写入到网络“线路”上。

## 6、服务器端建立套接字Socket对象

- 编写TCP网络服务器程序时，要用ServerSocket类用以创建服务器Socket，ServerSocket类的构造方法为：

ServerSocket(int port);

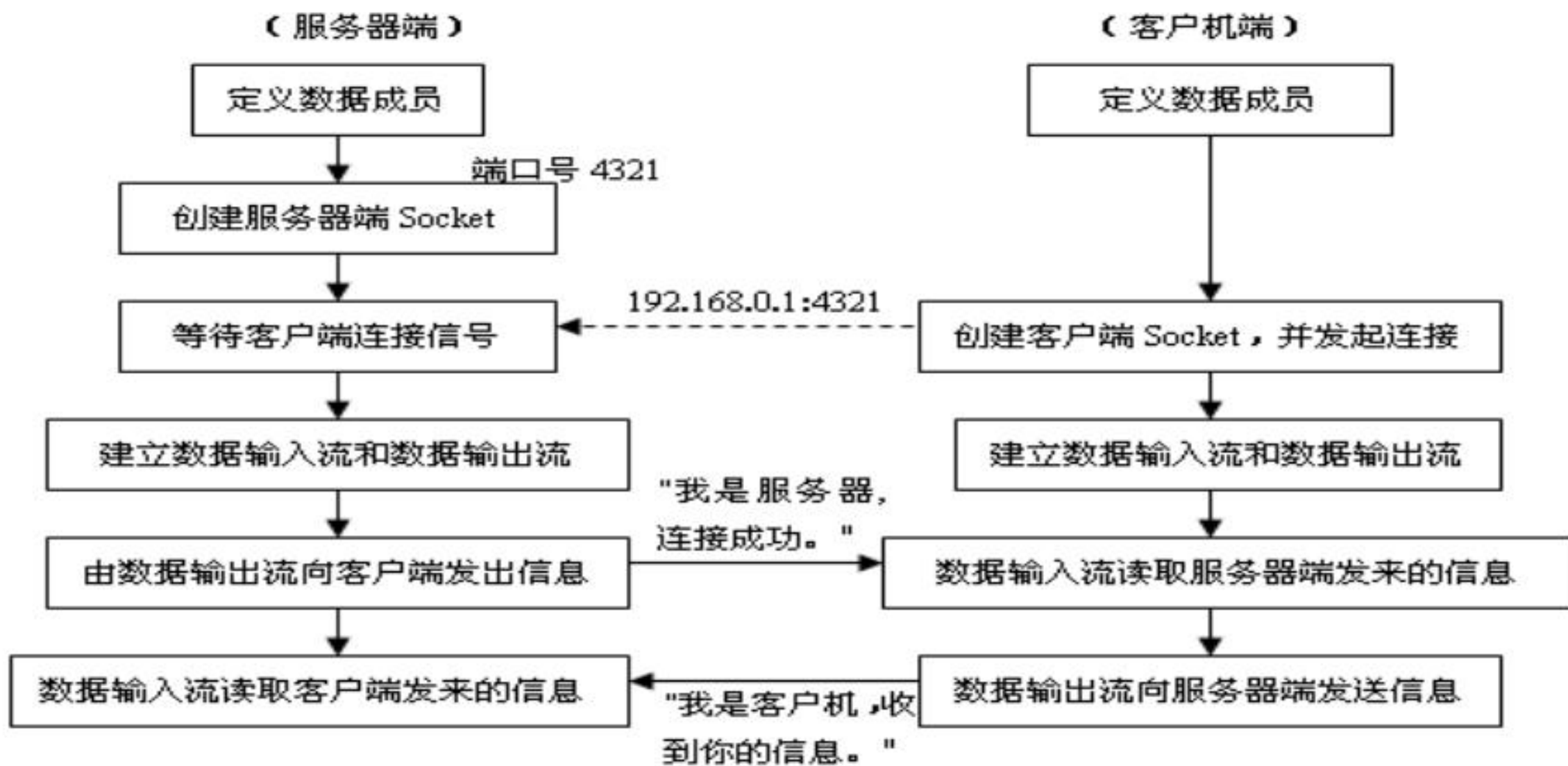
- 创建ServerSocket实例是不需要指定IP地址的，ServerSocket总是处于监听本机端口的状态。
- ServerSocket类的主要方法：

Socket accept();

该方法用于在服务器端的指定端口监听客户机发起的连接请求，并与之连接,其返回值为Socket对象。

## 6.2.2 基于TCP协议网络程序设计

- 利用套接字Socket设计客户机/服务器系统程序，进行数据通信与传输，大致有如下步骤：
  - (1) 创建服务器端ServerSocket，设置建立连接的端口号。
  - (2) 创建客户端Socket对象，设置绑定的主机名称或IP地址，指定连接端口号。
  - (3) 客户机Socket发起连接请求。
  - (4) 建立连接。
  - (5) 取得InputStream和OutputStream。
  - (6) 利用InputStream和OutputStream进行数据传输。
  - (7) 关闭Socket和ServerSocket。



客户机/服务器模式

## 【例6-5】远程数据通信示例，本例由客户端程序和服务端程序两部分组成。

在配置文件AndroidManifest.xml中加入允许访问网络的权限语句：

```
<uses-permission android:name="android.permission.INTERNET" />
```



(a) 服务器端运行结果



(b) 客户端运行结果



## 6.2.3应用Callable接口实现多线程socket编程

### 1. Callable接口

在Java语言中，经常使用的Thread类在run()方法执行完之后是没有返回值的，要实现子线程完成任务后返回值给主线程需要借助第三方转存。Callable接口则提供了一种有返回值的多线程实现方法。

Callable接口的定义如下：

```
public interface Callable<V>
{
    V call() throws Exception;
}
```

## 2. 线程接口Runnable和Callable的区别

**Callable**接口是与Java的**Runnable**接口的作用很类似，但它们之间又有很多不同，其区别如下：

- (1) **Runnable**接口是自从java1.1就有了，而**Callable**接口是1.5之后才新增加的；
- (2) **Callable**接口中定义的方法是**call()**,**Runnable**中定义的方法是**run()**；
- (3) **Callable**接口的任务执行后可以有返回值，而**Runnable**接口的任务是不能有返回值的(其返回类型为**void**)；
- (4) **call()**方法可以抛出异常，**run()**方法不能抛出异常；
- (5) 运行**Callable**接口任务可以返回一个**Future**对象，该对象为异步计算的结果。它提供了检查计算是否完成的方法，以等待计算的完成，并检索计算的结果。通过**Future**对象可以了解任务执行情况，可取消任务的执行，还可获取执行结果。

### 3. Callable接口实现多线程socket编程示例

**【例6-6】** 应用Callable接口实现多线程远程数据通信示例。

Android客户端程序由实现callable接口的connSocket.java和主程序MainActivity.java两部分组成。

## 6.3 基于HTTP协议网络程序设计

## 6.3.1 建立PHP服务器网站



手机获取网络服务器的数据示意图

## 6.3.2 应用HttpURLConnection访问Web服务器

### 1. HttpURLConnection类

HttpURLConnection是java的标准类，由于没有进行封装，需要进行比较复杂的设置，用起来不太方便。

### 2. StrictMode类

StrictMode通常用于捕获磁盘访问或者网络访问中与主进程之间交互产生的问题，因为在主进程中，UI操作和一些动作的执行是最经常用到的，它们之间会产生一定的冲突问题。将磁盘访问和网络访问从主线程中剥离可以使磁盘或者网络的访问更加流畅，提升响应度和用户体验。

在 `HttpURLConnection` 使用前，需要调用 `HttpURLConnection` 的两个方法：

- `StrictMode.setThreadPolicy()`：线程对象管理策略
- `StrictMode.setVmPolicy()`：`StrictMode`虚拟机对象管理策略

`StrictMode` 方法需要在主页面的 `onCreate` 方法里加上如下代码：

```
StrictMode.setThreadPolicy(  
    new StrictMode.ThreadPolicy.Builder() // 构造 StrictMode 线程对象  
        .detectDiskReads() // 当发生磁盘读操作时输出  
        .detectDiskWrites() // 当发生磁盘写操作时输出  
        .detectNetwork() // 访问网络时输出，包括磁盘读写和网络 I/O  
        .penaltyLog() // 以日志的方式输出  
        .build()  
);
```

set VmPolicy 是关于 VM虚拟机等方面的策略。

```
StrictMode.setVmPolicy(  
    new StrictMode.VmPolicy.Builder() // 构造 StrictMode VM虚拟机对象  
        .detectLeakedSqlLiteObjects() // 探测 SQLite 数据库操作  
        .detectLeakedClosableObjects() // 探测关闭操作  
        .penaltyLog() // 以日志的方式输出  
        .penaltyDeath()  
        .build()  
);
```



## 【例 6-7】从Web服务器读取图像文件

### (1) 界面布局文件activity\_main.xml

设置一个按钮，2个用于显示信息的文本框，1个显示图像的ImageView。如图所示。

(代码详见教材)



## 【例6-8】以GET方式及POST方式向Web服务器读取及发送数据。

- 依题意，在手机客户端编写界面布局文件和主程序文件，在Web服务器端编写接收GET请求的play-get.php文件和接收POST请求的play-post.php文件。
- (1) 界面布局文件  
如图所示



- (2) 主控程序 (代码见教材)
- (3) 修改配置文件, 设置网络访问权限
- (4) Web服务器端接收GET请求的play-get.php文件 (代码见教材)
- (5) Web服务器端接收POST请求的play-post.php文件 (代码见教材)

# 习题六

- 1、编写一个用户注册程序，向远程服务器注册。
- 2、编写一个具有密码验证功能的远程用户登录程序。

Q&A

谢谢大家！