



1.JavaEE的概念

J2EE(Java 2 Platform Enterprise Edition)企业版

是为开发企业环境下的应用程序提供的一套解决方案。

该技术体系中包含的技术如 Servlet Jsp等，主要针对于Web应用程序开发。

J2SEJava 2 Platform Standard Edition) 标准版

是为开发普通桌面和商务应用程序提供的解决方案。

该技术体系是其他两者的基础，可以完成一些桌面应用程序的开发。

比如Java版的扫雷。

J2ME(Java 2 Platform Micro Edition)小型版

是为开发电子消费产品和嵌入式设备提供的解决方案。

该技术体系主要应用于小型电子消费类产品，如手机中的应用程序等。

Java5.0版本后，更名为 **JAVAEE** **JAVASE** **JAVAME** http://www.csdn.net/BruceLiu_code

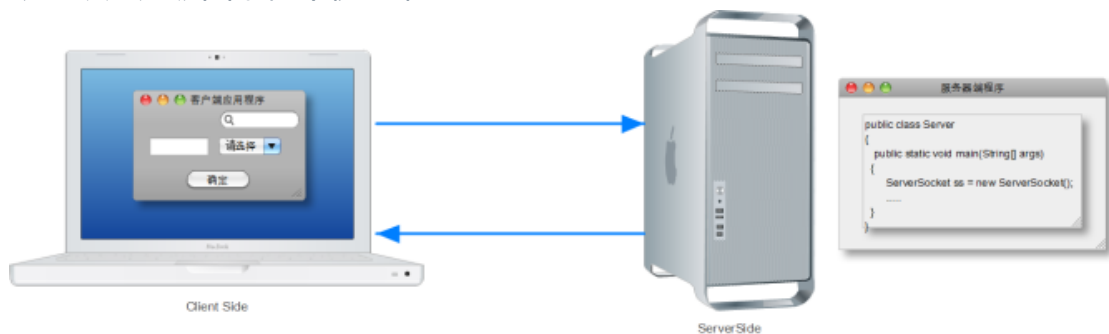
Java Enterprise Edition (Java企业版)。JavaEE并不是一个具体的技术。而是由SUN公司提出的一个Java 企业级开发的平台，是一种标准。其中包含13个技术规范。

(13个技术规范，具体参见文档)

2.Web开发常用的概念

2.1 B/S架构和C/S架构

- C/S Client/Server (客户端服务器) 需要在机器安装一个客户端来访问服务器
比如：QQ 迅雷 旺旺...好处就是访问的速度快（服务端的压力小），安全性高。
缺点就是升级困难，不便于维护。



C/S 架构需要安装对应的客户端，白话说需要下载安装包程序！

- B/S Browser/Server (浏览器服务器) 不需要安装客户端，用浏览器作为客户端来访问服务器端资源。例如：京东、淘宝、百度、优酷..... 12306售票网站 优点就是不需要安装客户端程序,只需要一个浏览器即可！ 缺点就是服务器端压力大，访问速度慢，安全性差。降低服务器压力：JS Ajax ..Jquery...



2.2 web开发是什么

WEB，在英语中web即表示网页的意思，它用于表示Internet主机上供外界访问的资源。Web开发：主要指网站的开发。JavaEE开发，主要是基于网页开发后台的应用程序。

2.3 web开发中使用到的技术

- **网页技术**：静态网页技术：HTML、CSS、JavaScript、DOM技术等。动态网页技术：ASP（微软的.NET技术）、PHP技术(使用在服务器端脚本语言。能生成动态网页)、Java中的Servlet/JSP技术。
- **其他技术**：AJAX技术、数据库技术、Web服务器技术、JDBC技术、各种框架技术等。

2.4 Web资源

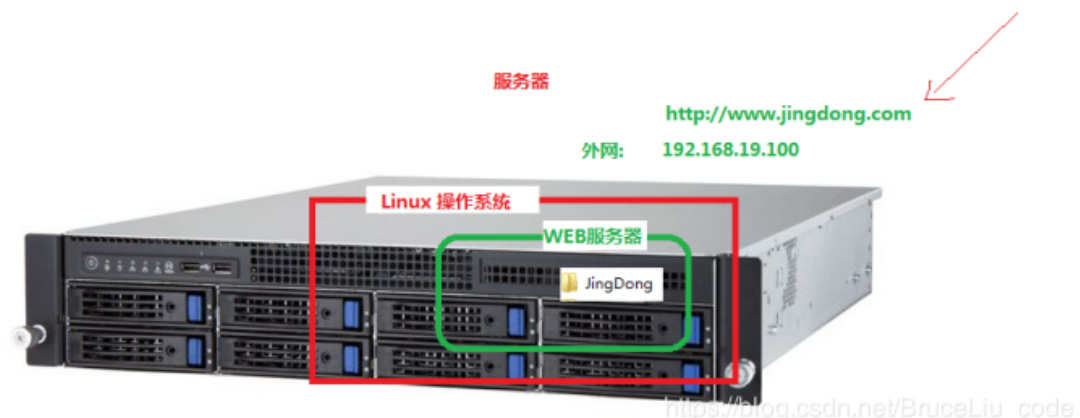
使用上述技术，就可以开发出一个web项目，web项目中包含html、css、js、图片、jsp、servlet、音频、视频等各类文件，我们把这些文件统称为web资源，而这些web资源又分为两类：

- **静态资源**：如html、css、js、图片等，一些具体的、真实存在的文件，其内容是固定的。
- **动态资源**：如jsp、servlet、PHP、ASP等。这些文件可以在运行的时候，动态的生成静态资源的内容，而没有对应的具体的文件，在不同的时间、不同的人访问这些动态资源时，程序生成的内容是不同的。

2.5 服务器的概念

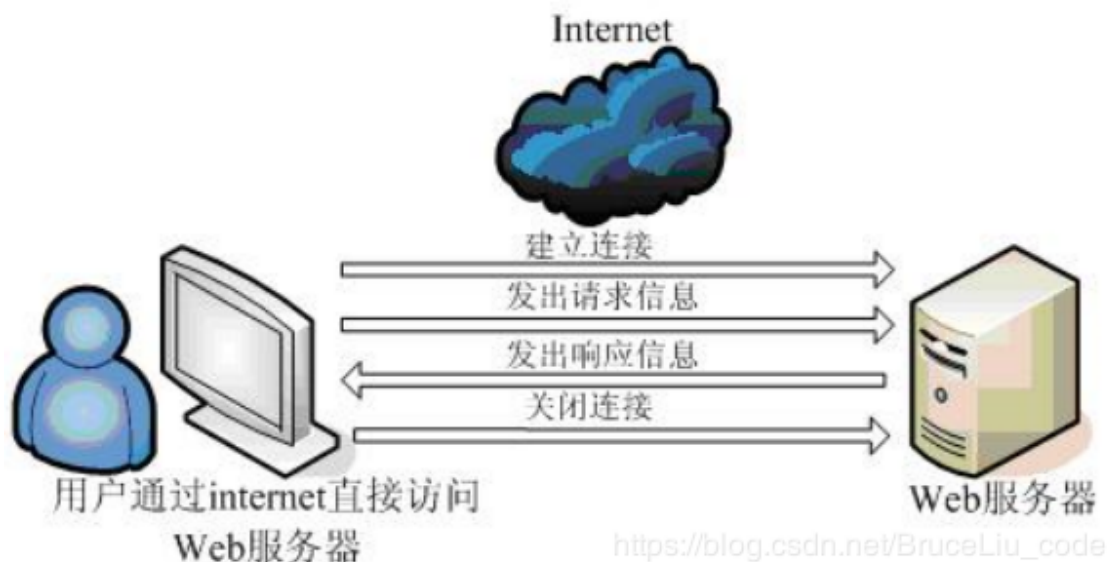
服务器分为硬件服务器和软件服务器：

- **硬件服务器**：一台电脑，而这台电脑一般不是个人电脑，而是一台性能强悍的超级计算机。硬件服务器接入互联网时，需要给其分配一个固定的IP，只有给其分配固定的IP，才能真正称之为硬件服务器。



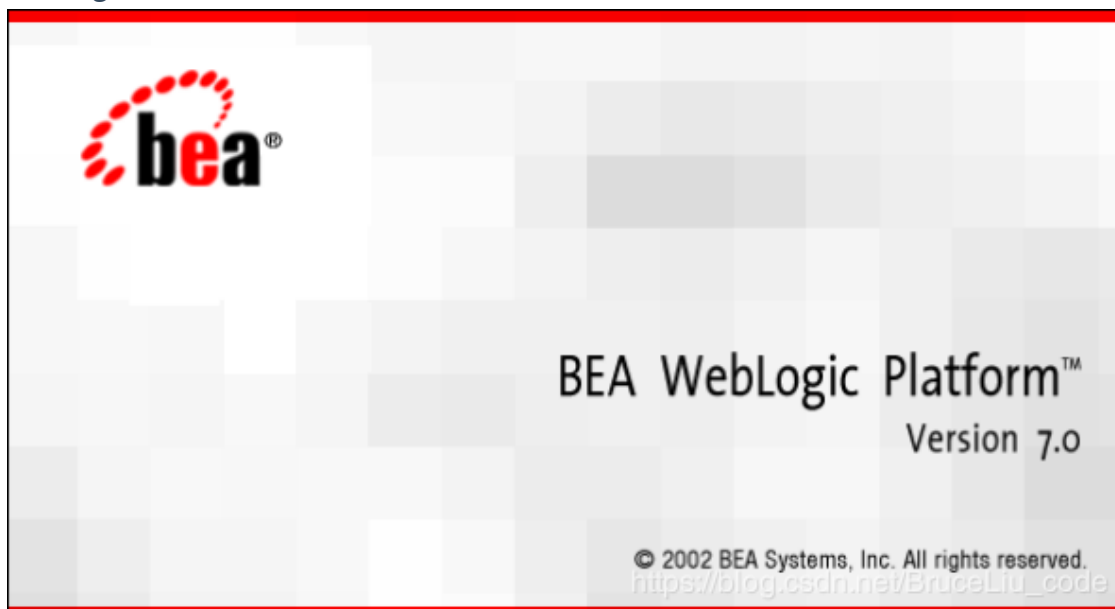
- **软件服务器**：安装在硬件服务器上的一种特殊软件，根据不同的用途，又分为：web应用服务器、邮箱服务器等。web资源需要部署到web应用服务器中，

外界才能够访问到这些web资源。



2.6 常见的服务器

- **WebLogic** WebLogic是美国bea公司出品的一个application server。BEA WebLogic是用于开发、集成、部署和管理大型分布式Web应用、网络应用和数据库应用的Java应用服务器。将Java的动态功能和Java Enterprise标准的安全性引入大型网络应用的开发、集成、部署和管理之中。是目前应用最广泛的Web服务器，支持J2EE规范，而且不断的完善以适应新的开发要求，启动界面如图 Oracle公司 Oracle数据库！ Oracle74亿美元并购SUN公司！ Oracle并购瑞典的MySQL公司！ Weblogic后来被Oracle公司收购！



- **WebSphere** WebSphere Application Server 是一种功能完善、开放的Web应用程序服务器，是IBM公司电子商务计划的核心部分，它是基于 Java 的应用环境，用于建立、部署和管理 Internet 和 Intranet Web 应用程序。这一整套产品进行了扩展，以适应 Web应用程序服务器的需要，范围从简单到高级直到企业级。启动

界面如图：



- **Tomcat** Tomcat是一个实现了JAVA EE标准的最小的WEB服务器，是Apache 软件基金会的Jakarta 项目中的一个核心项目，由Apache、Sun 和其他一些公司及个人共同开发而成。因为Tomcat 技术先进、性能稳定，而且开源免费，因而深受Java 爱好者的喜爱并得到了部分软件开发商的认可，成为目前比较流行的Web 应用服务器。学习JavaWeb开发一般都使用Tomcat服务器，该服务器支持全部JSP以及Servlet规范，启动界面如图：性能优良，小巧 免费...



- **IIS服务器**

Microsoft的Web服务器产品为Internet Information Services（IIS），IIS 是允许在公共Intranet或Internet上发布信息的Web服务器。IIS是目前最流行的Web服务器产品之一，很多著名的网站都是建立在IIS的平台上。IIS提供了一个图形界面的管理工具，称为Internet信息服务管理器，可用于监视配置和控制Internet服务。



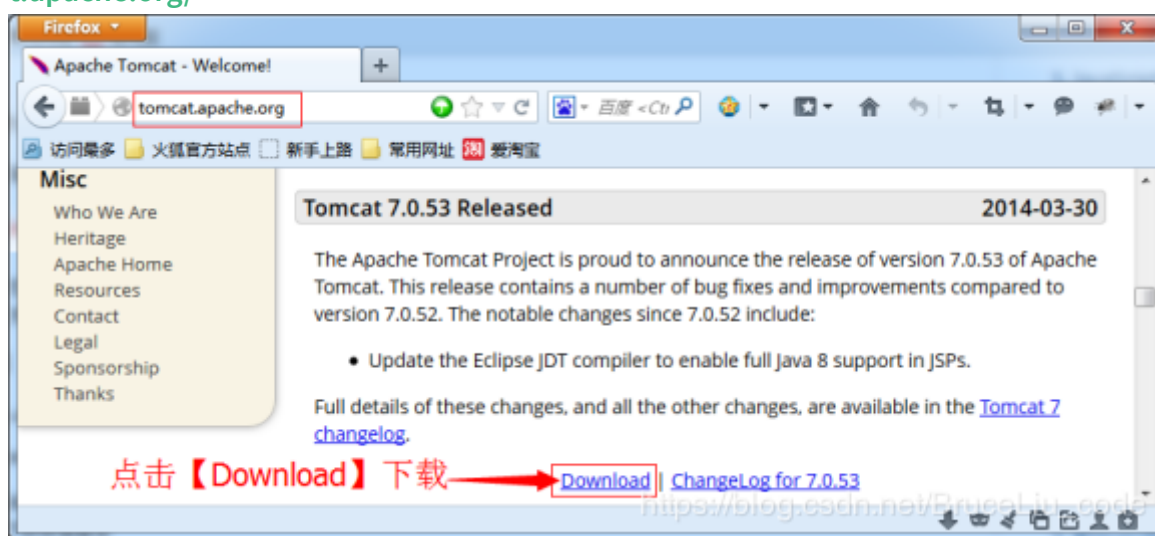
学习web开发，需要先安装一台web服务器，然后再在web服务器

中开发相应的web资源，供用户使用浏览器访问。

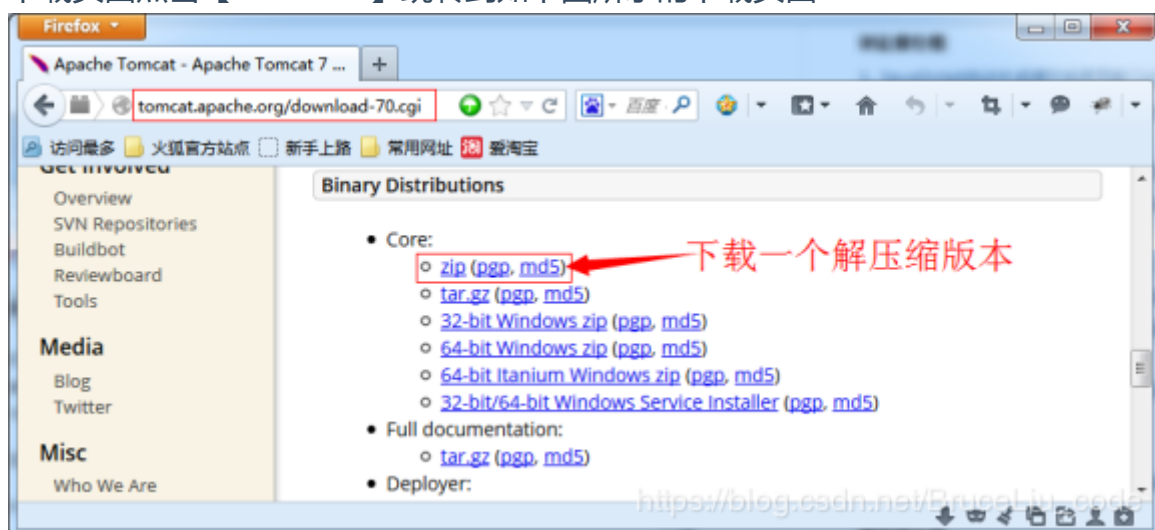
3. Tomcat服务器


3.1 下载TomCat

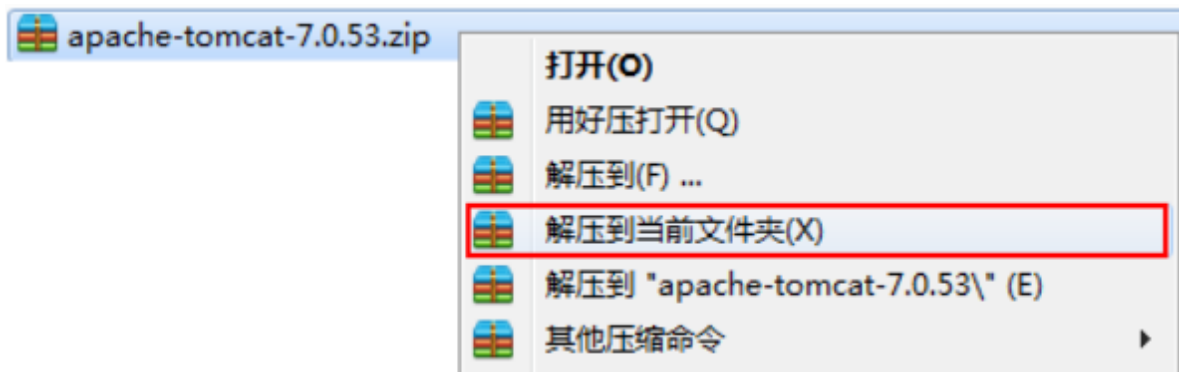
Tomcat官方站点: <http://jakarta.apache.org> 下载Tomcat安装程序包: <http://tomcat.apache.org/>




下载页面点击【Download】跳转到如下图所示的下载页面



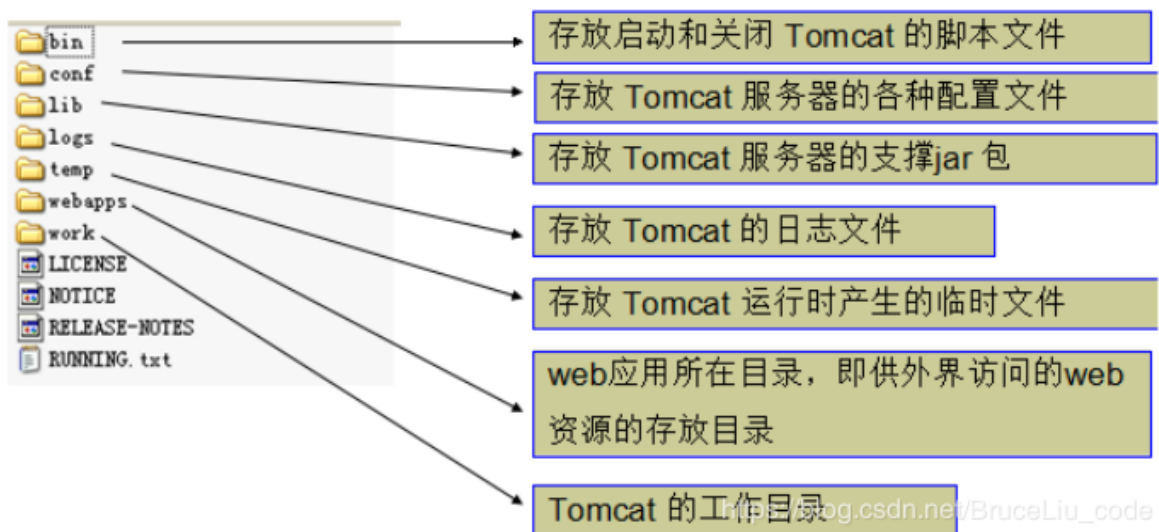
tar.gz文件是Linux操作系统下的安装版本 exe文件是Windows系统下的安装版本 zip文件是Windows系统下的压缩版本  **apache-tomcat-7.0.53.zip** 下载完成后，得到的是一个压缩包，将压缩包解压就可以完成Tomcat服务器的安装



将压缩包解压后，得到的就是如下图所示的文件夹，这样就完成了Tomcat服务器的安装。  **apache-tomcat-7.0.53**

3.2 TomCat目录结构

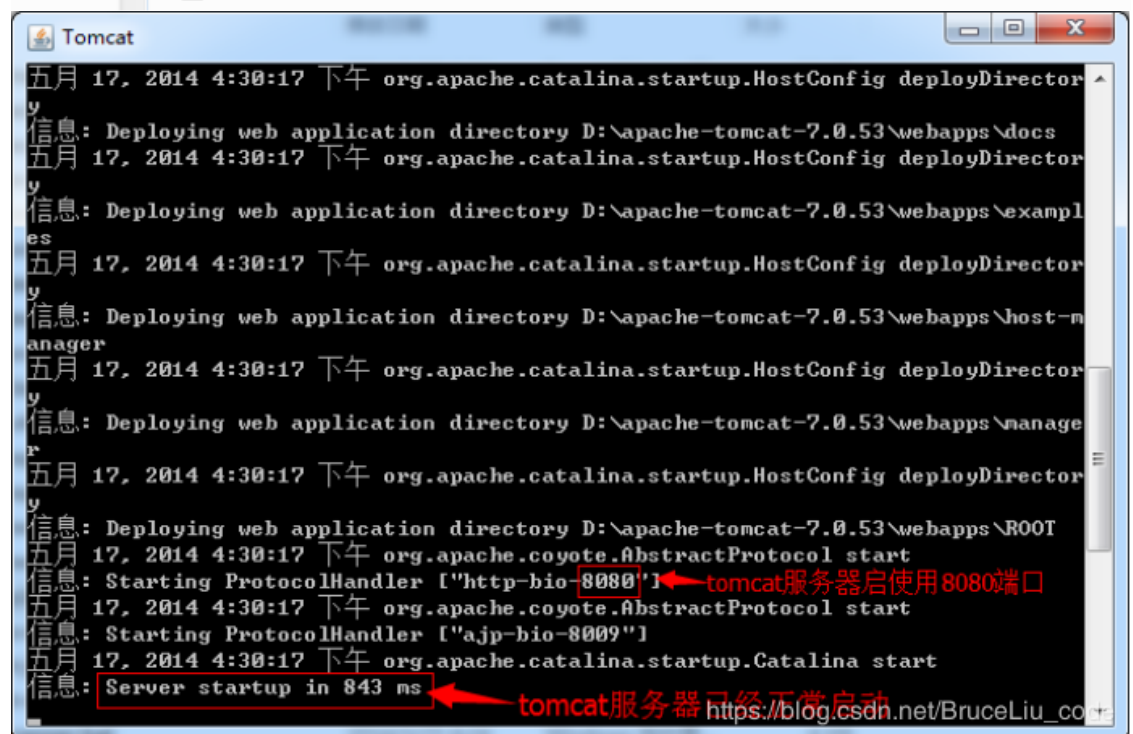
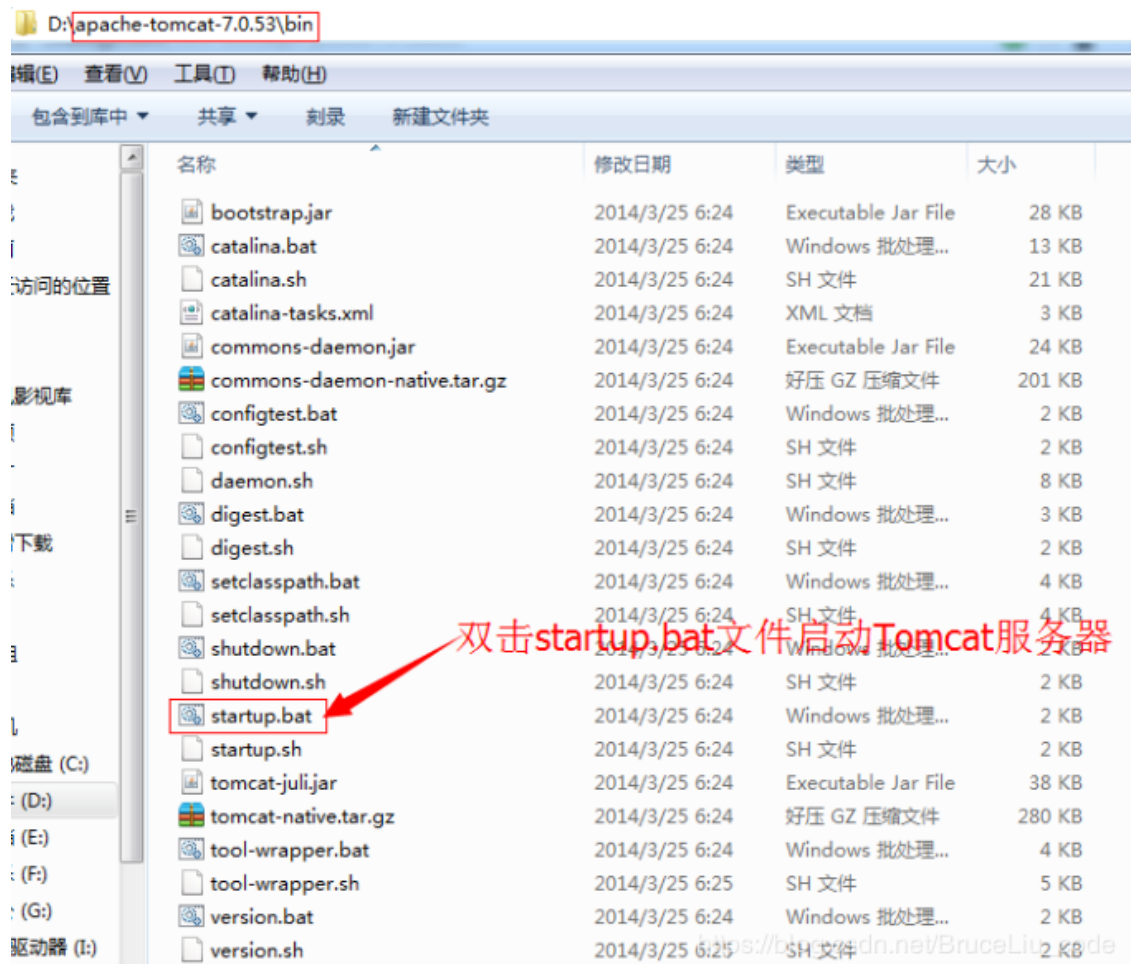
Tomcat 的目录层次结构



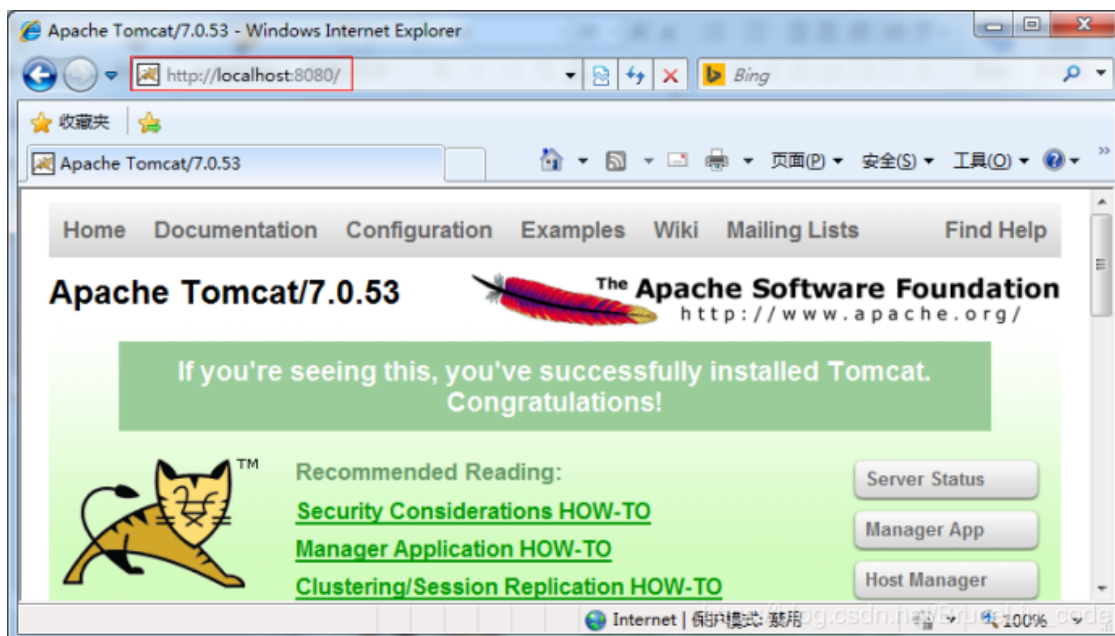
- **bin**：所有可以执行文件及命令 启动脚本：startup.bat 启动服务器 停止脚本：shutdown.bat
- **conf** (config)：tomcat的配置目录 服务器核心配置：server.xml 所有web项目默认配置：web.xml web.xml是web项目的核心配置文件，其中的很多配置，都在tomcat中的conf目录中的web.xml
- **lib**：tomcat运行时所需的jar包
- **logs**：tomcat运行时产生的日志文件
- **temp**：临时文件目录
- **webapps**：web项目默认的发布目录，我们开发的web项目，需要放在这个目录下，外界才能访问到 其中ROOT是访问tomcat时默认访问的目录，如果将其中内容删除，将web项目发布到该位置，则访问时不需要添加项目名
- **work**：tomcat运行时的工作目录，后期我们学习的JSP文件翻译后就在这个目录。

3.3 启动tomcat服务器

- 启动：双击 bin 目录下的 startup.bat 文件启动Tomcat服务器



- 测试Tomcat服务器 打开浏览器在，在浏览器的地址栏中输入：<http://127.0.0.1:8080> (推荐) 或 <http://localhost:8080>，能显示如下界面代表安装成功。



PS: localhost是在本地的hosts文件中配置的本地映射（本地DNS解析）。hosts文件在系统中的位置：C:\Windows\System32\drivers\etc\hosts

3.4 Tomcat启动常见问题

导致Tomcat服务器无法正常启动的原因一般来说就是如下的两个：

- 启动时闪退或启动过程中窗口自动关闭
- 端口占用问题

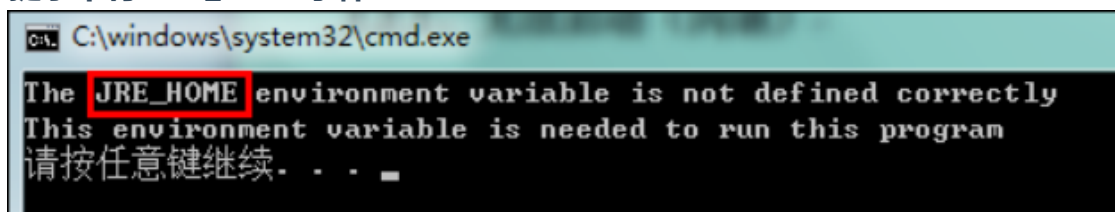
3.4.1 启动时闪退或启动过程中窗口自动关闭

查看原因：在startup.bat文件的最后书写pause命令，让运行的窗口暂停，方便查看相关提示。

```
56 call "%EXECUTABLE%" start %CMD_LINE_ARGS%
57
58 :end
59 pause
```

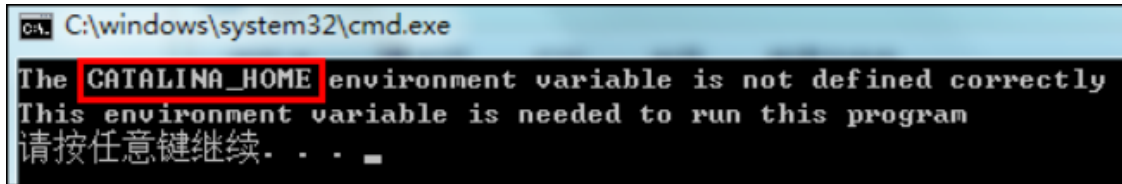
保存之后，再次双击startup.bat，并查看提示，闪退现象解决之后，去掉startup.bat文件的pause命令。

- 提示中有JAVA_HOME字样

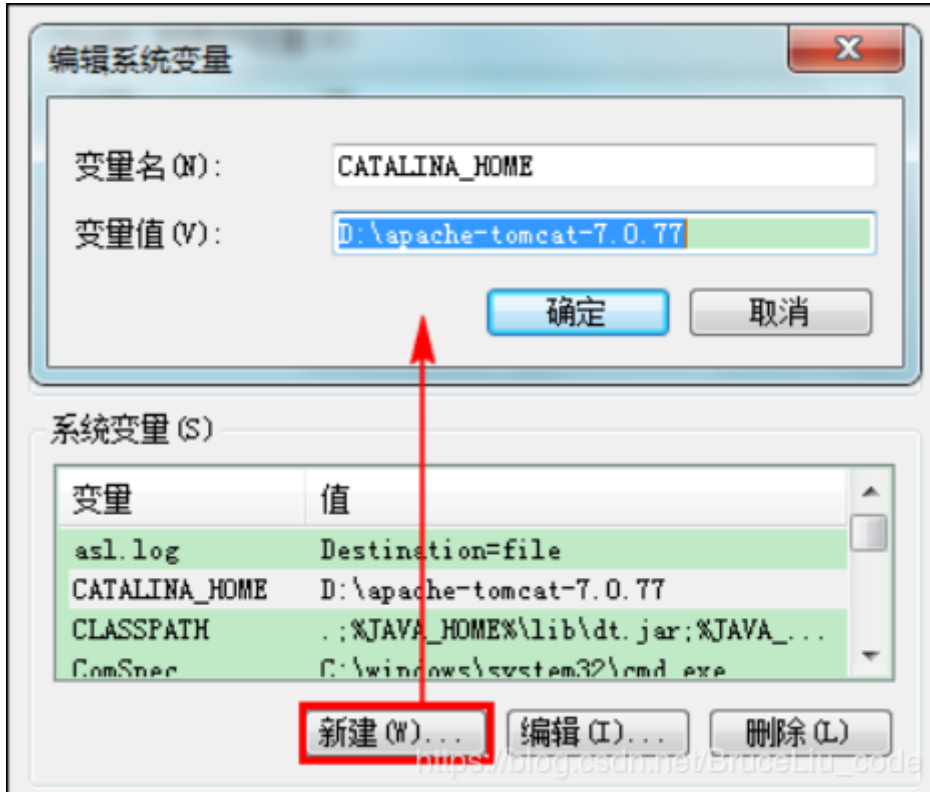


表示没有配置JAVA_HOME环境变量，也可能是配置不正确，重新配置即可！

- 提示中有CATALINA_HOME字样

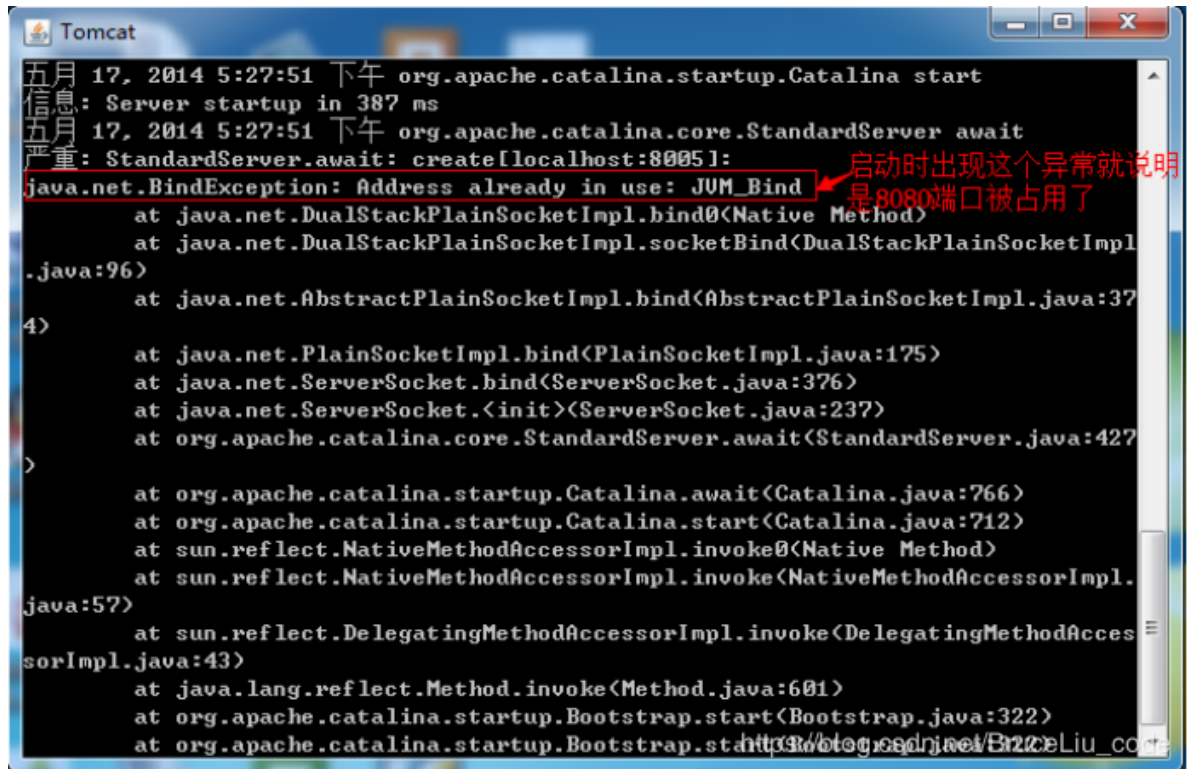


表示没有配置CATALINA_HOME环境变量，CATALINA_HOME环境变量中配置的是tomcat的安装目录，不包含bin目录。一般不需要配置。

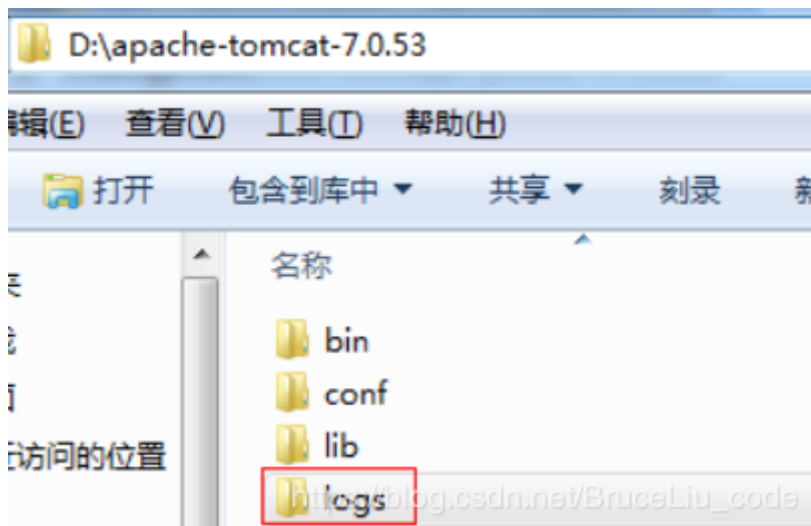


3.4.2 tomcat端口占用

因为Tomcat服务器启动时是需要默认是使用8080端口的，如果这个8080端口被别的应用程序占用了，那么Tomcat服务器就无法正常启动，看到的现象就是"Tomcat服务器启动界面会打印出异常错误信息，然后就自动关闭了"，如下图所示：



由于这个窗口从启动到关闭的时间非常短，我们很难通过这个窗口看到Tomcat启动时的报错异常信息，因此我们一般只能通过Tomcat服务器的记录的log(日志)信息去查看Tomcat服务器的运行情况。在Tomcat服务器的根目录下有一个logs文件夹，



logs文件夹存放Tomcat 的

日志文件，打开logs文件夹，可以看到里面的log文件，其中有一个以"catalina.yyyy-MM-dd.log"形式命名的log文件。打开log文件，看看里面的日志记录信息，

```

严重: Failed to initialize end point associated with ProtocolHandler ["http-bio-8080"]
java.net.BindException: Address already in use: JVM_Bind <null>:8080
    at org.apache.tomcat.util.net.JIoEndpoint.bind(JIoEndpoint.java:410)
    at org.apache.tomcat.util.net.AbstractEndpoint.init(AbstractEndpoint.java:640)
    at org.apache.coyote.AbstractProtocol.init(AbstractProtocol.java:434)
    at org.apache.coyote.http11.AbstractHttp11JsseProtocol.init(AbstractHttp11JsseProtocol.java:119)
    at org.apache.catalina.connector.Connector.initInternal(Connector.java:978)
    at org.apache.catalina.util.LifecycleBase.init(LifecycleBase.java:102)
    at org.apache.catalina.core.StandardService.initInternal(StandardService.java:559)
    at org.apache.catalina.util.LifecycleBase.init(LifecycleBase.java:102)
    at org.apache.catalina.core.StandardServer.initInternal(StandardServer.java:813)
    at org.apache.catalina.util.LifecycleBase.init(LifecycleBase.java:102)
    at org.apache.catalina.startup.Catalina.load(Catalina.java:638)
    at org.apache.catalina.startup.Catalina.load(Catalina.java:663)
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
    at java.lang.reflect.Method.invoke(Method.java:601)
    at org.apache.catalina.startup.Bootstrap.load(Bootstrap.java:280)
    at org.apache.catalina.startup.Bootstrap.main(Bootstrap.java:454)
Caused by: java.net.BindException: Address already in use: JVM_Bind
    at java.net.DualStackPlainSocketImpl.bind0(Native Method)
    at java.net.DualStackPlainSocketImpl.socketBind(DualStackPlainSocketImpl.java:96)
    at java.net.AbstractPlainSocketImpl.bind(AbstractPlainSocketImpl.java:374)
    at java.net.PlainSocketImpl.bind(PlainSocketImpl.java:175)
    at java.net.ServerSocket.bind(ServerSocket.java:376)
    at java.net.ServerSocket.<init>(ServerSocket.java:237)
    at java.net.ServerSocket.<init>(ServerSocket.java:181)
    at org.apache.tomcat.util.net.DefaultServerSocketFactory.createSocket(DefaultServerSocketFactory.java:49)
    at org.apache.tomcat.util.net.JIoEndpoint.bind(JIoEndpoint.java:397)
    ... 17 more

```

日志信息里面清楚的记录了Tomcat服务器的运行情况，如果因为8080端口被占用而无法启动，就会记录如上图所示的异常信息，通过查看异常信息，我们就知道Tomcat服务器为什么不能够正常启动了！

解决方案一： 查看本地端口使用情况：在dos窗口中输入netstat -nao，回车：

```

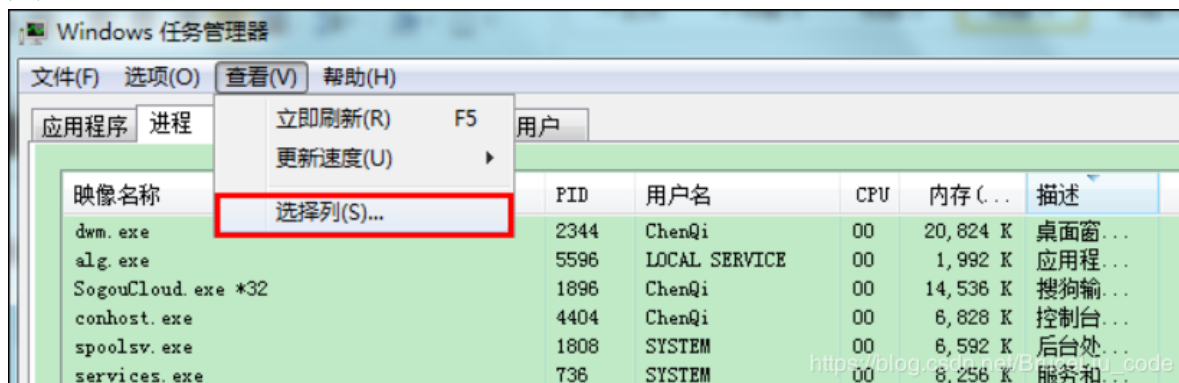
C:\Users\ChenQi>netstat -nao

```

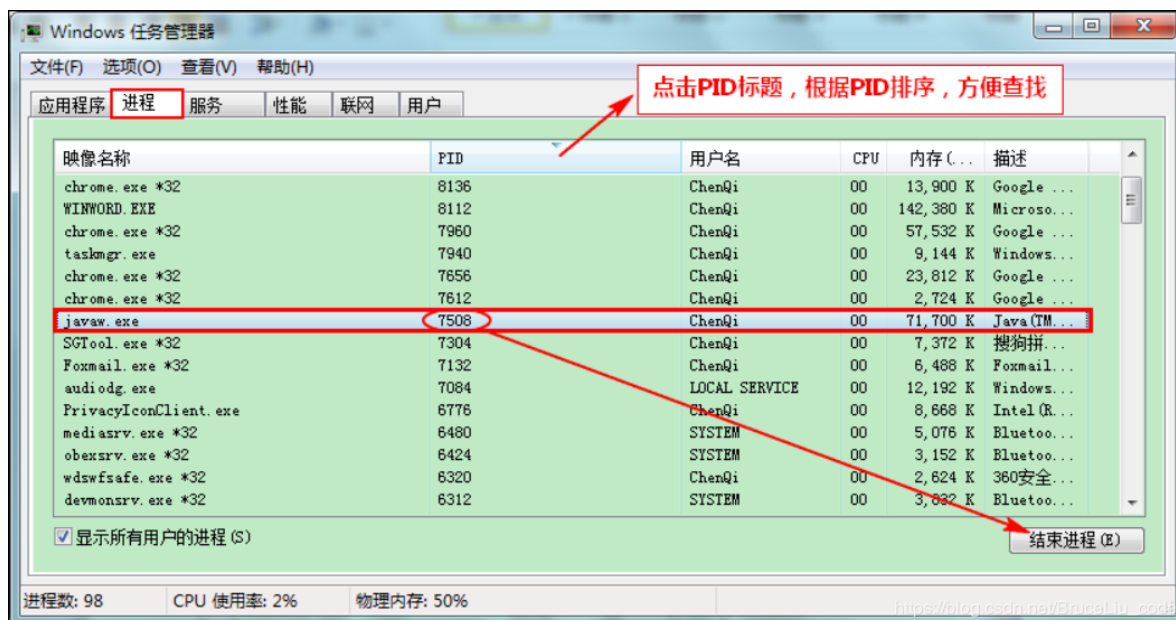
活动连接

协议	本地地址	外部地址	状态	PID
TCP	0.0.0.0:135	0.0.0.0:0	LISTENING	292
TCP	0.0.0.0:443	0.0.0.0:0	LISTENING	2588
TCP	0.0.0.0:445	0.0.0.0:0	LISTENING	4
TCP	0.0.0.0:623	0.0.0.0:0	LISTENING	4200
TCP	0.0.0.0:1521	0.0.0.0:0	LISTENING	2332
TCP	0.0.0.0:2425	0.0.0.0:0	LISTENING	4128
TCP	0.0.0.0:2869	0.0.0.0:0	LISTENING	4
TCP	0.0.0.0:3306	0.0.0.0:0	LISTENING	2124
TCP	0.0.0.0:7000	0.0.0.0:0	LISTENING	2332
TCP	0.0.0.0:8009	0.0.0.0:0	LISTENING	7508
TCP	0.0.0.0:8080	0.0.0.0:0	LISTENING	7508
TCP	0.0.0.0:16992	0.0.0.0:0	LISTENING	4200
TCP	0.0.0.0:30985	0.0.0.0:0	LISTENING	4128

找到8080端口那一行的PID列，暂且先记下对应的数字，然后使用任务管理器，查看是哪个进程占用了该端口：任务管理器的进程中默认不显示PID列，需要在查看中让其显示出来：



切换到进程选项卡，根据第1步中PID的值，找到对应的进程，把进程干掉：



注意：如果这个进程是关键进程（用户名为SYSTEM），这时不能停止这个进程！否则可能导致系统不稳定甚至更严重的后果。

解决方案二：修改tomcat的默认端口： 进入tomcat的安装目录，打开conf/server.xml，找到Connector标签，修改port属性：8090或9090都可以！

71	<Connector port="8080" protocol="HTTP/1.1"
72	connectionTimeout="20000"
73	redirectPort="8443" />