

环境变量

环境变量 (environment variables) 一般是指在操作系统中用来指定操作系统运行环境的一些参数, 如: 临时文件夹位置和系统文件夹位置等。

环境变量是在操作系统中一个具有特定名字的对象, 它包含了一个或者多个应用程序所将使用到的信息。例如 Windows 和 DOS 操作系统中的 path 环境变量, 当要求系统运行一个程序而没有告诉它程序所在的完整路径时, 系统除了在当前目录下面寻找此程序外, 还应到 path 中指定的路径去找。用户通过设置环境变量, 来更好的运行进程。

环境变量相当于给系统或用户应用程序设置的一些参数, 具体起什么作用这当然和具体的环境变量相关。比如 path, 是告诉系统, 当要求系统运行一个程序而没有告诉它程序所在的完整路径时, 系统除了在当前目录下面寻找此程序外, 还应到哪些目录下去寻找; 再如 tc 或 vc++ 中, set include=path1;path2; 是告诉编译程序到哪里去找 .h 类型的文件; 当然不仅仅是指定什么路径, 还有其它的作用的, 如 set dircmd=/4 设置一个环境变量的作用是在使用 dir 命令时会把 /4 作为缺省的参数添加到你的 dir 命令之后, 就像你的每个命令都加了 /4 参数, 它实际上是给命令解释程序 command 设置的一个环境变量, 并且是给 dir 这个内部命令设置的。

DWORD GetEnvironmentVariable(LPCSTR lpName, LPSTR lpBuffer, DWORD dSize), 参数 lpName 是你要求查询的环境变量的名, lpBuffer 是返回你所指定的环境变量的值的, dSize 是告诉这个函数 lpBuffer 可以存放多少个字节。

分析本地故障时原因很可能就是因为环境变量中的默认路径被删除的结果, 默认路径一经设置, 当前系统如有程序运行时需要某些 DLL 或 EXE 文件, 以及 Active 控件时就会到所有默认路径中去查找, 如果在这些目录中查找到相应的程序则自动加载, 查找不到则报告缺少某某文件的错误信息。

很多朋友会在自己的计算机上安装双系统, 例如 C 盘安装 Windows XP, D 盘安装 Windows 7。可是某些软件往往只在 Windows XP 系统中安装, Windows 7 系统中是无法正常使用的, 比较麻烦却有效的方法是再安装一遍。当我们了解了环境变量中的用途后就可以很好解决双系统的软件共用问题。

为什么在 Windows XP 中安装了的软件在 Windows 7 下无法运行呢 (绿色软件除外)。原因是安装软件时往往须要向系统目录中复制某些文件, 而使用另外一个系统时会由于缺少这些文件而无法运行。因此, 我们可以通过设置环境变量的方法来解决这个问题。

总之一句话, 环境变量就是告诉系统哪个软件在哪个地方!!!

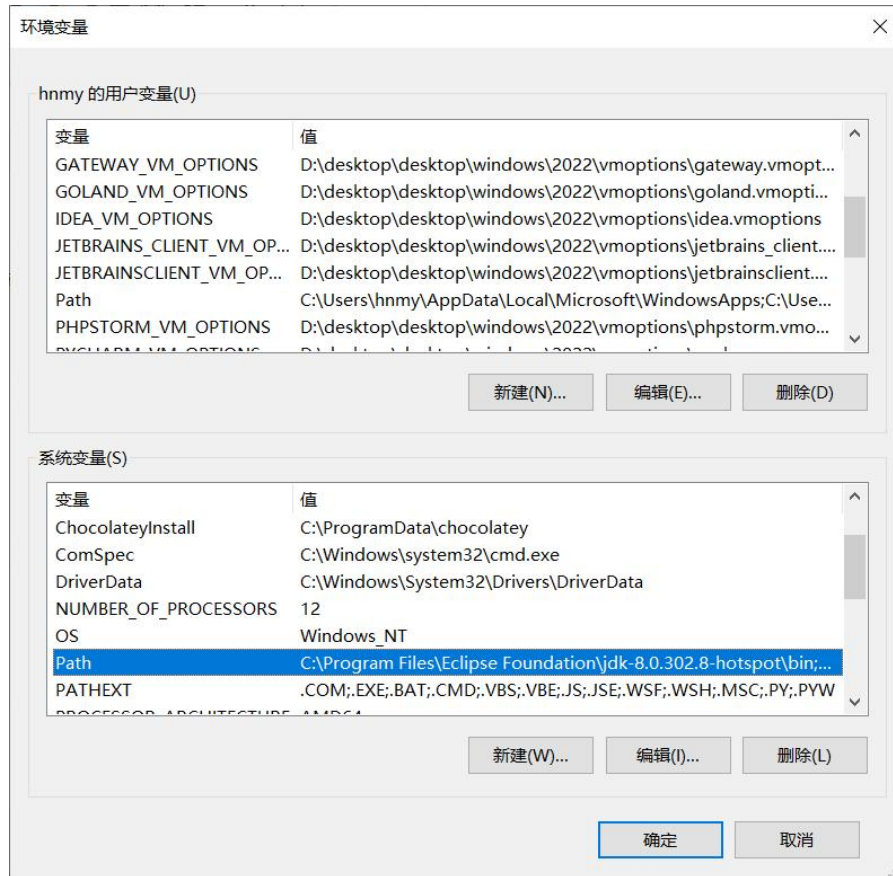
以 Java 环境为例,我们编译 Java 代码需要用到 Java 给我们提供的 Java 的 jdk,我们可以在 cmd 里面输入 java 来查看 cmd 的返回数据。

```
C:\Windows\system32\cmd.exe
C:\Users\hnmmy>java
用法: java [-options] class [args...]
           (执行类)
 或 java [-options] -jar jarfile [args...]
           (执行 jar 文件)
其中选项包括:
  -d32          使用 32 位数据模型 (如果可用)
  -d64          使用 64 位数据模型 (如果可用)
  -server       选择 "server" VM
                默认 VM 是 server.

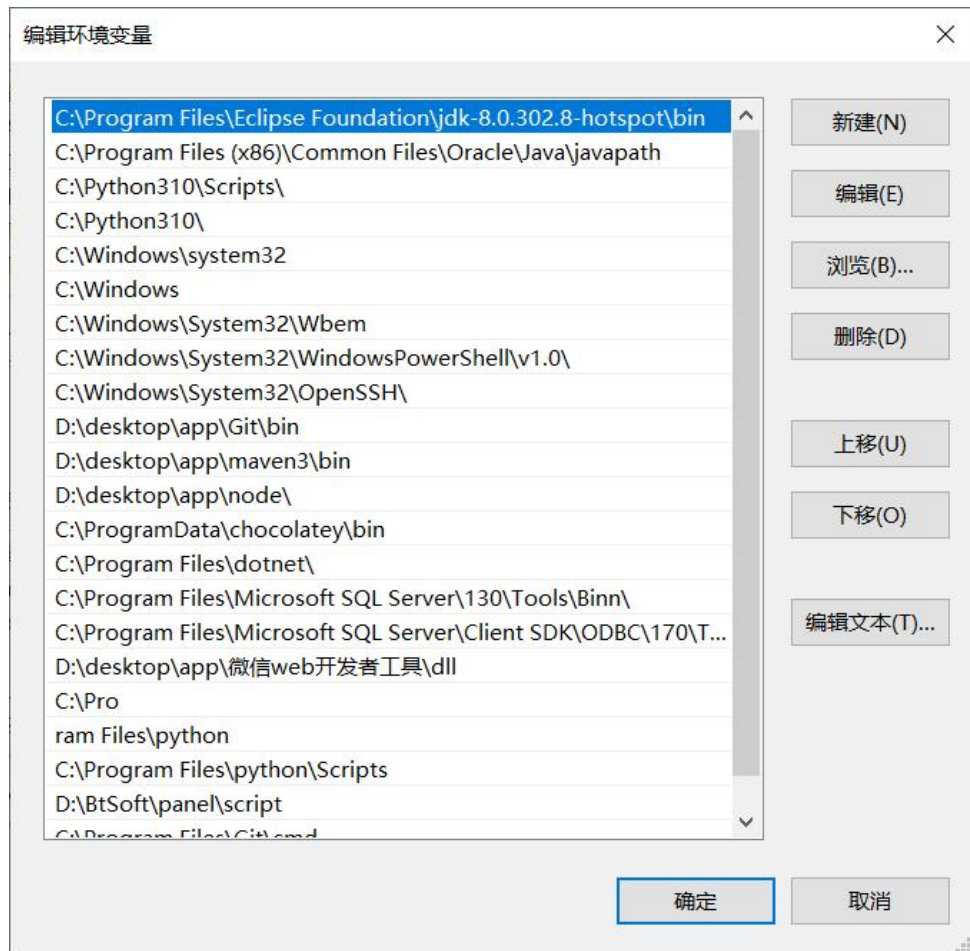
  -cp <目录和 zip/jar 文件的类搜索路径>
  -classpath <目录和 zip/jar 文件的类搜索路径>
              用 ; 分隔的目录, JAR 档案
              和 ZIP 档案列表, 用于搜索类文件。
  -D<名称>=<值> 设置系统属性
  -verbose:[class|gc|jni] 启用详细输出
  -version      输出产品版本并退出
  -version:<值> 警告: 此功能已过时, 将在
                未来发行版中删除。
                需要指定的版本才能运行
  -showversion 输出产品版本并继续
  -jre-restrict-search | -no-jre-restrict-search
                警告: 此功能已过时, 将在
                未来发行版中删除。
                在版本搜索中包括/排除用户专用 JRE
```

事实上,我们输入 Java 命令后,我们实际上运行的程序为环境变量里面的 Java.exe 文件,并且将我们后面的参数传递给 Java.exe,然后该文件运行后做出反应,返回数据,该数据将展示在 cmd 命令行里面,这就是我们看到的内容。





在这里我们可以看到我们 Java 所在的目录



其实，我们也可以用命令来查找我们的 Java 在哪个地方，这个命令就是 `where java`，这样我们就可以看到我们的 Java 文件的目录的绝对目录。

```
C:\Windows\system32\cmd.exe
C:\Users\hnmmy>where java
C:\Program Files\Eclipse Foundation\jdk-8.0.302.8-hotspot\bin\java.exe
C:\Program Files (x86)\Common Files\Oracle\Java\javapath\java.exe
C:\Users\hnmmy>
```

同理可得，我们可以查看其他软件的位置

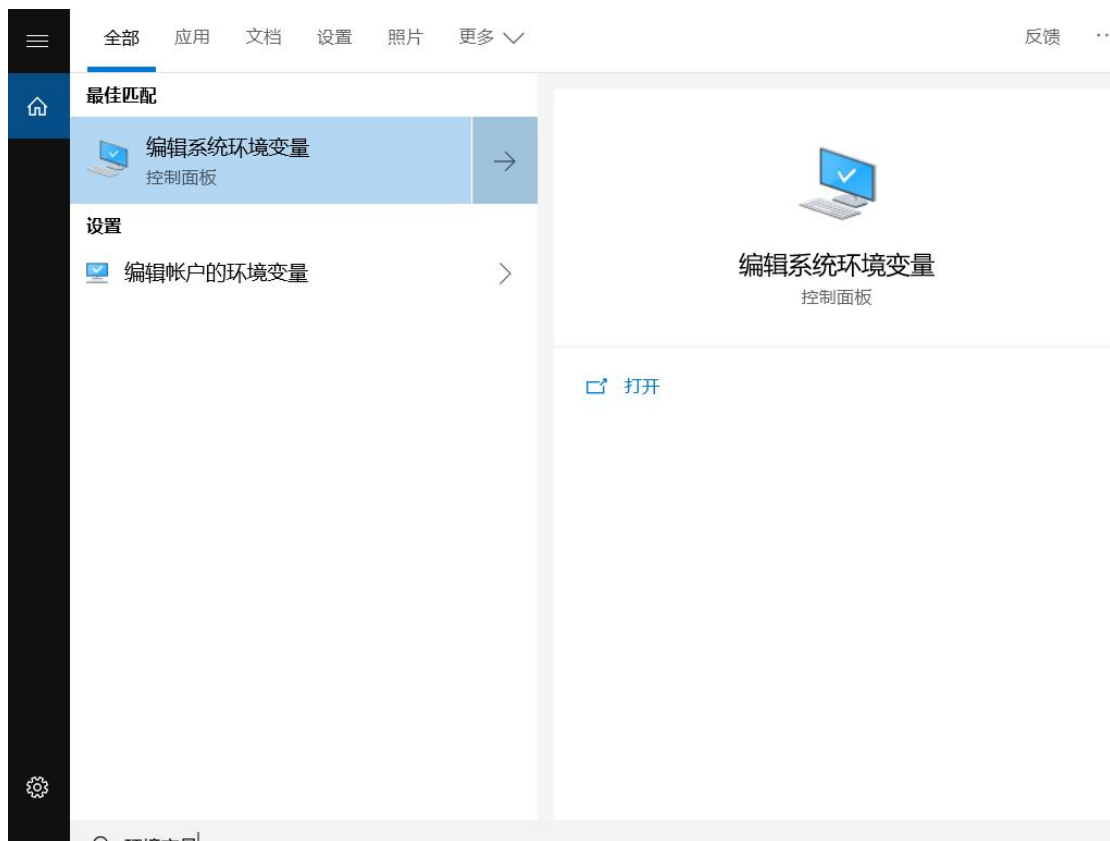
```
C:\Windows\system32\cmd.exe

C:\Users\hnmmy>where git
D:\desktop\app\Git\bin\git.exe
C:\Program Files\Git\cmd\git.exe

C:\Users\hnmmy>
```

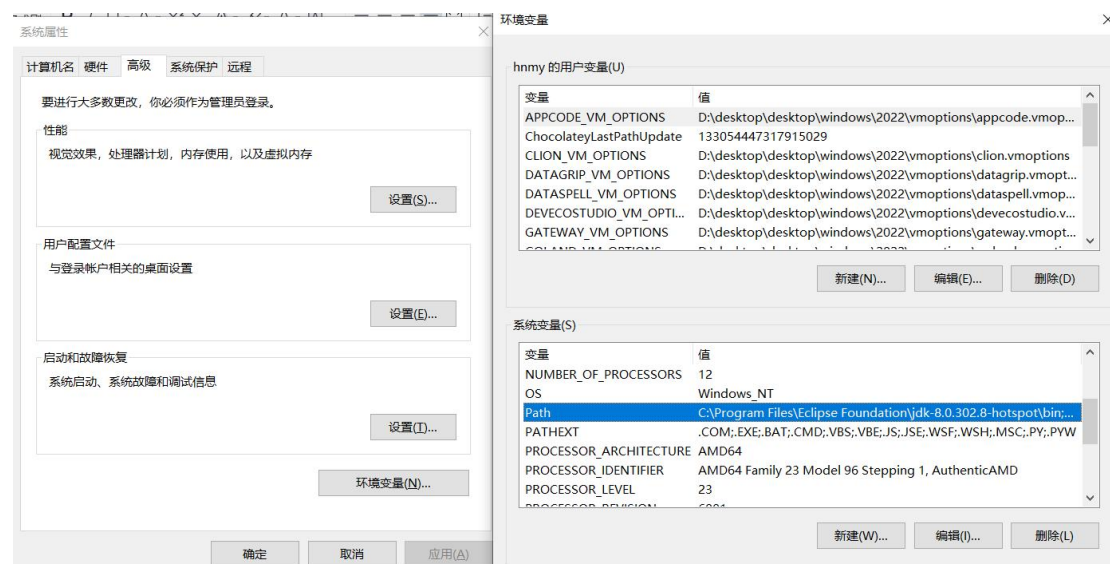
那么，如何配置环境变量呢？

我们可以直接按 **win** 键，然后输入 环境变量 4 个字来进行搜索，然后按回车键就行，就可以打开环境变量的主页面。





然后我们点击右下角的 环境变量 按钮，进入环境变量编辑页面，后面点击 Path 进入配置页面。



最后点击右上角新增按钮，将目录地址粘贴进入输入框即可

编辑环境变量



C:\Python310\Scripts\
C:\Python310\
C:\Windows\system32
C:\Windows
C:\Windows\System32\Wbem
C:\Windows\System32\WindowsPowerShell\v1.0\
C:\Windows\System32\OpenSSH\
D:\desktop\app\Git\bin
D:\desktop\app\maven3\bin
D:\desktop\app\node\
C:\ProgramData\chocolatey\bin
C:\Program Files\dotnet\
C:\Program Files\Microsoft SQL Server\130\Tools\Binn\
C:\Program Files\Microsoft SQL Server\Client SDK\ODBC\170\T...
D:\desktop\app\微信web开发者工具\dll
C:\Pro
ram Files\python
C:\Program Files\python\Scripts
D:\BtSoft\panel\script
C:\Program Files\Git\cmd

新建(N)

编辑(E)

浏览(B)...

删除(D)

上移(U)

下移(O)

编辑文本(T)...

确定

取消