**SHELL：**

在计算机科学中，Shell俗称壳（用来区别于核），是指“提供使用者使用界面”的软件（命令解析器）。它类似于DOS下的command和后来的cmd.exe。它接收用户命令，然后调用相应的应用程序。在Linux系统中shell有自己的内部命令和环境变量。

**环境变量：**

|  |  |
| --- | --- |
| myshell常用环境变量：变量名 | 变量值 |
| HOST | 主机名 |
| USER | 用户名 |
| PATH | 文件搜索路径 |
| PWD | 当前工作目录 |
| HOME | 主目录 |
| COMP | 一级提示符变量 |
| ACOMP | 二级提示符变量 |
| SHELL | 当前SHELL所在目录 |

**输入输出重定向：**

Linux环境中支持输入输出重定向，用符号<和>来表示。  
0、1和2分别表示标准输入、标准输出和标准错误信息输出，可以用来指定需要重定向的标准输入或输出，比如 2>lee.dat 表示将错误信息输出到文件lee.dat中。  
同时，还可以在这三个标准输入输出之间实现重定向，比如将错误信息重定向到标准输出，可以用 2>&1来实现。  
Linux下还有一个特殊的文件/dev/null，它就像一个无底洞，所有重定向到它的信息都会消失得无影无踪。这一点非常有用，当我们不需要回显程序的所有信息时，就可以将输出重定向到/dev/null。  
如果想要正常输出和错误信息都不显示，则要把标准输出和标准错误都重定向到/dev/null， 例如：  
# ls 1>/dev/null 2>/dev/null  
还有一种做法是将错误重定向到标准输出，然后再重定向到 /dev/null，例如：  
# ls >/dev/null 2>&1  
注意：此处的顺序不能更改，否则达不到想要的效果，此时先将标准输出重定向到 /dev/null，然后将标准错误重定向到标准输出，由于标准输出已经重定向到了/dev/null，因此标准错误也会重定向到/dev/null。

**程序环境：**

程序环境并不是把各种功能简单地拼装在一起，而是把它们有机地结合起来，统一在一个图形化操作界面下，为程序设计人员提供尽可能高效、便利的服务。例如，程序设计过程中为了排除语法错误，需要反复进行编译——查错——修改——再编译的循环，集成开发环境就使各步骤之间能够方便快捷地切换，输入[源程序](http://baike.baidu.com/view/546605.htm" \t "http://baike.baidu.com/_blank)后用简单的菜单命令或[快捷键](http://baike.baidu.com/view/103895.htm" \t "http://baike.baidu.com/_blank)启动编译，出现错误后又能立即转到对源程序的修改，甚至直接把[光标](http://baike.baidu.com/view/236729.htm" \t "http://baike.baidu.com/_blank)定位到出错的位置上。再如，集成开发环境的[编辑器](http://baike.baidu.com/view/499644.htm" \t "http://baike.baidu.com/_blank)除了具备一般[文本编辑器](http://baike.baidu.com/view/487023.htm" \t "http://baike.baidu.com/_blank)的基本功能外，还能根据C的语法规则，自动识别[程序](http://baike.baidu.com/view/17674.htm" \t "http://baike.baidu.com/_blank)文本中的不同成分，并且用不同的颜色显示不同的成分，对使用者产生很好的提示效果。

**管道：**

管道是一种两个进程间进行单向通信的机制。因为管道传递数据的单向性，管道又称为半双工管道。管道的这一特点决定了器使用的局限性。管道是Linux支持的最初Unix IPC形式之一，具有以下特点：  
  
1.数据只能由一个进程流向另一个进程（其中一个读管道，一个写管道）；如果要进行双工通信，需要建 立两个管道。  
2.管道只能用于父子进程或者兄弟进程间通信。，也就是说管道只能用于具有亲缘关系的进程间通信。  
  
   除了以上局限性，管道还有其他一些不足，如管道没有名字（匿名管道），管道的缓冲区大小是受限制的。管道所传输的是无格式的字节流。这就需要管道输入方和输出方事先约定好数据格式。虽然有那么多不足，但对于一些简单的进程间通信，管道还是完全可以胜任的。  
   使用管道进行通信时，两端的进程向管道读写数据是通过创建管道时，系统设置的文件描述符进行的。从本质上说，管道也是一种文件，但它又和一般的文件有所不同，可以克服使用文件进行通信的两个问题，这个文件只存在内存中。  
    通过管道通信的两个进程，一个进程向管道写数据，另外一个从中读数据。写入的数据每次都添加到管道缓冲区的末尾，读数据的时候都是从缓冲区的头部读出数据的。

**后台程序执行:**

在控制台运行环境的时候再命令后边加上&这个标志就可以了，Linux就可以在后台运行该任务。，主线程直接跳转到等待下一次输入，而不必等待上一个命令执行完毕。子线程执行完毕后也自动结束该进程。在后台运行任务，表示在前台的Console中可以执行其他任务。