

Linux系统中重定向命令行的技巧总结 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/291/2021_2022_Linux_E7_B3_BB_E7_BB_c103_291005.htm 1.标准输入的控制 语法：命令 例如：

mail -s “ mail test ” wesongzhou@hotmail.com 题名称

为mail test，送给收信人。 2.标准输出的控制 语法：命令> 文件

将命令的执行结果送至指定的文件中。 例如：ls -l > list将

执行“ls -l”命令的结果写入文件list中。 语法：命令> ! 文件

将命令的执行结果送至指定的文件中，若文件已经存在，

则覆盖。 例如：ls -lg > ! list 将执行“ls -lg”命令的结果覆

盖写入文件list中。 语法：命令> & error 将编译file1.c 文件

时所产生的任何信息写入文件error中。 语法：命令>> 文件将

命令执行的结果附加到指定的文件中。 例如：ls -lag >> list将

执行“ls -lag”命令的结果附加到文件list中。 语法：命

令>> & error 将编译file2.c 文件时屏幕所产生的任何信息附

加到文件error中。 关于输入、输出和错误输出 在字符终端环

境中，标准输入/标准输出的概念很好理解。输入即指对一个

应用程序或命令的输入，无论是从键盘输入还是从别的文件

输入；输出即指应用程序或命令产生的一些信息；与

Windows系统下不同的是，Linux系统下还有一个标准错误输

出的概念，这个概念主要是为程序调试和系统维护目的而设

置的，错误输出于标准输出分开可以让一些高级的错误信息

不干扰正常的输出信息，从而方便一般用户的使用。在

Linux系统中：标准输入（stdin）默认为键盘输入；标准输出

（stdout）默认为屏幕输出；标准错误输出（stderr）默认也

是输出到屏幕（上面的std表示standard）。在BASH中使用

这些概念时一般将标准输出表示为 1，将标准错误输出表示为 2。下面我们举例来说明如何使用他们，特别是标准输出和标准错误输出。输入、输出及标准错误输出主要用于 I/O 的重定向，就是说需要改变他们的默认设置。先看这个例子：
`$ ls > ls_result $ ls -l >> ls_result` 上面这两个命令分别将 `ls` 命令的结果输出重定向到 `ls_result` 文件中和追加到 `ls_result` 文件中，而不是输出到屏幕上。"`>`" 就是输出（标准输出和标准错误输出）重定向的代表符号，连续两个 "`>>`" 符号，即 "`>>`" 则表示不清除原来的而追加输出。下面再来看一个稍微复杂的例子：
`$ find /home -name lost* 2> err_result` 这个命令在 "`>`" 符号之前多了一个 "2"，"`2>`" 表示将标准错误输出重定向。由于 `/home` 目录下有些目录由于权限限制不能访问，因此会产生一些标准错误输出被存放在 `err_result` 文件中。大家可以设想一下 `find /home -name lost*2>>err_result` 命令会产生什么结果？如果直接执行 `find /home -name lost* > all_result`，其结果是只有标准输出被存入 `all_result` 文件中，要想让标准错误输出和标准输入一样都被存入到文件中，那该怎么办呢？看下面这个例子：
`$ find /home -name lost* > all_result 2> & . all_result` 如果那些出错信息并不重要，下面这个命令可以让你避开众多无用出错信息的干扰：
`$ find /home -name lost* 2> /dev/null` 同学们回去后还可以再试验一下如下几种重定向方式，看看会出什么结果，为什么？
`$ find /home -name lost* > all_result 1> & . 2 $ find /home -name lost* 2> & & & .-` 表示将 n 号输出关闭 `> & .-` 表示将标准输出关闭 100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com