重定向命令列表如下：

|  |  |
| --- | --- |
| **命令** | **说明** |
| command > file | 将输出重定向到 file。 |
| command < file | 将输入重定向到 file。 |
| command >> file | 将输出以追加的方式重定向到 file。 |
| n > file | 将文件描述符为 n 的文件重定向到 file。 |
| n >> file | 将文件描述符为 n 的文件以追加的方式重定向到 file。 |
| n >& m | 将输出文件 m 和 n 合并。 |
| n <& m | 将输入文件 m 和 n 合并。 |
| << tag | 将开始标记 tag 和结束标记 tag 之间的内容作为输入。 |

*需要注意的是文件描述符 0 通常是标准输入（STDIN），1 是标准输出（STDOUT），2 是标准错误输出（STDERR）。*

## 输出重定向

command1 > file1

### **实例**

执行下面的 who 命令，它将命令的完整的输出重定向在用户文件中(users):

$ who > users

输出重定向会覆盖文件内容，请看下面的例子：

$ echo "菜鸟教程：www.runoob.com" > users

$ cat users

菜鸟教程：www.runoob.com

$

如果不希望文件内容被覆盖，可以使用 >> 追加到文件末尾，例如：

$ echo "菜鸟教程：www.runoob.com" >> users

## 输入重定向

和输出重定向一样，Unix 命令也可以从文件获取输入，语法为：

command1 < file1

### **实例**

$ wc -l < users

2

### **重定向深入讲解**

一般情况下，每个 Unix/Linux 命令运行时都会打开三个文件：

标准输入文件(stdin)：stdin的文件描述符为0，Unix程序默认从stdin读取数据。

标准输出文件(stdout)：stdout 的文件描述符为1，Unix程序默认向stdout输出数据。

标准错误文件(stderr)：stderr的文件描述符为2，Unix程序会向stderr流中写入错误信息。

默认情况下，command > file 将 stdout 重定向到 file，command < file 将stdin 重定向到 file。

如果希望 stderr 重定向到 file，可以这样写：

$ command 2 > file

如果希望 stderr 追加到 file 文件末尾，可以这样写：

$ command 2 >> file

**2** 表示标准错误文件(stderr)。

如果希望将 stdout 和 stderr 合并后重定向到 file，可以这样写：

$ command > file 2>&1

或者

$ command >> file 2>&1

## Here Document

它的基本的形式如下：

command << delimiter

document

delimiter

它的作用是将两个 delimiter 之间的内容(document) 作为输入传递给 command。

*注意：*

结尾的delimiter 一定要顶格写，前面不能有任何字符，后面也不能有任何字符，包括空格和 tab 缩进。

开始的delimiter前后的空格会被忽略掉。

### **实例**

在命令行中通过 wc -l 命令计算 Here Document 的行数：

$ wc -l << EOF

欢迎来到

菜鸟教程

www.runoob.com

EOF

3 # 输出结果为 3 行

$

我们也可以将 Here Document 用在脚本中，例如：

#!/bin/bash

# author:菜鸟教程

# url:www.runoob.com

cat << EOF

欢迎来到菜鸟教程

www.runoob.com

EOF

## /dev/null 文件

如果希望执行某个命令，但又不希望在屏幕上显示输出结果，那么可以将输出重定向到 /dev/null：

$ command > /dev/null

如果希望屏蔽 stdout 和 stderr，可以这样写：

$ command > /dev/null 2>&1

***注意：****0 是标准输入（STDIN），1 是标准输出（STDOUT），2 是标准错误输出（STDERR）。*