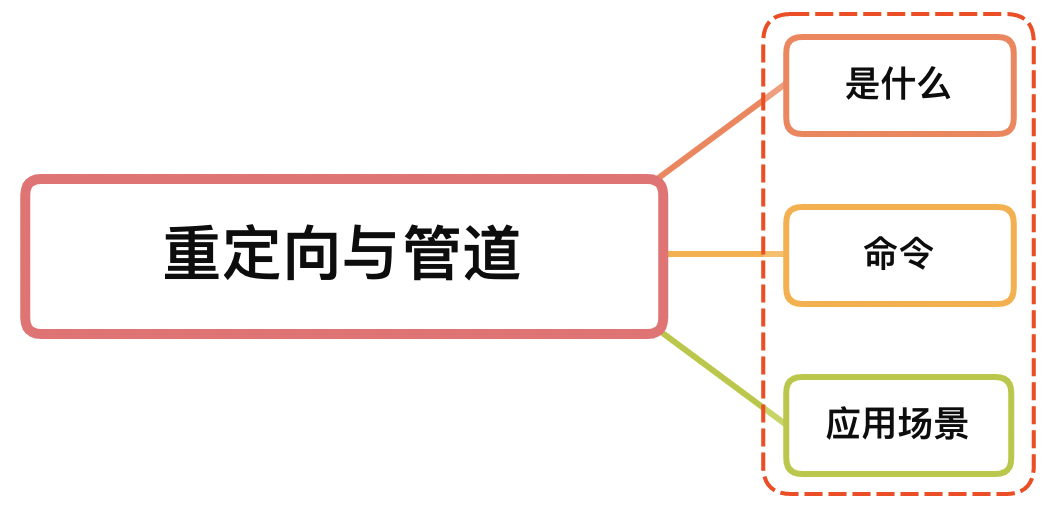
# 面试官：说说你对输入输出重定向和管道的理解？应用场景？

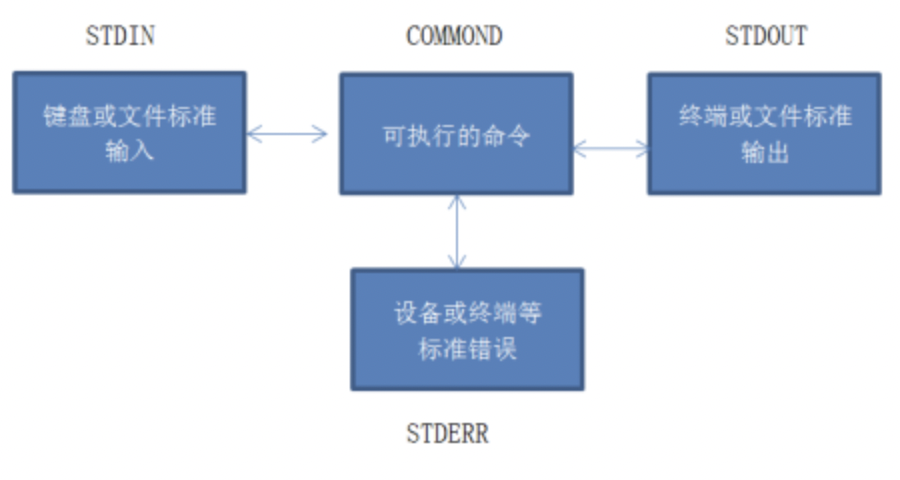


## 一、是什么

linux中有三种标准输入输出，分别是STDIN，STDOUT，STDERR，对应的数字是0、1、2：

* STDIN 是标准输入，默认从键盘读取信息
* STDOUT 是标准输出，默认将输出结果输出至终端
* STDERR 是标准错误，默认将输出结果输出至终端

对于任何linux命令的执行会有下面的过程：



一条命令的执行需要键盘等的标准输入，命令的执行和正确或错误，其中的每一个双向箭头就是一个通道，所以数据流可以流入到文件端（**重定向或管道**）

简单来讲，重定向就是把本来要显示在终端的命令结果，输送到别的地方，分成：

* 输入重定向：流出到屏幕如果命令所需的输入不是来自键盘，而是来自指定的文件
* 输出重定向：命令的输出可以不显示在屏幕，而是写在指定的文件中

管道就是把两个命令连接起来使用，一个命令的输出作为另一个命令的输入

两者的区别在于：

* 管道触发两个子进程，执行 | 两边的程序；而重定向是在一个进程内执行。
* 管道两边都是shell命令
* 重定向符号的右边只能是Linux文件
* 重定向符号的优先级大于管道

## 二、命令

重定向常见的命令符号有：

* > ： 输出重定向到一个文件或设备 覆盖原来的文件

如果该文件不存在，则新建一个文件

如果该文件已经存在，会把文件内容覆盖

这些操纵不会征用用户的确认

* >> ：输出重定向到一个文件或设备，但是是 追加原来的文件的末尾
* < ：用于制定命令的输入
* << ：从键盘的输入重定向为某个命令的输入

以逐行输入的模式（回车键进行换行）

所有输入的行都将在输入结束字符串之后发送给命令

* 2> 将一个标准错误输出重定向到一个文件或设备，会覆盖原来的文件
* 2>> 将一个标准错误输出重定向到一个文件或设备，是追加到原来的文件
* 2>&1：组合符号，将标准错误输出重定向到标准输出相同的地方

1就是代表标准输出

* >& 将一个标准错误输出重定向到一个文件或设备覆盖原来的文件
* |& 将一个标准错误管道输出到另一个命令作为输入

## 三、应用场景

将当前目录的文件输出重定向到1.txt文件中，并且会清空原有的1.txt的内容

ls -a > 1.txt

或者以追加的形式，重定向输入到1.txt中

ls -a >> 1.txt

将标准错误输出到某个文件，可以如下：

$ touch 2> 2.txt  
$ cat 2.txt  
touch: 缺少了文件操作数  
请尝试执行 "touch --help" 来获取更多信息。

通过组合符号将两者结合一起，无论进程输出的信息是正确还是错误的信息，都会重定向到指定的文件里

[root@linguanghui home]# abc &> file.txt  
[root@linguanghui home]# cat file.txt   
-bash: abc: command not found

再者通过管道查询文件内容是否包含想要的信息：

cat test.txt | grep -n 'xxx'

上述cat test.txt会将test.txt的内容作为标准输出，然后利用管道，将其作为grep -n 'xxx'命令的标准输入。

### 参考文献

* https://segmentfault.com/a/1190000020519335
* https://murphypei.github.io/blog/2018/04/linux-redirect-pipe
* https://www.huaweicloud.com/articles/0fb70e8c724ae79f4fc8d676cd6160d3.html