

# 第1章 重定向介绍--全部需要掌握

## 1.什么是重定向

1 | 简单来说，就是把本来输出到屏幕上的信息保存到指定的文件中，这个过程就叫做重定向。

## 2.重定向使用场景

- 1 | 1. 希望将输出的信息保存下来  
2 | 2. 希望将运行的程序的执行结果保存下来  
3 | 3. 希望忽略某些命令的警告或者错误信息  
4 | 4. 希望将错误信息和正确信息保存在不同的文件中  
5 | 5. 将如果程序运行报错了我们希望把报错信息保存下来

## 3.重定向分类

运行一个程序通常会有三种情况，标准输入，标准输出，错误输出

名称	文件描述符	作用
标准输入 (STDIN)	0	接受输入，默认为键盘，也可以是其他文件或命令的输出。
标准输出 (STDOUT)	1	正确的输出，默认输出到屏幕
错误输出 (STDERR)	2	错误的输出，默认输出到屏幕

# 第2章 输出重定向

## 1.什么是输出重定向

1 | 简单来说，就是改变输出内容的位置，比如将输出信息保存到文件中。

## 2.输出重定向分类

名称	操作符	作用
标准覆盖输出重定向	>	将程序输出的正确结果输出到指定的文件中，但是会覆盖原文件的内容
标准追加输出重定向	>>	将程序输出的正确结果追加输出到指定文件中，不会覆盖原文件，而是在后面追加内容
错误覆盖输出重定向	2>	将程序的错误结果输出到指定的文件中，但是会覆盖原文件的内容
错误追加输出重定向	2>>	将程序的错误结果输出以追加的形式输出到指定文件中，不会覆盖原文件，而是在后面追加内容

### 3.输出重定向案例

案例1: 标准覆盖输出重定向

```
1 | echo linux5 > oldboy.txt
2 | echo linux6 > oldboy.txt
3 | cat test.txt
```

案例2: 标准追加输出重定向

```
1 | echo linux6 >> oldboy.txt
2 | echo linux6NB >> oldboy.txt
3 | cat oldboy.txt
```

案例3:错误覆盖输出重定向

```
1 | echooo > oldboy.txt
2 | echooo 2> oldboy.txt
```

案例4:错误追加输出重定向

```
1 | echooo >> oldboy.txt
2 | echooo 2>> oldboy.txt
```

案例5:正确和错误输出到不同的文件

```
1 | useradd oldboy
2 | su - oldboy
3 | find /etc -name "*.conf" 1>ok.txt 2>no.txt
4 | cat ok.txt
5 | cat no.txt
```

案例6:正确和错误都输出到同一个文件

```
1 | find /etc/ -name "*.conf" >> all.txt 2>&1
```

案例7:将错误结果输出到null

```
1 | ls /rooooot 2> /dev/null
```

案例8:将正确和错误结果都输出到null

```
1 | ls /rooooot 2>&1 > /dev/null
```

## 第3章 输入重定向-应用不多

### 1.什么是输入重定向

1 | 输入重定向默认是从键盘里接受输入，我们也可以将命令或文本的内容做为输入源输入给其他程序

### 2.输入重定向案例

案例1:写入多行文本

```
1 | cat > oldboy.txt << EOF
2 | 1a
3 | 2b
4 | 3c
5 | EOF
```

案例2:发送邮件

```
1 | mail -u 526195417@qq.com -t "重金求子" < message.txt
```

## 第4章 管道符号-每天都用

### 1.什么是管道

- 1 | 管道符号为"|"
- 2 | 简单来说，管道就是将多个命令连接起来处理信息，将左侧命令的标准输出做为右侧命令的标准输入。
- 3 | 前面一个命令处理成功的输出信息，交给后面的命令做为输入继续处理。
- 4 | 只要命令不出错，可以接多个管道命令。
- 5 | 错误命令不会传递给管道后面的命令。

### 2.管道应用场景

- 1 | 1.处理日志，比如找出日志中访问排名前10的IP，URL等。
- 2 | 2.过滤输出，比如一个命令输出了10行信息，但是我们只需要其中关键的信息

### 3.管道使用案例

案例1:取出网卡IP

```
1 | ifconfig |grep 10.0.0
2 | ifconfig |grep inet|awk '{print $2}'|grep 100
3 | hostname -I
```

案例2:将/etc/passwd中的用户按UID大小排序并显示前10行

```
1 | sort -t ":" -k3 -n -r /etc/passwd|head -n 10
```

案例3:统计/etc/passwd文件中用户使用的shell情况

```
1 | awk -F':' '{print $7}' /etc/passwd
2 | awk -F':' '{print $7}' /etc/passwd|sort
3 | awk -F':' '{print $7}' /etc/passwd|sort|uniq -c
4 | awk -F':' '{print $7}' /etc/passwd|sort|uniq -c|sort -rn
```

案例4:取出日志里排名前10的IP

```
1 | cat /var/log/nginx/access.log|awk '{print $1}'|sort|uniq -c|sort -rn|head -n 10
```