

管道与I/O 重定向



前言

- 在 Linux 中执行程序，命令行功能最强大的两个效果：就是管道与 I/O 重定向。
- I/O 重定向可将命令行的执行的输出或错误消息重定向至文件，方便当下保存或稍后进行分析。



培训目标

- 学完本课程后，您应该能：
 - 了解程序执行的流(Stream)有哪些
 - 熟悉程序执行的重定向运算符号
 - 掌握多程序搭配管道的用法



目 录

1. 程序执行的输入输出
2. 程序执行的重定向运算符号
3. 管道使用

程序执行的输入输出流

- 标准输入
 - 0: 键盘默认
 - 又称 STDIN
- 标准输出
 - 1: 终端默认
 - 又称 STDOUT

程序执行的输入输出流 (续)

- 标准错误
 - 2: 终端默认
 - 又称 STDERR



目 录

1. 程序执行的输入输出
2. 程序执行的重定向运算符号
3. 管道使用

程序执行的重定向运算符

- > : 将STDOUT重定向到文件
 - 文件内容会被覆盖
 - 举例: `#ls -Ra /etc > /root/backup/config-file-lists`
- >> : 将STDOUT重定向到文件
 - 文件内容会被添加
 - 举例: `$(date ; who -l) >> /root/monitor/who-online`

程序执行的重定向运算符 (续)

- < : 重定向STDIN
 - 将键盘输入改由读入文件提供
 - 举例: `#mail -s "Warning" root < /root/mail-record/alert-notify`



目 录

1. 程序执行的输入输出
2. 程序执行的重定向运算符
- 3. 管道使用**

管道使用

- 运算符管道
 - | : 将一个命令的 STDOUT 发送到另一个命令的 STDIN
 - 举例: `#grep pattern /var/log/messages | mail -s "Issue notify" root`
- 命令行 T 管道
 - tee : 将上一个命令的STDOUT 通过T 管道重定向到文件, 再发送到另一个命令的STDIN
 - 举例: `#ifconfig eth0 | grep pattern | tee /root/interface-info | cut -f2 -d: | cut -f1 -d" "`

管道与重定向

- 比较管道与重定向
 - 标准的命令用法: `grep root /etc/passwd`
 - 重定向: `grep root < /etc/passwd`
 - 管道: `cat /etc/passwd | grep root`
- 三种完全是不同架构, 却得到相同结果

说明文件 - 1

- `bash(1)`
- `cat(1)`
- `find(1)`
- `mail(1)`
- `sort(1)`

问 题

- 要转换执行程序的输出结果，可搭配的运算符号有哪些？
- 请说明执行程序执行的流(Stream)有哪些？



总 结

- 本课程中，我们学习了：
 - 有哪些程序执行的流
 - 熟悉程序执行的重定向运算符号
 - 掌握多程序搭配管道的用法

谢谢

Thank You