

# 重定向与管道

教师: 李军辉

# 标准输入输出

大多数 UNIX/Linux 系统命令从终端接受输入并将所产生的输出发送回到您的终端.

# 标准输入输出

大多数 UNIX/Linux 系统命令从终端接受输入并将所产生的输出发送回到您的终端。

命令举例：

- tr
- ls
- wc
- ...

# 标准输入输出

- 标准输入: 键盘
- 标准输出: 显示屏

# 标准输入输出

- 标准输入：键盘
- 标准输出：显示屏

重定向：改变程序运行的输入源和输出地点。

文件重定向：改变程序运行的输入源为某个文件，输出地点为某个文件。

# 输出重定向

## 两种标准输出

- 标准输出 (STDOUT), 文件描述符 1
- 标准错误输出 (STDERR), 文件描述符 2

输出重定向: >和 >>

# 输出重定向

## 两种标准输出

- 标准输出 (STDOUT), 文件描述符 1
- 标准错误输出 (STDERR), 文件描述符 2

输出重定向: >和 >> 特殊输出文件: /dev/null

写入到它的内容都会被丢弃, 将命令的输出重定向到它, 会起到“禁止输出”的效果.

# 输入重定向

一种标准输入:

- 标准输入 (STDIN), 文件描述符为0

输入重定向: <



# 管道 (Pipe)

- 管道是可以看作是一种特殊的重定向.
- 将一个命令的输出重定向为另一个命令的输入
- 管道的操作符是 |

# 管道 (Pipe)

- 管道是可以看作是一种特殊的重定向.
- 将一个命令的输出重定向为另一个命令的输入
- 管道的操作符是 |
- 格式: 命令1 | 命令2 | ... | 命令n
- 操作符 | 右侧命令必须能够接受标准输入.
- 一般接受标准输入的命令都与文本操作有关, 如 grep, head, tail, more, less, sort, wc 等.
- 也可以是用户定义的接受标准输入的程序.

# 管道使用举例

# 管道使用举例

管道不会使用标准错误输出.

# cut命令

将一行中的一段信息“切”出来, 处理的信息是以“行”为单位.

```
cut -d '分隔字符' -f filds
```

```
cut -c 字符范围
```

- (1) 将PATH变量取出，找出第5个路径.
- (2) 将PATH变量取出，找出第3, 第5个路径.
- (3) 将\$PATH输出的信息取得第12字符以后的所有字符串

- (1) 将PATH变量取出，找出第5个路径.
- (2) 将PATH变量取出，找出第3, 第5个路径.
- (3) 将\$PATH输出的信息取得第12字符以后的所有字符串

```
echo $PATH | cut -d ':' -f 5  
echo $PATH | cut -d ':' -f 3,5
```

```
echo $PATH | cut -c 12-
```

# sort命令

可以依据不同的数据型态来排序.

```
sort [-fbMnrtuk] [file or stdin]
```

选项与参数：

- f：忽略大小写的差异，例如 A 与 a 视为编码相同；
- b：忽略最前面的空格符部分；
- M：以月份的名字来排序，例如 JAN, DEC 等等的排序方法；
- n：使用『纯数字』进行排序(默认是以文字型态来排序的)；
- r：反向排序；
- u：就是 uniq，相同的数据中，仅出现一行代表；
- t：分隔符，默认是用 [tab] 键来分隔；
- k：以那个区间 (field) 来进行排序的意思



(1) 个人账号都记录在 `/etc/passwd` 下，请将账号进行排序。

(2) `/etc/passwd` 内容是以 `:` 来分隔的，我想以第三栏来排序，该如何？

(3) 利用 `last`，将输出的数据仅取账号，并加以排序

(1) 个人账号都记录在 `/etc/passwd` 下，请将账号进行排序。

(2) `/etc/passwd` 内容是以 `:` 来分隔的，我想以第三栏来排序，该如何？

(3) 利用 `last`，将输出的数据仅取账号，并加以排序

```
cat /etc/passwd | sort  
cat /etc/passwd | sort -t ':' -k 3  
last | cut -d ' ' -f1 | sort
```

想要将重复的数据仅列出一个显示

`uniq [-ic]`

选项与参数：

`-i`：忽略大小写字符的不同；

`-c`：进行计数

- (1) 使用 `last` 将账号列出，仅取出账号栏，进行排序后仅取出一位；
- (2) 承上题，如果我还想要知道每个人的登陆总次数呢？

- (1) 使用 `last` 将账号列出，仅取出账号栏，进行排序后仅取出一位；
- (2) 承上题，如果我还想要知道每个人的登陆总次数呢？

```
last | cut -d ' ' -f1 | sort | uniq
```

```
last | cut -d ' ' -f1 | sort | uniq -c
```

计算输出的信息整体数据

**wc [-lwm]**

选项与参数：

- l：仅列出行；
- w：仅列出多少字(英文单字)；
- m：多少字符；

(1) 那个 `/etc/man.config` 里面到底有多少相关字、行、字符数？

(2) 我知道使用 `last` 可以输出登陆者，但是 `last` 最后两行并非账号内容，那么请问，我该如何以一行命令串取得这个月份登陆系统的总人次？

(1) 那个 `/etc/man.config` 里面到底有多少相关字、行、字符数？

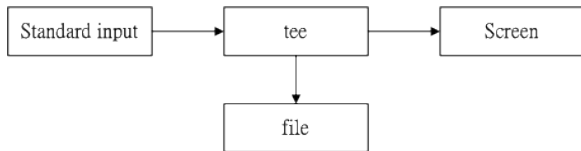
(2) 我知道使用 `last` 可以输出登陆者，但是 `last` 最后两行并非账号内容，那么请问，我该如何以一行命令串取得这个月份登陆系统的总人次？

```
cat /etc/man.config | wc  
last | grep [a-zA-Z] | grep -v 'wtmp' | wc -l
```



➤ 会将数据流整个传送给文件或装置，因此我们除非去读取该文件或装置，否则就无法继续利用这个数据流。万一我想要将这个数据流的处理过程中将某段信息存下来，应该怎么做？

# tee 命令



tee 会同时将数据流分送到文件去与屏幕 (screen) ;  
而输出到屏幕的, 其实就是 stdout

**tee [-a] file**

选项与参数:

**-a** : 以累加的方式, 将数据加入 file 当中!

```
last | tee last.list | cut -d " " -f1  
ls -l /home | tee ~/homefile | more  
ls -l / | tee -a ~/homefile | more
```

tr 可以用来删除一段信息当中的文字，或者是进行文字信息的替换！

```
tr [OPTION] SET1 [SET2]
```

# tr使用1

```
tr [OPTION] SET1 [SET2]
```

当SET1和SET2给出, 并且没有给出-d选项, 替换SET1中的每一个字符为相应SET2中的字符.

```
tr abcdefghijklmnopqrstuvwxyz  
ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

```
tr [:lower:] [:upper:]
```

```
tr a-z A-Z
```

# tr使用2

```
tr '{}' '()' < inputfile > outputfile
```

# tr使用3

```
echo "This is for testing" | tr [:space:] '^'
```

# tr使用4

```
echo "This is for testing" | tr [:space:] ^
```

```
echo "This is for testing" | tr -s [:space:] ^
```



# tr使用5

删除某个字符

```
echo "the geek stuff" | tr -d 't'
```

```
echo "my username is 432234" | tr -d [:digit:]
```

# tr使用5

```
echo "my username is 432234" | tr -cd [:digit:]
```

# tr使用6

仅保留可打印字符

```
tr -cd [:print:] < file.txt
```

# paste命令

**paste** 将两个文件中的每一行合并为一行, 中间以 [tab] 键隔开.

```
paste [-d] file1 file2
```

选项与参数:

**-d** : 后面可以接分隔字符。默认是以 [tab] 来分隔的!

**-** : 如果 **file** 部分写成 **-** , 表示来自 **standard input** 的数据的意思。

# expand命令

将 [tab] 按键转成空格键

`expand [-t] file` 选项与参数：

`-t`：后面可以接数字。一般来说，一个 `tab` 按键可以用 8 个空格键取代。

我们也可以自行定义一个 `[tab]` 按键代表多少个字  
符！

# split命令

将一个大文件，依据文件大小或行数来分割。

**split [-bl] file PREFIX**

选项与参数：

-b：后面可接欲分割成的文件大小，可加单位，例如 b, k, m 等；

-l：以行数来进行分割。

**PREFIX**：代表前导符的意思，可作为分割文件的前导文字。

# split命令

```
split -b 300k /etc/termcap termcap
```

```
ls -al / | split -l 10 - lsroot
```

# 练习

1 **单选** (1分) 下面那些命令可以将当前目录下的名为file.txt文本文件中的所有小写t都替换成大写T。

- ☐ A. `tr 't' 'T' &lt; file.txt`
- ☐ B. `cut 't' 'T' &lt; file.txt`
- ☐ C. `tr 't' 'T' file.txt`
- ☐ D. `cut 't' 'T' file.txt`

2 **单选** (1分) 如何将`ls -l file01 file02` 命令的标准输出重定向到名为stdout.txt的文件中,并将错误输出重定向到名为stderr.txt的文件中

- ☐ A. `ls -l file01 file02 &>stdout.txt`
- ☐ B. `ls -l file01 file02 1>stdout.txt &>stderr.txt`
- ☐ C. `ls -l file01 file02 1>stdout.txt 2>stderr.txt`
- ☐ D. `ls -l file01 file02 2>stdout.txt 1>stderr.txt`



3 单选 (1分) 如何将ls -l file01 file02 命令的标准输出重定向到名为stdout.txt的文件中,并将错误输出重忽略.

- ☐ A. ls -l file01 file02 &>stdout.txt 2>/dev/null
- ☐ B. ls -l file01 file02 1>stdout.txt 2>/dev/zeros
- ☐ C. ls -l file01 file02 2>dev/null 1>stderr.txt
- ☐ D. ls -l file01 file02 &>stdout.txt 2>dev/null

4 单选 (1分) 关于如下命令, 哪个说法是错误的?

```
sort -k3 <<EOF
>002 张三 5500
>005 李四 6600
>009 王五 5120
>EOF
```

- ☐ A. 该命令的输出将对输入内容按字段三排序
- ☐ B. 该命令使用了输入重定向的Here Document功能
- ☐ C. EOF是输入文件的界定符, 是固定的
- ☐ D. 界定符EOF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

5 多选 (4分) 关Linux中的输入输出重定向, 下面哪些描述是正确的?

- ☐ A. 对于任何一条linux 命令,其结果都是通过stdout输出到输出设备上;
- ☐ B. 标准输入重定向是让命令从指定文件中获取输入而非从stdin中获取输入。
- ☐ C. 标准输入流 (stdin) , 向命令提供输入, 文件描述符是 1
- ☐ D. Linux中使用> 和>> 操作符来重定向标准输出

6 多选 (4分) 关于命令echo -e "1 apple\n2 pear\n3 banana">>fruit.txt, 下面哪些描述是错误的。

- ☐ A. 这个命令将echo的输出重定向到了fruit.txt文件中。
- ☐ B. echo的输出重定向将追加到fruit.txt文件最后
- ☐ C. 如果fruit文件不存在, 命令将出错
- ☐ D. fruit.txt文件中如果原有内容将被覆盖

7 多选 (4分) 关Linux中的管道, 下面哪些描述是正确的?

- ☐ A. 就是将一个命令的标准输出重定向为另一个命令的标准输入
- ☐ B. 管道就是一种特殊的输入输出重定向
- ☐ C. 完成复杂任务往往需要构造命令管道
- ☐ D. 所有命令都可以用于构造管道

8 填空 (4分) 显示/etc/passwd文件中的带有root字样的行,并存储到一个名为root.txt的文件中 (提示, 使用cat和grep命令, 请勿要在重定向、管道符号两边加空格, 系统会判错)。

请输入答案

9 填空 (4分) 分页查看ls -al /etc 命令的输出 (请勿要在管道符号两边加空格, 系统会判错)。

请输入答案

10 填空 (4分) 统计当前用户家目录中符号链接文件的个数 (提示, 使用ls、cut、grep和wc命令, 请勿要在管道符号两边加空格, 系统会判错)。

请输入答案