本章內容：

* 標準輸入和標準輸出
* 管道和過濾器

第五章

輸入／輸出重導

標準輸入和標準輸出

UNIX 有許多讀入（read）輸入（例如檔案）和寫出（write）輸出的命令。

通常，如果在命令中沒有輸入檔名的話，那麼按下 RETURN 鍵後，shell 就會去執行你輸入的命令，它將鍵盤作為預設的輸入，你的終端鍵盤就是命令的標準輸入（stardard input）。

當一個命令在被執行後，它的結果通常顯示在終端螢幕上，你的終端螢幕就是命令的標準輸出（standard output）。

所以，在預設值下，每一個命令從標準輸入接收其輸入，而將輸出結果發送給標準輸出。

輸入/輸出的預設值是可以改變的，這種改變被稱為“輸入/輸出重導”(I/O Redirection)。命令一般來說是不接受檔名的，但你可以利用小於符號 "<" 將一個檔案作為一個命令的輸入。例如，下面的命令將 to_do 檔案的內容作為電子郵件的內容發送給 bigboss@corp：

```
% mail bigboss@corp < to_do
%
```

你也可以將命令的結果寫入一個已命名的檔案或者其他設備，而不是將輸出顯示在螢幕上。在這種情況下，你應該使用大於 ">" 符號。管道符號 "|" 將一個命令的標準輸出發送給另一個命令作為它的標準輸入。輸入/輸出重導是 UNIX 最漂亮的功能之一，因為它給 UNIX 帶來了強大的威力和靈活性。

在檔案中放置文字

命令的輸出可以顯示到螢幕，除此之外，你可以將輸入重導到一個檔案內部。當你有多個命令輸出時，在一個螢幕上同時閱讀它們是比較困難的，這時，你可以將這些輸出放置在一起，而得到一個較大的檔案。這個功能在實際中往往很有用處。

前面我們介紹過，cat 命令可以顯示一個短的檔案，它的文字輸出還可以放置在一個檔案中，或者將較小的幾個檔案合併成為一個較大的檔案。

> 運算符號

當你在命令的後面增加 "> filename" 時，命令的結果會作為標準輸出，交給這個已命名的檔案。大於符號 ">" 被稱為輸出重導運算符號（output redirection operator）。

例如，我們可以將這一運算符與 cat 命令一起使用。你平常在螢幕上看到的（來自標準輸出的）檔案內容將交給另一個檔案：

```
% cat /etc/passwd > password
% cat password
root::0:0:Root:/:/bin/sh
daemon:NONE:1:1:Admin:/:
        .
        .
        .
john::128:50:John Doe:/usr/john:/bin/sh
%
```

在第三章“你的 UNIX 工作環境”中我們列舉過一個例子，我們介紹過 cat /etc/passwd 在螢幕上顯示 /etc/passwd 檔案。上面的例子增加了 ">" 運算符號，所以 cat 的輸出被存入工作目錄的 password 檔案中。如果你在螢幕上顯示 password 檔案，那麼你會發現它的內容與 /etc/passwd 檔案是一樣的，這個作業與 cp /etc/passwd password 在效果上是一致的。

不僅僅是 cat 命令可以使用 ">" 重導運算符號，任何命令都可以與之配合，並將命令的輸出文字作為標準輸出。例如：

```
% who > users
% date > today
% ls
password  today users   ...
```

我們將 who 的輸出發送給稱為 users 的檔案，而 date 的輸出發送給稱為 today 的檔案。在列出目錄時，你將會看到這兩個新檔案。現在讓我們看看這兩個檔案中來自 who 和 date 所生成的輸出。

```
% cat users
tim    tty1   Aug 12  07:30
john   tty4   Aug 12  08:26
% cat today
Tue Aug 12 08:36:09  EDT  1999
%
```

你還可以使用 cat 命令和 ">" 運算符號生成一個小的文字檔案。我們在稍早曾提起過，如果你輸入 cat 命令而忘了加上檔名時，可按下 CTRL-D 鍵。這是因為 cat 命令本身將會把你在鍵盤上敲入的任何東西視作輸入。因此，這個命令：

cat > *filename*

將會把你在鍵盤上鍵入的文字重新指向到一個檔案，試試下面這個例子：

```
% cat > to_do
Finish report by noon
Lunch with Xannie
Swim at 5:30
^D
%
```

cat 命令將會把你在鍵盤上鍵入的文字作為輸入，而 ">" 運算符號將把這個輸入重新指向到稱為 to_do 的檔案中。在新的一行鍵入 CTRL-D 將作為文字結束的信號，最後你會得到 shell 提示符號。

你還可以使用 cat 命令和 ">" 運算符號將幾個較小的檔案產生成為一個較大的檔案。如下面的形式：

cat *file1 file2* > *newfile*

它將建立一個 newfile 的新檔案，這個檔案由 file1 和 file2 組成，file1 在前，file2 在後。

```
% cat today to_do > diary
% cat diary
Tue Aug 12  08:36:08  EDT 1999
Finish report by noon
Lunch with Xannie
Swim at 5:30
%
```

警告 在使用 ">" 輸出重導符號時，你應加倍小心，以免意外地被輸出改寫掉一個檔案的內容，你的系統可能讓你將輸出重導到一個已經存在的檔案。如果這樣，原來的檔案將被刪除掉，或者用 UNIX 術語來說，叫作 "clobbered"。一定要注意不要刪除一個你仍然非常需要的檔案！一些 shell 可以使你免受這種風險，例如，在 C shell 中，你可以使用 set noclobber 命令，Korn shell 和 bash 使用 set -o noclobber 命令。你可以在 shell 提示符號下輸入這個命令，或者將這個命令放置在 shell 的啟動檔案中。一旦你做了這樣的設定，那麼 shell 將不再允許你將輸出重導到一個現存的檔案並改寫掉檔案的內容。

當然，此設定完成後對於 UNIX 的其他命令沒有什麼影響，例如 cp 命令仍然可以用來改寫檔案，這種設定只是對 ">" 輸出重導起作用。如果你需要更多的檔案保護，你可設定 UNIX 的檔案存取權限。

>> 運算符號

透過 ">>"（增添重導）運算符號，你可以將更多的文字增加到一個現存的檔案的尾部，而不是改寫檔案原有的內容。">>" 的用法與 ">" 是一樣的，因此：

cat *file2*>>*file1*

將 file2 檔案的內容增加到 file1 檔案的尾部。例如，在下面的例子中，我們將增加 users 檔案的內容，並且將目前的日期和時間增加到 diary 檔案中，然後，我們顯示這個檔案：

```
% cat users >> diary
% date >> diary
% cat diary
Tue Aug 12  08:36:09  EDT 1999
Finish report by noon
Lunch with Xannie
Swim at 5:30
tim     tty1   Aug 12    07:30
john    tty4   Aug 12    08:06
Tue Aug 12 09:07:24  EDT 1999
%
```

管道和過濾器

除了將輸入/輸出重導到一個已經命名的檔案之外，UNIX 可允許你將兩個命令或者更多的命令連接在一起，使得一個程式的輸出成為另一個程式的輸入。這樣連接起來的兩個或者多個命令稱為一個“管道”(pipe)，我們需要在兩個命令之間放置一個豎線符號 "|" 來標明管道。一旦在兩個命令之間生成了管道，那麼管道符號左邊命令的標準輸出將成為管道符號右邊命令的標準輸入。只要一個程式寫出標準輸出，另一個程式則需要讀入標準輸入，那麼在任何這樣的兩個程式之間我們都可以設定管道。

當一個程式從另一個程式取得輸入時，如果對輸入進行某種或者某些動作，然後將動作之後的結果寫到標準輸出（即可以通過管道發送給另一個程式），這一機制稱為過濾器（filter）。最常見的過濾器用法是對輸出進行修改。就跟日常生活中見到的過濾器一樣，UNIX 的過濾器可以過濾掉輸出中一些你不想要的東西。

幾乎所有的 UNIX 命令都可以用來產生管道，下面將介紹幾個用來當過濾器的程式，注意這些程式不僅可以當作過濾器或者管道的一部份來使用，它們在單獨使用時也是非常有用的。

grep

grep 程式搜索一個檔案或者多個檔案，看這個檔案或者這些檔案中是否含有指定類型的行。語法是：

grep *pattern file(s)*

grep 的含意來自 ed（UNIX 文字行的編輯器）的命令 g/re/p。它的意思是全面搜索一個常規表示法並列出所有包含這些檔案的行（globally search for a regular expression and print all lines containing it）。常規表示法（regular expression，或者簡稱 regexp）既可以是簡單的文字（例如一個單字），又或可以是一些用於句型對應的特殊字元。當你熟悉學習常規表示法之後，就可用它來表達非常複雜的文字句型。請參考《Mastering Regular Expressions》一書（Jeffrey Friedl 著，O'Reilly出版）詳細介紹了這方面的內容。

grep 最簡單的用途在於檢查含有某個單字的句型。它可以用在管道中，在這種場合下，輸入檔案中只有那些含有特定字的行才能被發送到標準輸出。 如果你沒有輸入 grep 要讀入的檔名時，它會從標準輸入讀入資料，這就是所有過濾器程式的工作方式：

```
% ls -l | grep "Aug"
-rw-rw-rw-    1  john  doc             11008  Aug   6   14:10   ch02
-rw-rw-rw-    1  john  doc              8515  Aug   6   15:30   ch07
-rw-rw-r--    1  john  doc              2488  Aug  15   10:51   intro
-rw-rw-r--    1  carol doc              1605  Aug  23   07:35   macros
%
```

在這個例子中，我們首先用 ls -l 命令列出你的目錄，ls -l 的標準輸出通過管道發送給 grep，grep 將只輸出含有 "Aug" 的文字行（換句話說，我們得到的結果就是最後一次修改時間是在八月份的所有檔案）。由於 grep 的標準輸出沒有被重導，所以輸出將被顯示在終端螢幕上。（我們前面講過，終端螢幕是預設值下的標準輸出）。

grep 選項可以讓你修改搜尋的方式。表 5-1 介紹了幾個常用的選項。

表 5-1：幾個常用的 grep 選項

選項	含義
-v	列出所有不對應句型的行
-n	列出對應的行和行號
-l	只列出含有對應行的檔名（注意，是字母 l，而非數字 1）
-c	只列出對應行的統計總數
-i	對應大寫或是小寫字母

接下來，讓我們使用常規表示法告訴 grep 尋找含有 carol 在它後面跟著零或其他更多字母的行（用常規表示法縮寫成 ".*"），然後緊接著 "Aug"。關於常規表示法的細節，請參考 Mastering Regular Expressions 一書。

```
% ls -l | grep "carol.*Aug"
-rw-rw-r--    1  carol doc      1605 Aug 23  07:35  macros
%
```

sort

sort 程式將文字行按照字母的順序或數字大小的順序進行排序。在下面的例子中，我們對 food 檔案按照字母的順序進行排序。sort 對檔案本身並不作任何改動，它只是讀入檔案，對檔案排序後，將排序後的文字發送到標準輸出。

```
% sort food
Afghani Crisine
Bangkok Wok
Big Apple Deli
```

```
Isle of Java
Mandalay
Sushi and Sashimi
Sweet Tooth
Tio Pepe's Peppers
%
```

sort 在預設值下，按照字母的順序進行排序。當然還有許多選項可以讓你按照其他形式來控制排序。表 5-2 介紹了幾個選項。

表 5-2：幾個 sort 的選項

選項	含義
-n	按照數字進行排序（例如，2 在前，10 在後），忽略空格和定位格。
-r	顛倒排序的順序
-f	將大寫和小寫一起排序
+x	在排序時，忽略前 x 行

兩個以上的命令可以聯成一個管道，在前面使用了 grep 命令的範例的基礎上，我們還可以進一步地將八月份修改過的檔案按照檔案的尺寸大小進行排序。下面的管道中還包括了 ls、grep 和 sort 命令：

```
% ls -l | grep "Aug" | sort +4n
-rw-rw-r--    1  carol doc              1605  Aug  23  07:35  macros
-rw-rw-r--    1  john  doc              2488  Aug  15  10:51  intro
-rw-rw-rw-    1  john  doc              8515  Aug   6  15:30  ch07
-rw-rw-rw-    1  john  doc             11008  Aug   6  14:10  ch02
%
```

這個管道將你的目錄中八月份修改過的所有檔案，按照檔案的大小進行排序，然後將結果輸出到終端螢幕上。sort 的 +4n 選項將跳過前四個字段（字段由空格隔開），然後對行按照數字的順序進行排序。所以 ls 命令的最後輸出（實際上是 grep 的輸出）將按照檔案的大小進行排序，這裡的 grep 和 sort 對 ls -l 命令的輸出進行修改，所以它們都是當作過濾器來使用的。如果你想將最終的列表用電子郵件寄給某人，還可以在管道的最後增加 mail 命令。當然還可以將 sort 的結果透過管道發送到印表機（例如 lp 或者 lpr）上去，把結果列印出來。

pg 和 *more*

前面我們看到的 more 和 pg 程式也可以作為過濾器來使用，篇幅較長的輸出一般不可能在你的終端螢幕上一次就能看得清楚，使用了 more 或者 pg 過濾之後，每顯示滿一幕就暫停，而接著顯示的是下一幕的內容。

現在，假設你要試試列出一個很長的目錄，例如，你可以使用 cd 命令切換到 /bin 或 /usr/bin 的目錄中，為了更加容易閱讀排序過的列表，我們將輸出通過管道發送 more 命令：

```
% ls -l | grep "Aug" | sort +4n | more
-rw-rw-r--    1  carol doc        1605   Aug  23  07:35    macros
-rw-rw-r--    1  john  doc        2488   Aug  15  10:51    intro
-rw-rw-rw-    1  john  doc        8515   Aug  6   15:30    ch07
-rw-rw-r--    1  john  doc       14827   Aug  9   12:40    ch03
      .
      .
      .
-rw-rw-rw-    1  john  doc       16867   Aug  6   15:56    ch05
--More-(74%)
```

在螢幕上，你首先看到的是一整幕的文字，文字的內容就是一行行按照檔案大小排序的檔名及檔案的其他訊息。當你看完這一幕的內容之後，在螢幕的最底行，你可以看到一個 more 的提示符號，你可以在此輸入命令（參考前面第三章中介紹過的 more 命令選項來操作），以便在排序過的文字中遊走（例如，進入下一頁、往後尋找某個字 /word、結束 more 命令，等等）。

練習題：重導輸入/輸出

在下面的練習中，你將重導輸出，建立一個簡單的管道，使用過濾器來修改輸出。

將輸出重導給一個檔案	鍵入 who > users
對一個命令的輸出進行排序	鍵入 who | sort
將排序結果增添到一個檔案後面	鍵入 who | sort >> users
將結果顯示在螢幕上	鍵入 more users 或 pg users
顯示長輸出到螢幕上	鍵入 ls -l /bin | more 或 ls -l /bin | pg
利用 pr格式化並列印一個檔案	鍵入 pr users | lp 或 pr users | lpr