

## 第 二十二 章

# 系統管理 —— 例行性管理

**在**講過檔案系統之後，現在是時候可以來談談例行的系統管理了。這件工作通常是由一個人負責，即系統管理員，也叫做超級使用者或根使用者。她有絕對的權限，可以存取任何資源。任何一個UNIX系統的成功與穩定，端賴系統管理員的稱職與否。

系統管理員的工作涵蓋整個系統的管理，從維持系統安全性、備份資料、管理磁碟空間，到管理使用者帳號都算是。這些工作必須利用scripts(shell scripts)以自動化的方式週期性執行。而啟動系統服務、或者停止，都是藉由編輯與執行系統組態設定檔。所有的這些，都有賴系統管理員對於系統各個元件的深入瞭解。不過這些工作份量並不會大得無法承受，因為UNIX系統中已提供各式工具協助您能完成上述工作，而且相關文件遠比其他的系統來得齊全。

今天，每個使用者都必須瞭解一些原本僅有系統管理員才會的工作。所以她必須對進階的過濾器指令有點概念，也必須熟悉shell程式寫作。即便您不是具有特殊權限的使用者，您也應該讀這一章。

### 目 標

- 利用su登入成為超級使用者。(22.1)
- 利用passwd更改任意使用者的密碼。(22.2.1)
- 利用date更改系統日期或利用wall和calendar與所有使用者溝通。(22.2.2和22.2.3)
- 利用useradd, usermod和userdel來建立、更動、或刪除使用者資料。(22.3)
- 讓程式藉由設定SUID (set-user-id) 位元，在執行時具有根使用者的權限。(22.4.1)
- 讓目錄可藉由sticky位元的設定，與其他使用者共享。(22.4.2)
- 藉由限制性shell (restricted shell)，限制使用者的活動。(22.4.3)
- 瞭解 init 和 /etc/inittab 在系統啟動和關機時所扮演的角色，並控制系統執行的層級。(22.5和22.6)
- 利用format, fdformat和dd指令，格式化磁片以及複製磁碟。(22.7.1和22.7.2)

- 利用一系列的DOS指令，處理DOS檔案與磁片。(22.7.3)
- 利用cpio備份與還原檔案。(22.8)
- 利用tar備份整個目錄樹，並且附加檔案至現存的備份檔。(22.9)
- 利用du, find和xargs指令，管理磁碟空間。(22.10)

### ► 進階篇

- 利用passwd啟動密碼使用限期機制 (password aging)，強制定期更改密碼。(22.11)
- 瞭解init如何使用rc scripts 來啟動與停止系統服務程式（或系統常駐程式system daemon）。(22.12)
- 瞭解getty如何被執行，並學習設定終端機參數。(22.13)
- 藉由System V的lp系統，瞭解列印的基礎知識，管理印表機以及控制列印排程。(22.14和22.15)
- 利用Linux系統的lpd，進行列印相關管理、設定。(22.15)

## 22.1 root：系統管理員登入

UNIX系統，為系統管理員專用而準備了一個特殊的登入名稱，稱為根(root)。這個帳號不需要額外建立，而是每個系統內建就有的。這個帳號的密碼通常是系統安裝時就已設定妥當：

```
login: root
password: *****[Enter]
# _
```

根使用者的提示符號為#，而不是非特權使用者的\$或%。當您以root登入後，您就會位在root的目錄中。而根據系統的不同，這個目錄可能是/或者/root。

在現今的系統中，大部分的系統管理指令都位在/sbin和/usr/sbin中，不過舊式系統的指令，則是位在/etc下面。此外，root的PATH變數也和一般使用者不同：

```
/sbin:/bin:/usr/sbin:/usr/bin:/usr/dt/bin
```

因為root總是在檔案系統中到處移動，所以很可能她會誤執行了別人所寫的程式或scripts。所以root的PATH變數，刻意不包含現行目錄。



Note

許多專為系統管理員而提供的 shell scripts（尤其關於系統開機設定方面）都是要由Bourne shell來執行。所以您身為一個系統管理員，以 Korn shell來撰寫 scripts 要特別注意；因為別的主機可能沒有此種shell，所以無法執行此 scripts。而且不管在任可狀況下都不能使用C shell script。在Linux上包括一般使用者以及系統管理員都是使用bash，所以這倒不會有什麼問題。

### 22.1.1 su：取得超級使用者的權限

任何使用者如果知道root密碼的話，都可以藉由su取得超級使用者權限。例

如，使用者local（家目錄位在 `/home/local`）在下列情形，就可以變成超級使用者：

```
$ su
Password: *****[Enter]          必須輸入root的密碼

# pwd
/home/local                        提示字串改變，但目錄不變
```

su預期使用者輸入root的密碼。提示符號 `#` 代表使用者已經是超級使用者。雖然使用者代碼（`$LOGNAME`）仍是local，目前目錄也沒有改變，但是local現在已經具有root的權限了。

**建立使用者環境** 使用者因為某個程式停止執行，常會急忙跑去求助系統管理員。而系統管理員會先試著在模擬的使用者環境下，進行測試、模擬。su指令，當伴隨著連字號—使用時，可以重建一個使用者的環境，而不需知道這個使用者的密碼：

```
su henry                          不會提示輸入密碼
```

這個動作會執行henry的啟始設定檔（`.profile`或其他類似檔案），並且為管理員建立henry的使用者環境。因為su是在子shell下執行，所以可以按 `[Ctrl-d]` 或利用`exit`離開這個模式。

超級使用者也可以不須登入另一個使用者的帳號，直接在這個使用者的環境下執行程式，利用選項 `-c`。例如，您可以藉由root執行`dbstart`指令，啟動甲骨文資料庫（Oracle database），即使此指令通常要由oracle使用者親自執行：

```
su oracle -c dbstart
```

您可以將這些指令放在系統起始 `scripts` 中，以便資料庫可以在系統開機時自動被啟動。

## 22.2 系統管理員權限

超級使用者擁有系統至高的權限，有些指令只有她才能執行。有些指令則是在系統管理員執行時，會有不同的結果。概略來說，系統管理員的權限主要有：

- 更改任何檔案的屬性，包括它的內容、權限、甚至檔案擁有權。即使有防寫權限，她也能用`rm`移除任何檔案。
- 啟動或殺掉任何程序。除了一些系統執行必要的程序之外，系統管理員可以直接殺掉所有程序。而僅有超級使用者，可以執行`shutdown`指令將系統關機。
- 不需事先知道先前的密碼，就能更改使用者的密碼。
- 設定系統時間。

- 同時向所有使用者傳送訊息。
- 限制使用者所能建立的檔案最大長度。
- 控制使用者執行排程工具（at和cron）的權限。
- 控制使用者存取TCP/IP服務（rlogin, rcp和rsh）的權限。

因此您可以理解，為什麼系統管理員必須謹慎地使用這些權限。如果因為無心或者系統的漏洞，導致權限落在一個壞心眼的人士手上，那就是大悲劇發生的時候了。我們已經討論過使用者存取排程工具以及TCP/IP服務的情形，所以接下來的章節中，我們會討論剩下的部分。

### 22.2.1 passwd：任意更改密碼

當一普通使用者執行passwd時會被提示輸入舊密碼（3.1）。不過，當root執行這個指令時，情況就沒有那麼嚴苛：

```
# passwd
Changing password for root
Enter the new password (minimum of 5, maximum of 8 characters)
Please use a combination of upper and lower case letters and numbers.
New password: *****
Re-enter password: *****
```

**要輸入兩次以確認**

Password changed.

注意，更換密碼時並不需要輸入舊密碼。系統管理員不僅要好好保護這個密碼，還要默記在心；否則的話，整個系統可能就要重新啟動了。UNIX系統中，允許root可以不需事先知道舊密碼，即可任意設定每個人的密碼：

```
passwd henry
```

同樣的，不須輸入舊密碼，但要輸入兩次新密碼。

今日常常看到駭客意圖入侵系統事件的發生。使用者常常將密碼告訴別人，導致系統安全性的危機。而密碼總是隨著時間越來越多人知道。更糟的是，使用者很懶得常常更換密碼。為了因應這種情況，passwd指令提供了一個功能，可以強迫使用者固定時間要更換一次密碼。這會在「進階篇」中再討論（22.11）。



**Note**

當使用者是root時，passwd指令不會提示要輸入舊密碼，甚至在更改root密碼時也是如此。

### 22.2.2 date：設定系統時間

我們之前是「被動地」使用date這個指令（3.9），亦即僅能顯示系統日期。同樣的一個指令，在root手中就可以利用特殊參數來設定系統日期。這個參數通常是八個字元的字串，格式為「月月日時時分分」，也可以再加上二字元或四字元的年份字串：

```
# date 06010735
Fri Jun 1 07:35:00 EST 2000
```

UNIX系統的年份排列，應都是正確無誤（至少到2038年都還可以）。但系統管理員要確保日期時間精確，因為cron排程器(10.13)是根據系統時間動作。



Note

系統時間僅有系統管理員可以更改。

### 22.2.3 wall與calendar：和一般使用者溝通

wall 指令wall可以同時向所有使用者廣播。大部分的系統不允許一般使用者執行這個指令（Linux除外），只有root可以執行：

```
# wall
The machine will be shut down today
at 14:30 hrs. The backup will be at 13:30 hrs
[Ctrl-d]
```

所有在線上的使用者都會收到這個訊息。指令wall會忽略mesg的設定。這個指令root經常有機會使用，尤其是在系統關機前。

calendar 您已經見過calendar的用法(3.11)作為提醒使用者的服務工具。若以系統管理員身份來執行，可以加上一個參數(-)來提醒所有使用者有關她們原本的行程：

calendar - *讀取所有人的calendar檔案*

calendar會讀取每個人的calendar檔案，並將輸出寄給每個人。這個指令最好放在系統管理員的個人設定檔(/.profile)，或者放在系統開機執行的scripts中。更好的方式，則是讓cron在每天的固定時間執行它。

### 22.2.4 ulimit：設定檔案長度限制

有問題的程式或程序，可以在很短時間內佔掉所有磁碟空間。指令ulimit可以用來限制一使用者能建立的檔案之最大長度。當不加參數時，ulimit會顯示目前的設定：

```
# ulimit
2097151 Linux和Solaris會顯示“unlimited”
```

預設的上限單位是512（或1024，在某些系統下）位元組，這是在kernel中預先設定值。一般使用者僅能降低這個上限，只有系統管理員可以提高這個上限：

ulimit 20971510 *檔案最大長度增加10倍*

您可以將這行指令放在/etc/profile，以便讓這個指令對所有人生效。這個指令

對現在所有的shell來說都已經是內建了。不過內建之後還多了許多選項，我們並不會在本書中深入研究。

## 22.3 管理使用者帳號

UNIX系統中的「使用者」這個名詞不只是代表一個人，也可能代表一個計畫或一個應用程式。同一群任務類似的人，可以用相同的使用者名稱來使用系統。所以就會有類似marketing, accounts, mis這樣的使用者名稱出現。UNIX系統提供了三個指令來建立及管理使用者帳號，包括useradd, usermod以及userdel。

當建立一個使用者帳號時，您必須為她設定一個群組。一群組中通常有許多個使用者，每個人的權限不同。在同一個計畫下工作的成員，彼此都必須能夠讀取其他成員的檔案，所以要將她們設為同一個群組。指令 chmod 和 chgrp 可以更改檔案的群組權限。

建立一個使用者帳號要設定以下的資料：

- 使用者代碼 (UID) 以及使用者名稱。
- 群組代碼 (GID) 以及群組名稱。
- 家目錄。
- 登入shell。
- 在 /var/mail下設定信箱。
- 密碼。

這些參數大部分可以在 /etc/passwd中對應使用者的那行中看到。早期系統管理員必須自行建立家目錄以及信箱，不過現在已經有指令可以自動做這些事。現在我們將建立一群組給一使用者，並將使用者新增至系統中。

### 22.3.1 groupadd：新增一群組

如果要新增一個群組，就要在 /etc/group檔案中新增一欄位。一個使用者一定有一個主要群組，有可能會屬於其他次要的群組。這個檔案包含了系統上所有已經被命名的群組，以下是摘錄的一小段內容：

```
root:x:0:root
bin:x:1:root,bin,daemon
lp:x:7:
uucp:x:14:uucp,fax,root,fnet,sumit
users:x:100:henry,oracle,image,enquiry
pppusers:x:230:
```

*GUID為230*

每一行包含四個欄位，我們以users這個群組為例：

- 第一欄 (users) 顯示群組名稱。這個名稱也會顯示在檔案列表中群組的欄位。
- 第二個欄位可能空白或是個x。這一欄原本儲存群組密碼，但現在已經不用了。

- 第三欄位顯示user這個群組的群組代碼 (GUID) (這裡為100)。使用者若屬於同一群組則她們的GUID相同。
- 最後一欄是以分號隔開的數個使用者名稱 (henry,oracle,image,enquiry)。對這些使用者來說users是她們的次要群組。這一欄若空白，並不代表沒有人屬於這個群組。

要建立一新的群組dba，群組代碼241，您必須利用groupadd指令：

```
groupadd -g 241 dba 241為GUID
```

這個指令會將資料鍵入 `/etc/group`中，您也可以手動建立：

```
dba:x:241:
```

一旦這個群組建立，您就可以將使用者加入這個群組了。



Note

除了groupadd以外，您也可以用groupmod和groupdel來修改或刪除群組。

### 22.3.2 useradd：新增一使用者

指令useradd可以新增使用者至系統中。所有必須的參數，都要加在指令之後：

```
# useradd -u 210 -g dba -c "THE RDBMS" -d /home/oracle -s /bin/ksh -m oracle
# _
```

上列這個靜悄悄的動作，已經建立了使用者名稱oracle、UID為210、群組名稱為dba；家目錄為 `/home/oracle`、所使用的shell為Korn shell。選項 `-m` 確保家目錄會被建立，並且複製基本的 `.profile`和 `.kshrc`到家目錄中。上述的指令行 useradd 會將資料寫入 `/etc/passwd`，見圖22.1。

接下來您必須設定這個使用者的密碼，利用passwd oracle。當這些步驟都完成之後，這個帳號就可以登入了。

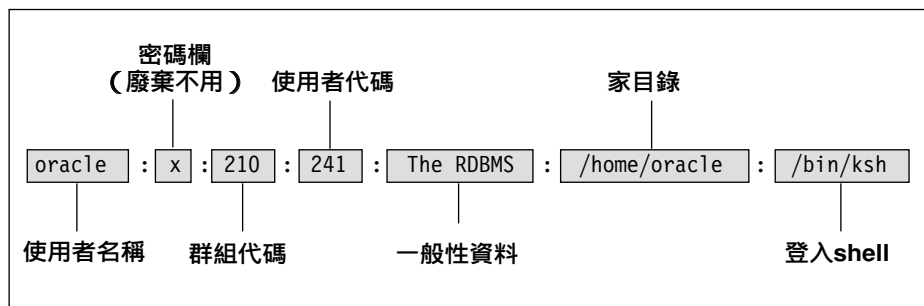
### 22.3.3 /etc/passwd與/etc/shadow：使用者設定檔

所有的使用者資料（除了密碼）都儲存在 `/etc/passwd`中。它曾經儲存過密碼，這也是為什麼這個檔案被如此命名的原因。編碼過的密碼資料，則儲存在 `/etc/shadow`中。這個檔案才是現在被 passwd 真正用來核對使用者輸入的密碼正確與否的密碼檔。

以 `/etc/passwd`中的oracle這一行為例。有七個欄位，意義依序如下（依據出現在 `/etc/passwd`檔案的順序）：

- 使用者名稱 登入系統所使用的名稱 (oracle)。

圖22.1 /etc/passwd中的一行



- 密碼 僅包含x，不包含真正加密的密碼。
- 使用者代碼 使用者數值代碼為 210。每個使用者的代碼都不能相同。
- 群組代碼 使用者所屬的群組代碼為241。這也是 **/etc/group**中第三個欄位的數值。如ls的指令會讀取這個檔案來顯示群組名稱。
- 註解或一般性資料 使用者詳細資料，例如名字、地址等等（此處為The RDBMS）。這個名稱會被加在電子郵件位址之前。任何由這個使用者寄出的信件都會顯示寄件者為 “The RDBMS” oracle@planets.com，在此假設使用者屬於planets.com這個網域。
- 家目錄 使用者登入時所在的家目錄 (/home/oracle)。
- 登入shell 使用者登入後第一個執行的程式。通常預設為 /bin/ksh、/bin/csh等。

雖然這個檔案正常只有系統管理員可以修改，但使用者也可以利用chsh來修改自己所使用的 shell (17.1)而不用與系統管理員聯絡。



Note

檔案 **/etc/passwd**的最後一欄，實際上是使用者一登入時所執行的程式。這通常是 shell，不過系統管理員可能會選一個不同的程式，來限制這個使用者的操作（比方 pppd）。

對 **/etc/passwd**的每一行來說，在 **/etc/shadow**都有相對的一行。上述的例子相對應的一行可能為：

```
oracle:$1$07VbeHwq$0qHK0W73boShhr093.txp.:10846:-1:99999:-1:-1:-1:135365660
```

編碼過的密碼列在第二欄。理論上是不可能由這一串字來推算出原先的密碼。然而，一個聰明的駭客可以使用密碼編碼的演算法來自己試著產生一組。這類演算法在網路上到處都有提供。而很有可能他剛好猜對，產生一組一模一樣的密碼編碼，所以這個檔案務必不可讓一般使用者讀取。



### 22.3.4 usermod與userdel：修改及刪除使用者

usermod可以用來修改一些使用者資料（利用useradd建立）。有可能使用者想要更改登入shell，下例將oracle的shell改為bash：

```
usermod -s /bin/bash oracle
```

*如果無法執行chsh就用這一行指令*

要刪除使用者，可以利用userdel。下例從系統中移除了使用者oracle：

```
userdel oracle
```

*並未刪除使用者的檔案*

這會將 `/etc/passwd`、`/etc/group` 和 `/etc/shadow` 中有關oracle的資料刪除。但使用者的家目錄尚未被移除，管理者需視情況分開去移除或保留。

## 22.4 確保安全性

安全性的考量，主要就是在於保護檔案的安全，所以檔案的權限設定要十分小心，否則就會輕易的被不法的使用者利用而造成無法補救的後果。身為一個系統管理員，您必須確保系統檔目錄（`/bin`、`/usr/bin`、`/etc`、`/sbin`等等）以及其下的檔案不能被其他人刪除。我們現在來討論一些與UNIX系統安全性有關的功能。

### 22.4.1 Set-User-Id 暫時的權限

許多系統程式，允許使用者更新重要的系統檔資料，例如 `/etc/shadow`，雖然它們無法直接使用編輯器更改。這是可能的因為這些程式擁有特別的權限模式，可以在程式一被執行時，讓使用者暫時得到該執行檔所有者的權限。passwd程式就是如此：

```
-rwsr-xr-x 1 root shadow 34808 Nov 30 17:55 /usr/bin/passwd
```

權限欄位中的字元**s**是一個特殊的模式，稱為設定的使用者代碼（set-user-id），又稱SUID。當普通使用者執行passwd時，這個程序有效的使用者代碼已經轉變為root，即程式的擁有者。這種SUID權限，就可以讓 passwd 程式修改 `/etc/shadow`。當這個程序結束後，這種權限也跟著消失了。

任何檔案的SUID都可以被設定，不過只能經由root以指令chmod加上特殊參數：

```
# chmod u+s filex ; ls -l filex
```

*如同chmod 4711*

```
-rws--x--x 1 root bin 113 Mar 24 11:18 filex
```

這個SUID是個潛在的安全性漏洞。當一個使用者執行了這樣的檔案且擁有者為root，他就突然擁有root的權限（他自己可能也不知道）。身為系統管理員，您必須隨時注意所有root擁有的SUID程式，可能被普通使用者建立或複製。這可以用指令

find來加以尋找：

```
find /home -perm -4000 -print | mail root
```

額外的八進位數值 (4) 代表SUID模式，不過此時 find 將 000視為比對至任意的權限組合。您也可以利用 cron 做定期性的檢查並郵寄檔案列表給root。

這個SUID的機制是由Dennis Ritchie所發明，也是UNIX唯一有專利保護的部分。另外還有set-group-id (SGID)，類似SUID，不同的是具有SGID的程式，可以讓使用者執行時取得擁有這個程式的群組的權限。SGID的八進位值為2。



Note

第四個位元，是檔案處在特殊模式下才會被設定的。對SUID來說值為4，對SGID來說則是2，對sticky位元來說則是1。剩下的三個位元意義和原來的權限設定意義相同。

### 22.4.2 Sticky位元

在系統負載較大的時候，程式可能會由記憶體中被交換 (swap) 到磁碟上，然後當需要時再重新由磁碟載回。sticky位元用來確保交換的動作只會發生一次，並且程式的磁碟映像檔 (image) (10.2) 會一直存在交換區域中。這可由檔案權限欄位中的字元t得知（第四個八進位數，數值為1）：

```
-rwxr-xr-t  2 root    root      2878448 Sep 25  1999 /usr/bin/emacs
```

這表示只要您使用emacs一次，它就會停留在交換區域中，直到機器關機為止。這也就是為什麼在舊機器上執行編輯器時，第一次要花費較久的時間，之後就比較快些。不過，這個設定已經快要不具意義了，以前主要是用於機器磁碟速度較慢，記憶體又不足的情形下。現在新的機器磁碟機速度又快，記憶體又多，所以一般檔案並不需要這個設定位元。

設定一目錄的Sticky 位元 然而當這個sticky位元配合目錄使用時，就成了一個相當有特色的安全性設定。UNIX系統允許每個使用者在 /tmp和 /var/tmp下面建立檔案，但每個人只能刪除屬於自己的檔案，為什麼呢？因為這兩個目錄被設定了sticky位元：

```
# ls -ld /tmp /var/tmp
drwxrwxrwt 15 root    root      6144 Nov 28 22:26 /tmp
drwxrwxrwt 37 root    root      1024 Nov 27 23:46 /var/tmp
```

這些目錄都可以被寫入，不過額外的字元t可以確保henry不能移除romeo的檔案。

利用chmod，您也可以為目錄設定這個位元，不過要額外加個1：

```
# chmod 1775 bar
# ls -l bar
drwxrwxr-t  2 sumit   dialout   1024 Apr 13 08:25 bar
```

這個sticky位元非常適合應用在專案管理上。要讓一群使用者可以工作在一組共用檔案上，但卻無須擔心安全性問題，您必須這樣做：

- 為這些使用者在 `/etc/group` 建立一個共用的群組名稱。
- 為各個使用者建立自己的帳號名稱，但共用同一個家目錄。
- 確認家目錄以及其下的所有子目錄，不屬於任一個使用者。利用 `chown` 將之它們更改為屬於 `root`。
- 將這些目錄設為任何人可寫入，同時設定這些目錄的sticky位元，利用 `chmod 1775`。

在這樣的安排下，這個群組裡的每個使用者都有寫入權限，也能夠建立自己的目錄和檔案。但也只能刪除這些屬於自己的檔案。這是個很有用的特色！



Tip

將一個屬於同一群組中所有使用者可分享的目錄，設定sticky位元。如此每個使用者的檔案，就不用擔心會被別人更動。但注意這個目錄，不能屬於這群組中的任一人。

### 22.4.3 限制性Shell

要限制一個使用者的行為，您必須在建立這個使用者帳號時，設定使用限制性shell。這個shell以前叫做 `rsh`，不過現在 `rsh` 指的是遠端執行程式的指令。現在較好的shell都有限制性的版本出現，如 `rbash` 和 `rksh`。這必須在 `/etc/passwd` 最後一欄明確指定。使用限制性shell的使用者，會被限制而無法做以下事項：

- 使用 `cd` 指令，意思是她不能變更目錄。
- 重新定義 `PATH` 變數，所以也不能存取其他目錄的指令。
- 重新定義 `SHELL` 變數，所以也無法更改成非限制性shell。
- 使用包含 `/` 的路徑名稱，意思是無法以絕對路徑或相對路徑表示法執行一指令。
- 使用 `>` 和 `>>` 運算子來建立或附加檔案。

在這樣的環境下，此使用者只能執行存在於一個全新且無法變更之 `PATH` 變數下的指令。這通常被設定成只包括目前目錄。如果使用者要執行位在 `/bin` 或者 `/usr/bin` 下的指令，則要將這些指令在限制的目錄下設定好鏈結。

某些指令提供跳回shell的功能（例如 `vi` 和 `mail`），而且有些版本的UNIX系統可以讓您在這樣跳脫的shell下，利用絕對路徑名稱執行指令。在限制的使用者環境下，要確保這些指令不能這樣動作，或者禁止使用這些功能。



Tip

如果您的系統上沒有限制性shell的程式，可以使用標準的shell加上選項 `-r`（例如 `sh -r`，`bash -r` 或 `ksh -r`）。不過您無法這樣寫進 `/etc/passwd`，所以只好先執行一普通的shell，然後利用 `exec` 從啟始設定檔中執行限制性shell。另外也要確保 `PATH` 變數內容中，僅包含一個目錄。

## 22.5 開機

系統的開機和關機程序，是由幾乎不修改的自動化 scripts 來控制執行。然而系統管理者必須詳細瞭解這兩個動作的精確步驟。如果開機不成功時，她才有能力處理。

### 22.5.1 init：主要的發動者

當系統開機時有許多程序被初始化。核心程式（`/stand/unix`，`/kernel/genunix`或`/vmlinuz`）被載入記憶體，然後由它進一步衍生其他的程序。最重要的程序就是`/sbin/init`，其程序代碼PID為1，用來產生接下來的所有程序。所以您必須瞭解 `init`的詳細行為模式；有下面兩點重要的理由：

- 它決定執行的層級（系統狀態），並決定在每個執行層級上什麼程序可以執行（或刪除）。
- 它衍生`getty`程序在每個終端機以及數據機埠上，所以使用者才能登入。

`init`同時也確保系統常駐程式能被執行。`lpsched`用來監控印表機的列印工作排程佇列。`cron`是系統的計時器。`httpd`是全球網伺服器的服務程式，而`sendmail`控制進出的信件。然而，`init`是這些程序的父程序（有些甚至是祖父程序）。

### 22.5.2 init執行層級 (Run Levels)

`init`負責讓系統處在不同的狀態下，稱之為執行層級。執行層級數值一般是個位數的數字（0到6），或者是個`s`或`S`。在不同的層級上，有不同的程序被執行。正常狀態下，系統會處在以下層級之一：

- 0 系統關機。
- 1 系統管理模式（僅掛上本地端的檔案系統）。
- 2 多人使用模式（不提供NFS）。
- 3 完整的多人使用模式。
- 6 關機重開模式。

`s`或`S` 單人使用模式（掛上基本的檔案系統）。

我們不考慮層級4和5，因為不是廢置不用就是沒有特殊用途。當系統開機時，`init`會先進入層級1，即系統管理模式。此時您還無法使用印表機或終端機，因為系統常駐程式還未經啟動。

單人使用模式，是系統管理員用來執行管理工作的，比方執行離線資料備份。系統層級1的行為，根據不同的系統而有所不同，在某些系統上1和`S`是一樣的。

根據不同的系統，正常的多人使用模式可能會在2或者3。您可以利用`init`加上系統層級當參數來改變不同的層級：

```
init 2
init 3
```

切換至多人使用模式

如果您仔細察看開機訊息（或者 `/etc/inittab`），您可以得知您的系統的多人模式層級為何。



Tip

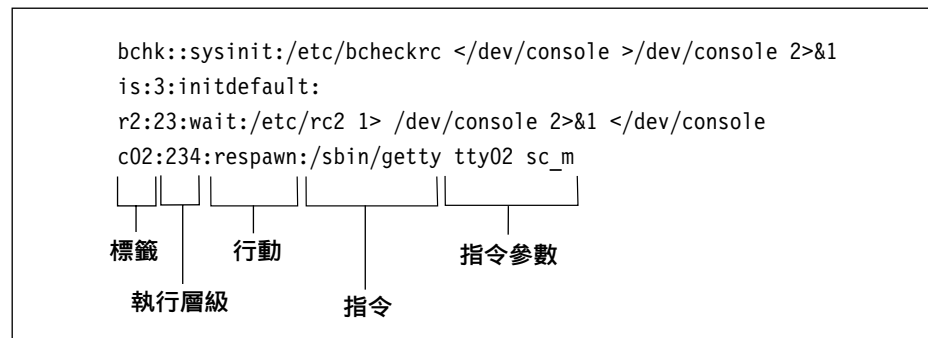
要知道您所在的執行層級為何，可以利用 `who -r` 指令。Linux 的使用者可以用 `run-level` 指令。

### 22.5.3 /etc/inittab：init 啟始設定檔

`init` 的行為由 `/etc/inittab` 所控制。當 `init` 被呼叫時，它只會讀取一次這個檔案的所有內容。檔案的內容決定了在不同執行層級時，所要衍生的程序以及要執行的程式和特定的通訊埠。我們來看一下這個檔案的部分內容（圖22.2）。

所有開機歷程中初始化設定的項目，都是由此控制。標籤僅是用來辨認這一行，並不具特別涵意。第二欄位顯示指令可執行的層級。動作和指令則是最末兩個欄位（如果我們將指令參數視為指令的一部分的話）。

圖22.2 /etc/inittab 檔案



現在我們來針對其中一行仔細討論。標註為 `r2` 的這行，意指的是：「當位在執行層級2或3時，執行 `/etc/rc2` 這個程式，並等待其完成之後才繼續下一行。」所有的輸入、輸出以及錯誤訊息都指示從控制台（console）而來或送去給控制台。我們等會再討論其他行的意義。

當 `init` 加上某一層級參數來執行時，它會讀進所有的符合執行層級的行列，然後依照設定的指令順序去執行。而如果執行層級欄位為空白（此處的第一行），代表所有層級都必須執行這個指令。藉由讀取包含 `initdefault` 動作的資料，`init` 也可以得知預設的執行層級為何。所以您可以知道，這個系統預設是開機成為執行層級3。

`initdefault` 和 `wait` 是兩個 `init` 所認得的動作。除此之外還有其他的：

- **sysinit** 用來初始化整個系統。系統會檢查檔案系統的「乾淨度」，啟動交換分割區 (swap partition)，並設定主機名稱。它可能會等候系統管理員輸入。
- **respawn** 確保一程序在終止重新時會啟動。這對getty程序是必須的。
- **once** 執行此程序一次，不用等待其完成。
- **boot** 當inittab第一次被讀入時，執行此程序。init會忽略這裡的執行層級欄位。
- **bootwait** 同上，但等待其完成。
- **off** 刪除此程序，如果它仍在執行中。
- **ctrlaltdel** 執行shutdown這個指令（Linux適用）。

使用telinit q 身為系統管理員，您也可以新增或修改 /etc/inittab。您可以修改預設的系統執行層級、新增欄位，以啟動新的終端機或數據機等等。不過之後您需要使用telinit來告訴init重新讀取組態設定檔：

telinit q

或者init q

init和telinit是鏈結檔，所以telinit可以經由ln指令隨時被建立（假如它不存在）。您也可以使用init q。



Note

您可以藉由察看 /etc/inittab 檔案中包含initdefault的那行，得知系統預設的執行層級。

## 22.6 關機

系統管理員也負責系統的關機工作。指令shutdown可以作這件事。shutdown通常執行以下的動作：

- 利用wall指令廣播，通知使用者系統即將關機。讓使用者有時間準備結束工作離開。shutdown本身在傳送第一個訊息給使用者之後則先等候一分鐘不動作，然後可能會再送出一或二個提醒的訊息。
- 送出訊號給所有執行中的程序，讓它們正常結束。
- 將所有使用者登出，殺掉遺留的程序。
- 卸下所有次要的檔案系統。
- 將檔案系統資訊寫入磁碟中 (21.5.4) 以確保檔案系統的完整性。
- 通知使用者將重開機，或將系統切換至單人模式。

當螢幕出現以下的訊息時

```
Reboot the system now or turn power off
System halted
```

出現在控制台，可以視為系統已經成功的完成關機程序了。您可以關掉電源、或者重開機。

選項 **-g** 可以覆蓋預設的一分鐘等候時間。這個指令可以有列用法：

```
shutdown -g2                                兩分鐘後關機
shutdown -y -g0                             立刻關機
shutdown -y -g0 -i6                         關機後重開（執行層級6）
```

因 `init` 可以使用執行層級 **0** 和 **6** 來關機，所以您也可利用 `init` 來關機；不過這和 `shutdown` 比起來是比較不適當的方法：

```
init 0                                       關機
init 6                                       關機重開
```



Note

某些系統，比方 `Solaris`，另提供 `reboot` 和 `halt` 指令，可以直接關機而不警告使用者。另外還有一個 `haltsys`，可以立刻關閉系統。因此除非您確知自己在做什麼，否則建議您在管理多使用者的系統時，還是使用 `shutdown` 來關機。



Linux  
注意事項

按鍵 `[Ctrl][Alt][Del]` 也可以用來關機。每個 Linux 系統中的 `inittab` 一定會包含這樣一行：

```
ca::ctrlaltdel:/sbin/shutdown -t3 -r now
```

Linux 也支援選項 **-t** 來覆蓋預設的一分鐘等候時間。因而 `shutdown` 指令，也可以這樣使用：

```
shutdown 17:30                             在17:30時關機
shutdown -r now                             立刻關機並重開機
```

### 22.6.1 允許某一使用者關機

指令 `shutdown` 只有 `root` 可以執行。然而，當她不在時，也可以授權另一位使用者來執行關機的任務。這表示她必須建立另一個類似 `root` 的帳號。但大部分時候，這個使用者帳號不能被允許進入 `shell`。有個方法可以解決這樣的問題。

超級使用者的權力主要來自於檔案 `/etc/passwd` 中的一個簡單欄位。只有 `root` 的使用者代碼是 **0**。這個技巧是可以先建立一個普通的帳號（比方 `shut`），利用 `useradd`：

```
useradd -u 210 -g users -s /bin/sh -d /home/shut -m shut
```

現在您必須授權給這個使用者 `root` 權限，藉由修改 `/etc/passwd` 中的 `UID`，由 **210** 變為 **0**。因為 `useradd` 中預設不允許 `root` 的 `UID` 被重複使用，您必須自己手動編輯這個檔案。現在，將 `shutdown` 指令放進 `shut` 的 `.profile` 中（最好以 `exec` 執行），如此 `shut` 便不能作任何其他事。這個用法，並沒有出現在正式的文件上，但在我們的 UNIX 系統上測試相當正常。而您也可以將 `GID` 設為 **0**。

這樣做的問題，在於如果某人想要偷用root的帳號，他可能也會花些力氣才能真正取得權限。而除非將UID改回來（或改密碼），root並沒有什麼辦法控制這個帳號。

## 22.7 管理軟碟機存取

雖然磁帶是最常用的備份工具，軟碟機在桌上型電腦上卻早已廣泛使用。磁片是極為方便的工具，可以用作不同機器間的檔案交換。在本書中我們以3.5吋1.44 MB的磁片做為例子。

### 22.7.1 format與fdformat：磁片格式化

在您使用磁片之前，必須先加以格式化。這可以用指令format或fdformat（看您的系統上提供哪一個）加上設備名稱作為參數：

```
format /dev/rdisk/f0q18dt System V
```

這個指令會格式化1.44 MB磁片。System V是透過原始資料裝置(raw device)來格式化。格式化的過程，也同時會檢查磁區表面以找出可能遇到的錯誤。

format指令在Solaris系統上，是用來建立硬碟分割區。所以Solaris上是使用fdformat來格式化磁碟片：

```
# fdformat
Press return to start formatting floppy.
```

選項 **-d**代表使用DOS的格式。



Linux  
注意事項

Linux也使用fdformat來格式化軟碟片。而設備名稱必須明確指定：

```
fdformat /dev/fd0H1440
```

### 22.7.2 dd：複製磁片

dd(disk dump)是個多用途的程式。不過，因為許多其他的UNIX工具取代了它的過濾器功能，所以似乎是有些過時了。它可以被任何使用者呼叫，不過它的確是個系統管理工具。它的指令行參數格式有點奇怪，格式為選項 = 參數值。

dd曾經廣泛地使用在檔案系統的複製，不過現在它僅被用來複製磁片或磁帶。它是非互動式的指令，並且dd的參數要成對才能執行。

現在我們用dd來複製一片1.44 MB軟碟片。第一步是建立軟碟映像檔：



用到的關鍵字為 **if** = (輸入檔案名稱), **of** = (輸出檔案名稱), 以及 **bs** = (區塊大小)。上面的指令會複製 1.44 MB 軟碟片的內容, 存成一暫存檔 **\$\$**, 區塊大小為 147456, 是 1.44 MB 磁碟片容量的十分之一。

接著, 將來源磁片抽出, 換上另一片已格式化的空白磁片。接著執行類似的 **dd** 指令, 將暫存檔複製到磁片上:

```
# dd if=/dev/rdisk/f0q18dt of=$$ bs=147456
10+0 records in
10+0 records out
```

您應該以這種方式來複製開機磁片。同樣的方式, 您也可以複製磁帶, 但是如果系統上有兩個磁帶機, 則只需單一個 **dd** 指令就行:



Note

**dd** 僅能使用 **raw** 設備, 比方 **/dev/rdisk** 或這些位在 **/dev** 下面且開頭有字母 **r** 的設備。例如 **/dev/rdiskette** 或 **/dev/rct0**。Linux 並沒有將這兩種設備獨立開來, 但提供自動選擇的功能以確保能使用正確的模式。

### 22.7.3 處理 DOS 磁片

目前在桌上型電腦上, 常看到微軟視窗系統和 UNIX 系統並存。UNIX 現在提供了一系列的指令 (表 22.1) 可以讀寫 DOS 磁片。這些指令開頭都有 **dos** 字串 (SVR4)。它們大多以 UNIX 指令的模式執行類似的功能。

最常使用的指令為 **doscp**, 可以在磁碟和磁片之間複製檔案:

```
doscp emp.lst /dev/dsk/f0q18dt:/per.lst
```

目標參數有兩個部分, 包括設備名稱 (1.44 MB 軟碟機) 以及檔案名稱 (**/per.lst**), 中間用冒號: 隔開。如同 **cp**, 也可做多重檔案複製:

```
doscp emp[123].lst /dev/dsk/f0q18dt
```

**doscat** 執行簡單的 “cat” 功能。當指定多個檔案時, 在標準輸出上會將全部串接在一起:

```
doscat /dev/dsk/f0q18dt:/CHAP01 /dev/dsk/f0q18dt:/CHAP02 > newchap
```

類似 UNIX, DOS 也有階層式的檔案系統, 也有自己的根目錄。檔案也類似 UNIX 檔案; 不過, 行結尾字元在 UNIX 系統中是 LF (line feed 換行——八進位值 012) 字元, 之前也稱為換行字元。而在 DOS 下, 行結尾字元是 CR (carriage return 回車——八進位值 015) 和 LF (9.16) 的組合。當使用這些指令複製和顯示檔案時, 就會自動做這樣的轉換。**doscp** 和 **doscat** 也支援 **-r** 選項, 在這樣的情況下就不會自動

轉換換行字元。

在某些機器上，這些“DOS”指令也可以處理DOS硬碟分割，前提是這個機器上有獨立的DOS分割區。這個設備名稱隨系統而有所不同；有些系統叫**c:**和**d:**。要看這些DOS分割區的檔案，這些指令都可以在SCO UNIX系統下使用：

```
dosdir c:
dosdir /dev/hd0d
dosdir /dev/dsk/0sC
```

**Xenix系統的DOS分割區設備名稱**

表22.1列出這些指令的使用範例。其中一個，應該可以適用在您的系統。如果**a:**和**b:**無效，那麼請使用**/dev**或**/dev/dsk**下適當的檔案名稱。



Linux  
注意事項

Linux下的“DOS”指令開頭字元均為**m**，指令名稱的剩餘部分和DOS上的指令相同。以下是一些例子：

```
mcop y emp.lst a:
mcop y emp[1-3].lst a:
```

```
mdir a:
mdel a:*.txt
```

注意到Linux使用的是DOS磁碟機名稱。這些指令均屬於“**mtools**”系列。要進一步瞭解的話，請用**man mtools**。

表22.1 DOS指令系列（小括號內為Linux指令名）

指令	動 作
<b>dosc p /dev/fd0135ds18:/tags .</b>	從DOS磁片複製檔案 <b>tags</b> ( <b>mcop y</b> )
<b>dosc at a:readme a:setup.txt</b>	將DOS磁片上的檔案 <b>readme</b> 和 <b>setup.txt</b> 串接在一起 ( <b>mtype</b> )
<b>dosdir /dev/dsk/f0q18dt</b>	在DOS磁片上列出檔案以DOS形式列表 ( <b>mdir</b> )
<b>dosls /dev/dsk/f0q18dt</b>	以UNIX形式列表
<b>dosmkdir a:bin</b>	在DOS磁片上建立目錄 <b>bin</b> ( <b>mmd</b> )
<b>dosrmdir a:bin</b>	移除DOS磁片上的目錄 <b>bin</b> ( <b>mrd</b> )
<b>dosrm /dev/dsk/f0q18dt:setup.inf</b>	刪除DOS磁片上的檔案 ( <b>mdel</b> )
<b>dosformat b:</b>	格式化DOS磁碟片 ( <b>mformat</b> )

## 22.8 cpio：複製輸入 輸出

通常要在經歷一次系統損毀、資料流失之後，系統管理員才會體認到備份資料的重要性。身為系統管理員，您的部分責任就是要維護系統內資料的安全性。您必須決定哪些資料需要備份，並且決定備份的時間週期。資料備份的效率，取決於您

救回資料的容易程度。

今日，最常用的兩個備份工具指令為cpio和tar。兩者都是將一群檔案，合併成一個備份資料檔(archive)，並為每個檔案內容加上適當的表頭。兩個指令各有各的優點，但兩種檔案格式之間並不是完全相容。tar的備份資料檔有時可以被cpio讀取，但反過來就不一定成立。

備份的設備可以是磁帶、卡帶或是磁片，甚至是磁碟檔案。小型的系統可能沒有磁帶機，所以這兒我們使用軟碟機來解說這兩個指令。



Note

在您的Solaris機器上執行cpio和tar指令之前，確認先將媒體管理系統常駐程式(volume management daemon)關閉，並利用指令 `/etc/init.d/volmgt stop`。這個常駐程式(vold)通常用來自動掛上光碟機，之後您可以下start參數重新啟動。注意在您停止這個常駐程式時，軟碟機內一定不可以有磁片。

指令cpio可以複製資料到備份設備或由設備來取出資料。它利用標準輸入，來讀取檔名清單。然後將這些檔案內容以及表頭複製進一個資料串流，此串流可以被轉向至一個檔案或設備。這表示cpio可以被用在轉向與管線處理。

cpio使用兩個「必要」選項，即**-o** (輸出) 和 **-i** (輸入)，指令行中一定要有這兩個選項其中一個 (不可同時存在)。另外還有伴隨這兩個選項的其他選項。這一節以及下一節的範例使用的是System V設備名稱。Linux使用者則應該使用 `/dev/fd0H1440` 這樣的名稱，而Solaris使用者則應使用 `/dev/rdiskette` 設備名稱。

### 22.8.1 備份檔案 (-o)

因為cpio只使用標準輸入，所以您可以使用ls來產生cpio所需的輸入。**-o** 選項會在標準輸出建立備份資料檔，所以您必須將它轉向到一個設備檔案。下例將目前目錄檔案，複製到1.44 MB磁片：

```
# ls | cpio -ov > /dev/rdisk/f0q18dt
array.pl
calendar
cent2fah.pl
convert.sh
xinitrc.sam
276 blocks
```

*在Solaris下使用 `/dev/rdiskette`*

*資料檔所佔空間*

選項 **-v** 使得cpio操作在回應模式，所以每個被複製的檔名會顯示在螢幕上。cpio所需要的輸入是檔案清單，一行一個檔案。如果這個清單已經建成一個檔案，就可以直接轉向：

```
cpio -o >/dev/rdisk/f0q18dt < flist
```

遞增式備份 指令find也可以產生一檔案清單，所以符合搜尋條件的檔案就會被備份。您會常常需要結合使用find和cpio來備份所選擇的檔案。例如，這些兩天內有被修改過的檔案：

```
find . -type f -mtime -2 -print | cpio -ovB >/dev/rdisk/f0q18dt
```

因為find所用的路徑清單為點，所以檔案是以相對路徑名稱來備份。然而，如果是/，那麼使用的就是絕對路徑名稱。

選項 **-B**將區塊大小設為**5120**位元組，此為預設數值的十倍大。若要更大或更小的區塊大小值，可以用選項 **-C**：

```
ls *.pl | cpio -ovC51200 >/dev/rdisk/f0q18dt
```

*預設的100倍大*

多磁片備份 當資料檔大小超過一磁碟片的容量時，cpio會提示插入新的磁片：

```
# find . -type f -print | cpio -ocB >/dev/rdisk/f0q18dt
Reached end of medium on output.
If you want to go on, type device/filename when ready
/dev/fd0
3672 blocks
```

*輸入設備名稱*

當cpio暫停並等候輸入時，輸入設備名稱。如果您的機器上有兩台軟碟機，您可以交替使用。這可以讓cpio一邊複製檔案一邊抽換磁片。依此方式，備份資料檔會被切成數個磁片檔（*extents* 或 *volumes*）。

## 22.8.2 還原檔案 (-i)

利用選項 **-i**可以將整個備份資料檔或選擇的部分檔案還原回來。要還原檔案時，利用轉向從裝置讀取資料：

```
# cpio -iv < /dev/rdisk/f0q18dt
array.pl
calendar
cent2fah.pl
convert.sh
xinitrc.sam
276 blocks
```

當還原子目錄時，cpio 假設硬碟上的子目錄結構仍然存在著；它無法自行建立這些目錄。然而，選項 **-d(directory)** 會覆蓋這個功能。

cpio也可以接受引號括住的萬用字元，所以可以還原所有符合的檔案。因此能很輕易地以下列指令，只還原 shell scripts：

```
cpio -i "*.sh" < /dev/rdisk/f0q18dt
```



Tip

檔案通常會被還原到原先備份時的目錄。如果使用絕對路徑名稱來備份（例如 `/home/romeo/unit13`），那麼這個檔案只會被還原到原先的目錄（`/home/romeo`）。然而，當使用相對路徑名來備份時，它們可以被還原至任意位置。所以建議系統管理員使用相對路徑名稱來備份資料。確認當您使用 `find` 時候，記得在路徑清單加個點而非 `/`。

處理修改的時間（`-m`）預設的情況下，當一個檔案被抽出時，檔案的修改時間會被設為抽取的時間。這並不是個好現象，因為這個檔案可能在下次要備份時被誤認為已經更動過了，而又備份一次。在這樣的情況下，您可以使用 `touch` (7.11) 來修改更動時間。另外，您也可以使用 `-m` 選項來讓 `cpio` 維持原先的檔案修改時間。

`cpio` 會去比較磁片上的檔案和磁碟上的檔案（如果存在的話）修改時間是否相同。如果磁碟上的檔案時間較新或者相同，那麼這個檔案就不會被還原，而 `cpio` 會顯示這個訊息：

```
" current <unit14> newer "
```

這是個有用的保護機制，可以保護最新版本的檔案。（`tar` 不支援這種功能）然而，這功能也可被 `-u` (unconditional) 選項覆蓋。



Tip

如果您常常將檔案從這個機器移到另一個機器上，建議您使用 `cpio` 而非 `tar`。這可以確保在某一機器上的最新檔案不會被舊的覆蓋。

### 22.8.3 顯示備份資料檔內容 (-it)

選項 `-t` 可以只顯示磁片內的檔案內容，而不會真的去還原。這個選項一定要配合 `-i` 使用：

```
# cpio -itv </dev/rdisk/f0q18dt
100755 henry      605  Oct 18 23:34:07 1997  cent2fah.pl
100755 henry      273  Oct 18 23:34:07 1997  check_number.pl
100755 henry      531  Oct 18 23:34:08 1997  dec2bin.pl
100755 henry      214  Oct 18 23:34:08 1997  get_home.pl
```

檔案的輸出列表會和上面相似（`Linux` 和 `Solaris` 系統的輸出會和上面的一樣）。這個檔案會顯示八進位值的檔案權限，以及檔案的修改時間（最接近的時間）。

### 22.8.4 其他選項

尚有其他三個重要的選項可以搭配 `-o` 和 `-i` 使用：

- 選項 `-r` (rename) 可以在每個檔案複製前，讓您更改檔名。系統會顯示每個檔案名稱，並提示您輸入。如果您輸入了一個檔名，這個檔案就會複製。若沒有輸入則檔案就不會被複製。
- 選項 `-f` 後面加一表示式，可以讓 `cpio` 去選擇表示式以外的檔案來處理：

```
cpio -ivf "*.c" </dev/rdisk/f0q18dt
```

還原C程式以外的所有檔案

- 選項 **-c**告訴cpio使用ASCII字元，而非二進位字元，來建立表頭。當在不同機器上使用時，您應該用這個選項來產生可攜度高的備份檔。

cpio的選項，列在表22.2。cpio要依靠另一個指令（通常是find）或檔案，才能提供檔案清單。它不能在指令行接受檔案名稱參數。cpio的備份資料檔在每次指令呼叫時，都會重新被覆蓋，無法附加新的檔案上去（Solaris和Linux除外）。這就是使用tar的時機了。

## 22.9 tar：「磁帶」備份資料程式

tar(tape archive)指令在cpio出現之前就存在了。今日，它不僅可以在磁帶上建立備份資料檔，也可以支援磁片。不同於cpio，tar正常狀態是不會寫到標準輸出（雖然也可以做到），但會在儲存媒介上建立備份資料檔。它是個多用途的指令，有些功能在cpio也找不到：

表22.2 cpio 附屬選項（搭配 **-i** 或 **-o** 使用）

選 項	意 義
<b>-d</b>	視需要建立目錄
<b>-c</b>	將表頭資料以ASCII字元建立，以維持較高的可攜度
<b>-r</b>	互動式更改檔名
<b>-t</b>	列出備份資料檔內容（僅能搭配 <b>-i</b> 選項）
<b>-u</b>	用舊檔案覆蓋新的檔案
<b>-v</b>	回應模式，即顯示被複製的檔名
<b>-m</b>	維持原先檔案的更動時間
<b>-f <i>exp</i></b>	複製所有除了表示式 <i>exp</i> 以外的檔案
<b>-C<i>size</i></b>	設定輸入—輸出的區塊大小為 <i>size</i> 位元組
<b>-A -O <i>device</i></b>	附加檔案至 <i>device</i> （Solaris和Linux適用）
<b>-H <i>tar</i></b>	建立或讀取 <i>tar</i> 表頭格式（Solaris和Linux適用）
<b>-E <i>file</i></b>	僅抽取列在 <i>file</i> 裡面的檔案清單（Solaris和Linux適用）

- 它不需要使用標準輸入來取得檔案清單。tar可以接受檔案和目錄參數。
- 它可以複製一個或多個完整的目錄樹；亦即，預設是遞迴式的處理。
- 它可以在單一個資料備份檔中建立同一個檔案的數個不同版本（Solaris和Linux下的cpio也可以）。
- 它可以附加新的檔案上去，不會覆蓋原有的整個資料檔（Solaris和Linux下的cpio也可以）。

`tar`有許多必要的選項，最常用的是 **-c**（複製），**-x**（解開）以及 **-t**（列表）。**-f** 選項用來指定裝置名稱，`tar`選項列在表22.3。

### 22.9.1 備份檔案 (-c)

`tar` 可以在指令行上接受檔案和目錄參數。選項 **-c**指令要將資料備份至設備上：

```
# tar -cvf /dev/rdsk/f0q18dt /home/sales/SQL/*.sql
a /home/sales/SQL/invoice_do_all.sql 1 tape blocks
a /home/sales/SQL/load2invoice_do_all.sql 1 tape blocks
a /home/sales/SQL/remove_duplicate.sql 1 tape blocks
a /home/sales/SQL/t_mr_alloc.sql 10 tape blocks
```

這個動作，將所有SQL scripts以絕對路徑表示法備份至磁片上。字元**a**表示每個檔案是附加上去的。選項 **-v**會顯示每個檔案所佔用的區塊數。

表22.3 `tar`選項

主要（僅能選一個用）	
選項	意義
<b>-c</b>	建立新的備份資料檔
<b>-x</b>	由解開備份資料檔，萃取檔案
<b>-t</b>	列出備份資料檔內容
<b>-r</b>	附加檔案至備份資料檔尾端
<b>-u</b>	如 <b>r</b> ，但前提是這些檔案比備份資料檔上面的還新
附屬選項	
選項	意義
<b>-f <i>device</i></b>	使用路徑名 <i>device</i> 做為裝置名稱而非使用預設值
<b>-v</b>	回應模式，可列出完整格式的檔案清單
<b>-w</b>	執行動作前須由使用者確認
<b>-b <i>n</i></b>	使用區塊因子 <i>n</i> ， <i>n</i> 上限為20
<b>-m</b>	將檔案修改時間設為解開時間
<b>-l <i>file</i></b>	由檔案 <i>file</i> 讀取檔案清單（Solaris適用）
<b>-T <i>file</i></b>	由檔案 <i>file</i> 讀取檔案清單（Linux適用）
<b>-k <i>num</i></b>	多磁片備份，可設定每一份的大小為 <i>num</i> 千位元組（Solaris適用）
<b>-M</b>	多磁片備份（Linux適用）
<b>-z</b>	使用gzip壓縮/解壓縮（Linux適用）
<b>-Z</b>	使用compress壓縮/解壓縮（Linux適用）
<b>-X <i>file</i></b>	排除 <i>file</i> 中的檔案（Solaris和Linux適用）



Note

tar對於參數的的寫法很自由，所以tar cvf同等於tar -cvf。符號-甚至不需要。然而，新版的tar將不支援這點。

當使用絕對路徑名稱複製檔案時，也受到同樣的限制；它們僅能被還原到原先同樣的目錄。若不想受這種限制，您必須先“cd”到/home/sales/SQL然後使用相對路徑名稱：

```
cd /home/sales/SQL
tar -cvf /dev/rdsk/f0q18dt ./*.sql
```

使用./

這個指令可以執行的快一點，如果區塊大小設為18（亦即，18×512位元組）：

```
tar -cvfb /dev/rdsk/f0q18dt 18 *.sql
```

./ 並非一定需要

因為-f和-b後面都必須加個參數，所以頭一個參數(/dev/rdsk/f0q18dt)是-f的參數，而第二個參數(18)則是-b的參數。



Tip

不要使用tar和cpio預設的區塊大小。選擇大一點且系統允許的值。這會讓I/O速度加快。

tar的優點在於它能複製整個目錄樹以及其下的子目錄。目前目錄下的隱藏檔可以備份，也可以不備份：

```
tar -cvfb /dev/rdsk/f0q18dt 18 *
tar -cvfb /dev/fd0 18 .
```

不備份隱藏檔  
連隱藏檔一起備份

檔案假設以一片磁片，採相對路徑名來備份，當容納不下時，tar在System V下會盡可能的塞，但若失敗的話會直接跳出程式而不會有警告訊息。



Caution

當遞迴式的複製遇到目錄層次很深時會有問題。tar被發展出來的時候UNIX上的檔案名稱被限制為十四字元。即使是現在，tar仍不能處理路徑名稱超過一百個字元以上。在現代的UNIX系統上可能會有很深的目錄層次，這個問題會是個嚴重的限制。然而，Solaris系統允許更高的上限，並且還可以利用選項-E進一步放寬名稱長度限制。

多磁片備份(-k) 對於多磁片備份，tar在Solaris（以及SCO UNIX）下有一個特殊選項(-k)，後面加上每一分片(volume)大小，單位為仟位元組。下例在SCO UNIX下將檔案index分片備份：

```
# tar -cvfkb /dev/rdsk/f0q18dt 1440 18 index
Volume ends at 1439K, blocking factor = 18
tar: large file index needs 2 extents.
tar: current device seek position = 0K
+++ a index 1439K [extent #1 of 2]
```

tar預估需要兩張1440 KB（選項-k）容量的磁片。當第一片裝滿時，tar提示插入



新磁片：

```
tar: please insert new volume, then press RETURN.
```

當檔案要還原時，也要加上同樣的選項。

### 22.9.2 還原檔案 (-x)

檔案的還原要使用選項 **-x**。當沒有指定檔案或目錄名稱時，它會將所有檔案從裝置中還原回來。以下是從剛剛做的備份中還原檔案的例子：

```
# tar -xvfb /dev/rdisk/f0q18dt 18
x /home/sales/SQL/invoice_do_all.sql, 169 bytes, 1 tape blocks
x /home/sales/SQL/load2invoice_do_all.sql, 456 bytes, 1 tape blocks
x /home/sales/SQL/remove_duplicate.sql, 237 bytes, 1 tape blocks
x /home/sales/SQL/t_mr_alloc.sql, 4855 bytes, 10 tape blocks
```

要選擇某些檔案加以還原的話也沒問題，只要指定一個或數個目錄或檔案：

```
tar -xvf /dev/rdisk/f0q18dt tulec1 project2
```

不同於cpio，當檔案被解開、萃取出來時，檔案的修改時間維持不變。這可以利用選項 **-m** 來改成為檔案解開時的系統時間。



Note

不同於cpio，某些版本的tar（比方Solaris）不支援萬用字元。如果您使用的指令為tar -xvf /dev/fd0 \*.pl，這是shell試著將萬用字元展開，意思是這些檔案必須存在於目前目錄。大部分情況下這不會成立，所以檔案萃取會不完整。您必須在指令行明確指定檔案名稱。

某些版本的tar（比方Linux）的確允許萬用字元；它不在乎是否檔案存在與否。tar會使用萬用字元來比對儲存媒介上的檔案，所以這打破了「未被引號括住的萬用字元一律由shell處理」這個理論。當您使用tar時應該先檢查這一點。



Caution

當多磁片備份檔要還原時，您必須從正確的第一片開始做起。tar無法幫您判斷，也不會警告您是否是第一片，它只會盡可能的解開資料檔。所以檔案很容易遭到截斷。

### 22.9.3 顯示備份資料檔內容 (-t)

如同cpio，選項 **-t** 僅顯示資料檔清單，而不會真正去還原。當搭配選項 **-v** 時，會以類似ls -l的完整格式來列出：

```
# tar -tvf /dev/rdisk/f0q18dt
rwxr-xr-x203/50    472 Jun  4 09:35 1991 ./dentry1.sh
rwxr-xr-x203/50    554 Jun  4 09:52 1991 ./dentry2.sh
rwxr-xr-x203/50   2299 Jun  4 13:59 1991 ./func.sh
```

這裡有一些是您必須注意的。上面的檔案是以相對路徑備份的。每個檔案名前面都有 `./`。如果您沒有注意到這點，而想直接萃取出檔案 `func.sh` 的話，您很可能會這樣嘗試：

```
# tar -xvf /dev/fd0 func.sh
tar: func.sh: Not found in archive
```

使用標準軟碟設備檔

`tar` 找不到這個檔案，因為是 `./func.sh` 而非 `func.sh`。加上 `./` 在檔名之前，才能成功。當您遇到類似的錯誤訊息時，要記住這種處理方式。

#### 22.9.4 使用tar搭配壓縮工具

`tar` 不一定只能使用設備檔名；選項 `-f` 後面也可以接一般檔案名稱。這表示您可以將一整群檔案，收集在硬碟上單獨的備份資料檔。而且因為 `tar` 可以遞迴式操作，您可以將整個目錄樹，收集成一個資料檔：

```
tar -cvf quotes.tar quotes.dir
tar -cvf quotes.tar *
```

quotes.dir是個目錄

頭一個指令使用目錄名作為參數，第二個則使用目前目錄下的所有檔案。在任一個情況下，檔案 `quotes.tar` 會被建立，內含多個檔案（以及目錄）。您現在就可以將此檔郵寄給另一位使用者，讓他與你共同處理這些檔案，而他也不用再“telnet”或“ftp”到你的帳號。一般來說，在郵寄的這個檔案還會經由 `compress` 或 `gzip` 來壓縮。以下指令的輸出是同樣檔名後面加上 `.Z` 或 `.gz` 的副檔名：

```
compress quotes.tar
gzip quotes.tar
```

建立 quotes.tar.Z

建立 quotes.tar.gz

`cpio`, `tar` 和壓縮工具 `compress`, `gzip` 實際上可以一起動作。您可以利用管線處理：

```
tar -cvf - quotes.dir | compress > quotes.tar.Z
tar -cvf - quotes.dir | gzip > quotes.tar.gz
```

注意到這次我們特意輸入 `.tar`, `.Z` 和 `.gz` 的副檔名。`compress` 和 `gzip` 都是根據這些副檔名來判斷壓縮的格式。這個檔案現在就可以被當作一些發信軟體像 `elm`、`pine` 和 `Netscape` 的附件來寄出。

在接收端，此附件必須先儲存在某個檔案，最好是相同的副檔名。解壓縮的過程類似但動作相反。現在利用 `uncompress` 來解壓縮：

```
uncompress quotes.tar.Z
tar -xvf quotes.tar
```

建立 quotes.tar  
還原目錄 quotes.dir

一個“tar-gzipped”的檔案也是採類似的解壓縮方式：

```
gunzip quotes.tar.gz
tar -xvf quotes.tar
```

建立quotes.tar  
還原目錄quotes.dir

當建立備份資料檔時，我們利用管線來避免中間檔的產生。類似的原理，我們也可以應用在解壓縮上：

```
uncompress -c quotes.tar.Z | tar -xvf -
gunzip -c quotes.tar.gz | tar -xvf -
```

包含文字的大型檔案可以壓縮的很小，至於GIF和JPEG檔案則很難壓縮，因為它們本身儲存的就是壓縮後的格式 (6.19)。注意到tar 使用 - 來同時代表標準輸入和標準輸出，並利用主要選項（-c或-x）來指定處理方式。

### 22.9.5 其他選項

tar尚有一些值得注意的選項：

- -r主要選項，用來將檔案附加至備份資料檔中。因此一個備份資料檔，可能會包含同一個檔案的不同版本。
- -u主要選項，也用來附加檔案至備份資料檔中，不過只有當附加檔案原先不存在或會將舊的版本覆蓋掉。選項 -c不能和 -r或 -u搭配使用。
- -w選項，提供互動式的複製與還原。它會顯示檔案名稱，並提示您輸入按鍵（y或n）。
- 某些版本的tar使用特殊的選項，可從一檔案中讀取檔案名稱清單。您可能曾使用這個功能，特別是您已經有上百個檔案的表列時，要一個一個檔案輸入指令行是不切實際的做法（有時也不可能）。不過不幸地，這個選項不是標準的格式。Solaris使用 -l，而Linux中是使用 -T。



Tip

使用 -r主要選項，來附加檔案到備份資料檔中。使用 -u選項，僅能儲存較新版的檔案。tar也可以在單一個備份資料檔中，儲存同一個檔案的多個版本。



Note

當使用cpio或tar時，確認使用的是raw設備；亦即，設備檔位在 /dev/rdisk（而非 /dev/dsk）之下，或在 /dev之下且開頭有個r字。如果使用的是block設備，在製作多磁片備份時可能會出問題。Linux使用者，則不需擔心此問題。



Linux  
注意事項

GNU tar具備的功能遠比System V的版本強大，也支援更多獨特的選項。不幸地，這些選項有些和System V中同樣的選項意義不同。所以會遇到一些可攜性的問題。

壓縮（-z和 -Z） GNU tar支援備份時一併進行資料壓縮。選項-z會呼叫gzip，而-Z則會呼叫compress：

```
tar -cvzf /dev/rft0 .
tar -cvZf /dev/rft0 .
```

使用gzip壓縮  
使用compress壓縮

當要還原檔案，或者列出檔案內容 **-t** 時，選項 **-z** 或 **-Z** 也必須一起搭配使用。

多磁片備份 (**-M**) 如果檔案無法塞進一片，tar 會顯示警告訊息。不過，若使用選項 **-M** 的話會聰明的多：

```
# tar -cvf /dev/fd0H1440 -M *
.....
Prepare volume #2 for /dev/fd0H1440 and hit return:
```

tar 會從設備名稱自行判斷磁片大小。System V 版本的 tar 則沒有辦法做到這點。

## 22.10 管理磁碟空間

管理磁碟空間，是系統管理員一項重要的工作。檔案的數目有很多，特別是在 **/tmp** 和 **/var/tmp** 下面，更是隨時間不斷增加成長。如果不去檢查，磁碟空間很快就被耗盡。如此系統的效能會大受影響。平時，系統管理員可以使用 **df** (6.17)，**du** (6.18) 以及 **find** (7.15) 指令來監控磁碟空間。我們會再用到這些指令，而這次是以系統管理員的觀點，來進行概念論述。

### 22.10.1 評估使用者所消耗的空間 (du -s)

大部分的系統動態空間，都是被使用者的家目錄和資料檔案所耗去。您可以使用 **du -s** 加上 **/home/\*** 參數來報告每個使用者家目錄的佔用空間。輸出雖然簡潔，卻包含了不少資訊：

```
# du -s /home/*
1204      /home/ftp
144208    /home/henry
1536      /home/httpd
98290     /home/image
28346     /home/sales
```

匿名ftp的家目錄

網頁伺服器的家目錄

**du** 也可以報告每個目錄下的檔案 (**-a** 選項) 所佔用的空間，不過這樣一來清單就太長了。您可能需要的是，找出真正佔用過多空間的兇手，所以這時候應該利用 **find** 指令。

### 22.10.2 find：系統管理工具

**find** 指令 (7.15) 可以利用任何檔案屬性來尋找檔案。在這裡我們考慮的是有關

磁碟管理方面的選項。

尋找大檔案 (-size) find利用選項 **-size**來尋找大檔案。也可以指定多個搜尋條件：

```
find /home -size +2048 -print
```

檔案大於1 MB

```
find /home -size +2048 -size -8192 -print
```

大於1 MB小於4 MB

find在這裡使用的單位是一區塊 (512位元組)。它也可以讓您精確地指定檔案大小。如果您不記得大小為38,765位元組的檔案被放在某個目錄，您可以下find 指把它找出來：

```
find / -size 38765c -print
```

尋找沒被使用的檔案 (-mtime以及 -atime) 許多檔案經年累月都沒有被存取過。find的選項 **-mtime**和 **-atime**可以根據檔案修改時間和檔案存取時間來找出這些檔案。以下的指令系統管理員用來尋找 /home 目錄下一年內沒被存取或半年內沒有被修改的檔案：

```
find /home -atime +365 -o -mtime +180 -print | mail root
```

遞增式備份 (-newer) 您也可以利用find來執行遞增式備份。首先，您要先產生一個零位元組，代表目前系統時間的檔案，後續的備份則會選擇時間比這個檔案還新的來備份。以下是這樣的範例：

```
find /home -newer .last_time -print | cpio -o > /dev/rct0
touch .last_time
```

這兩行隨時可以一起用，而touch用來確保最後一次備份的時間會被儲存在檔案的修改時間標籤。您也可以使用 tar：

```
tar -cvf /dev/rct0 `find /home -type f -newer .last_time -print`
touch .last_time
```

它的型態指定為**f**，因為如果不這樣指定的話，find會將目錄也顯示出來，這樣對tar來說就是備份整個目錄了。這樣檔案就可能被備份兩次了。

避免備份掛上的檔案系統 (-mount) 要備份整個根目錄檔案系統，一般來說先要將其他的檔案系統卸下。不過現在您不須這樣做了，find已經提供選項 **-mount**來避免這種情形發生。所以要備份根目錄系統下，在一天之內有修改過的檔案就相當簡單：

```
find / -depth -mount -mtime -1 -print | cpio -ocvB -0 /dev/rct0
```

此時find不會去處理到其他的檔案系統。選項 **-depth** 用來確保目錄裡的檔案會先被處理，然後才是目錄本身。如果選項 **-mtime** 忽略的話，這個動作就會備份整個根目錄檔案系統。find的選項列在表7.5。



Tip

您必須在系統一安裝完成後，立刻將整個根目錄檔案系統備份至磁帶。之後，您必須進一步的備份（一般是遞增式備份）修改過的檔案系統。

### 22.10.3 xargs：建立動態的指令行

當使用指令find的選項 **-exec** (7.15.1) 時會有個問題。如果find產生的清單有兩百個檔案要移除時，rm會被執行兩百次。xargs可以解決這個問題，因為它可以讓rm（或任何其他的UNIX指令）僅執行一次，而將兩百個檔案名稱當作參數。

xargs 是UNIX系統中的一個黑馬，但卻是個不被廣泛瞭解與使用的指令。它可以將標準輸入的資料轉成一個清單，然後將此清單當作指令執行的參數。下面的兩個指令功能一樣，但第二個速度快多了：

```
find /usr/preserve -mtime +30 -exec rm -f {} \;
find /usr/preserve -mtime +30 -print | xargs rm -f
```

此處xargs會由指令find取得清單，並且將它當作單一個參數集合，提供給rm。所以即便find找到了30的檔案，rm只會被執行一次。

指令通常都有參數數目的上限。xargs可以用 **-n** 選項來提供特定數目的參數給指令：

```
find / -name core -size +1024 -print | xargs -n20 rm -f
```

如果find找到一百個檔案，rm會被執行五次 每次二十個檔案參數。這不是個很方便的工具嗎？

#### ► 進階篇

## 22.11 利用passwd做密碼的管理

指令passwd有許多的選項，可以用來協助管理密碼。當一個使用者忘記自己的密碼時，或者要指定一個密碼給一新使用者時，系統管理員必須讓她們可以順利登入，然後再由她們自行設定新密碼。這只要將使用者的密碼刪除即可，而使用者就可以直接登入系統：

```
passwd -d henry
```

*此時登入就不需要密碼了*

這樣的動作可能太過冒險，而有額外的問題；有時候使用者可能根本無法設定新密碼。所以系統管理員可能希望自己先幫她設一個，然後強迫她第一次登入就要改密碼。這可以利用選項 **-f**（Linux不適用）：

```
passwd -f henry
```

下次登入時必須更改密碼

UNIX系統，原先是將密碼編碼後儲存在檔案 `/etc/passwd`中。不過因為這個檔案是眾人可讀的，常常成為駭客攻擊的對象，所以較新的系統，改將密碼儲存在 `/etc/shadow`中。`shadow`內含九個欄位，包括編碼後的密碼，並且是一般人不能讀取的。大部分的欄位可以利用`passwd`加上適當參數來更新。

對於密碼變更的管理，`shadow`也提供了一個不錯的方式，稱為密碼使用限期(`password aging`)。您可以強迫使用者定期更改密碼。您也可以限制她不要過於頻繁地更改密碼：

```
passwd -n 14 julie
passwd -x 30 julie
```

最少14天才能修改  
最多30天內一定要修改

`-n 14`限制了`julie`密碼變更期間，要大於兩個星期，而`-x 30`則是限制她30天內必須更換密碼。

有時候為了安全性考量，可能需要將一個使用者帳號鎖住。這時候您可以利用`passwd` 配合選項 `-l(lock)`來設定。

## 22.12 init所使用的rc Scripts

`init`以及 `/etc/inittab`完全地控制了系統的開機和關機的方式。進一步來說，當系統更換了執行層級(run level)，`init`會讀取`inittab`檔並決定在這個層級下什麼程序要執行，什麼程序要終止。它會先終止不該執行的，然後再產生出要執行的程序。

每個`inittab`檔案中，都指定了可能位在 `/etc`或 `/sbin`目錄下，要被執行的某些rc scripts。這些 scripts 的名稱為 `rc0`, `rc1`, `rc2`，而後面的數字代表執行層級。這也可以從以下`/etc/inittab`摘錄的幾行中看出：

```
s0:0:wait:/sbin/rc0 >/dev/console 2<>/dev/console </dev/console
s1:1:wait:/sbin/shutdown -y -iS -g0 >/dev/console 2<>/dev/console </dev/console
s2:23:wait:/sbin/rc2 >/dev/console 2<>/dev/console </dev/console
s3:3:wait:/sbin/rc3 >/dev/console 2<>/dev/console </dev/console
s6:6:wait:/sbin/rc6 >/dev/console 2<>/dev/console </dev/console
```

`init`會執行相對於執行層級的 scripts。然而，`rc2`可以在層級2和3執行。每個rc scripts 又進一步指定執行一系列位於 `/etc/rcn.d`目錄下的 scripts。以執行層級2為例，`init`會執行 `/etc/rc2`，然後又會去執行目錄 `/etc/rc2.d`下的一系列 scripts。

### 22.12.1 啟動與移除的 Scripts

現在來看一看位於目錄 `/etc/rcn.d`下的 scripts。這些目錄存放了兩種類型的檔



Linux  
注意事項

## rc檔案

Linux下的起始設定檔案是源自於BSD系統，不過現在有個功能更強大強的 System V 的版本。不過，rc檔始終是位於一個目錄下，即 **/etc/rc.d**。進一步來說，Linux並不使用 *rcn*，而是使用單一個rc後面加上不同的參數，如下所示：

```
I0:0:wait:/etc/rc.d/rc 0
I1:1:wait:/etc/rc.d/rc 1
I2:2:wait:/etc/rc.d/rc 2
```

所有位在目錄 **rcn.d** 下的 scripts 都是由 **/etc/rc.d/rc** 來執行。所以對Linux來說，執行的順序為：要切換到執行層級 *n*，init會執行 **/etc/rc.d/rc n**，這又會執行位於目錄 **/etc/rc.d/rcn.d** 下的 scripts。

案，以目錄 **/etc/rc3.d** 為例：

K60nfs.server	S69inet	S75cron	S91agaconfig
K76snmpdx	S70uucp	S76nsd	S91leoconfig
K77dmi	S71rpc	S80lp	S92rtvc-config
S20syssetup	S73cachefs.daemon	S85power	S99audit
S21perf	S73nfs.client	S88sendmail	S99dtlogin
S30sysid.net	S74autofs	S88utmpd	

這些位於目錄 **rcn.d** 裡的 scripts 會完整地啟動系統，藉由掛上檔案系統、設定網路以及啟動常駐程式。它們會分兩梯次執行。當系統進入層級3時，rc會執行（以ASCII順序）所有K(kill scripts)開頭的 scripts 加上 **stop** 的參數來終止某些不該在這個層級執行的程序。然後會在執行S(start scripts)開頭的 scripts 加上 **start** 的參數來啟動某些程序。如果您看一下rc3 script，您會發現有兩個for迴圈，做的就是上述的步驟。

因為一個 script 可能會在超過一個以上的層級被執行，所以這種 scripts 也會有一些鏈結指向位在目錄 **/etc/init.d** 下的檔案。在這個目錄下的這些檔案，有著類似的名稱，不過檔名的前三個字元被去掉。藉由這樣明確及合理的安排，您只需更改一個地方的檔案。

身為一個系統管理員，您必須瞭解這些 scripts 如何動作。您也必須能夠指出某個 script 所啟動的服務。通常檔案名稱會是個線索，不過有時候您也需要使用grep來找出那個 script。比方我們要確定哪個 script 會啟動網際網路常駐程式：

```
# grep -l inet *
S69inet
S72inetsvc
```



有可能第一個就是要找的 script。不過為了進一步確定，還是要看看 script 的內容。



Note

在目錄 `rcn.d` 下的 scripts 若名稱為 S 開頭的話代表是用以啟動某個服務，而以 K 開頭的話代表要終止某項服務。刪除的動作會先執行，然後才會去啟動服務。它們都由對應的目錄 `/etc/rcn`（或者 Linux 下是 `/etc/rc` 加上參數  $n$ ）下的 scripts 來執行， $n$  代表執行層級。

## 22.13 終端機管理

在工作站和 PC 出現之前，終端機一度支配了 UNIX 的世界。時至今天仍有它們應用的場所，不過前提是不需要圖形輸出時（本書僅有一章，即第十二章會討論圖形介面）。終端機可以經由串列埠的其中一個或經由串列介面卡提供多個連接埠，來連接到 UNIX 主機。PC 通常有兩個串列埠（COM1 和 COM2），可以用來連接終端機、數據機以及滑鼠。

UNIX 的優越之處，在於這些設備可以很簡單的就設定完成。幾乎不需要為終端機寫個驅動程式。即使是在 X 視窗下，`xterm` 所使用的驅動程式和組態設定檔都和簡單的終端機一樣。您會發現要更改一個終端機的特性是很容易的事。

### 22.13.1 getty 以及 /etc/gettydefs

除了使用起始 scripts 來啟動系統常駐程式之外，`init` 也直接控管終端機。它會在每個連接到終端機的串列埠上，啟動一個 `getty` 服務。在 System V 中，每個終端機需要一個在 `/etc/inittab` 裡面的設定值，以及在 `/etc/gettydefs` 裡面有個相對應的值。以下是兩個典型的設定值：

```
c02:234:respawn:/sbin/getty tty02 m
c03:234:off:/sbin/getty tty03 m
```

終端機已連接上  
終端機未連接

第一行的意思是：「當執行層級為 2, 3, 4 時，執行 `getty` 程式並附加兩個參數 `tty02` 和 `m`。」`respawn` 的設定可以確保終端機總是被啟動的；亦即，`getty` 在每次有使用者登出後程序就會被終止，然後再次啟動。您不會在終端機 `tty03` 上看到登入提示符號，因為動作設定值為 `off`。

`getty` 之前介紹過（10.4）是由 `init` 所複製執行（`fork-execd`）的程序，用來在螢幕上產生 `login`：提示字串。它也會去比對 `/etc/gettydefs` 的第一欄，檢視是否有符合第二個參數（`m`）的設定。`/etc/gettydefs` 這個檔案包含了終端機的定義：

```
l # B4800 HUPCL # B4800 CS8 SANE HUPCL TAB3 ECHOE IXANY #\r\nlogin: # m
m # B9600 HUPCL # B9600 CS8 SANE HUPCL TAB3 ECHOE IXANY #\r\nlogin: # n
n # B19200 HUPCL # B19200 CS8 SANE HUPCL TAB3 IXANY #\r\nlogin: # l
```

在這裡，`getty` 一開始會使用第二行，因為是 `m` 開頭。它試著將終端機速度設為鮑率

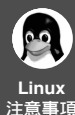
9600並產生提示符號。同一行的最後一欄為 **n**，意指如果在此連接線上接收到中斷 (break) 的訊號，下一次 **getty** 會去試 **n** 開頭的那行設定。所以這裡形成了一個循環搜尋：首先是 **9600**，然後 **19200**，之後為 **4800**，然後回到 **9600**。如果同一行的開頭和結尾標籤都一樣，就無法形成循環。

一旦終端機被安裝好，且這兩個檔案的設定值正確，就可以要求 **init** 重新讀取 **/etc/inittab**。同樣的，在 **/etc/gettydefs** 裡面的更動必須要利用 **getty** 的選項 **-c** 來加以檢查：

```
telinit q
getty -c /etc/gettydefs
```

**重新讀取inittab  
檢查gettydefs是否正確**

Solaris並沒有使用 **gettydefs**。**getty** 在這裡只是一個符號鏈結，指向 **ttymon** 這個埠監視程式，且會在 **inittab** 裡面加上選項被呼叫。



Linux有許多“**getty**”常駐程式，包括**agetty**、**mingetty**以及**uugetty**。它也使用了一個獨立的常駐程式用來處理數據機連接到串列埠，即**mgetty**。以上這些都不使用**gettydefs**。它們或者有自己的組態設定檔，或者以類似“**getty-defs**”的方式被呼叫。

### 22.13.2 設定終端機型態

除了終端機定義檔之外，許多系統使用另一個檔案來儲存終端機型態。這通常是 **/etc/ttytype** 檔。每個終端機設備名稱對應到一行，其中包含兩個欄位，也就是終端機類型以及終端機名稱：

```
vt220    tty01
ansi     tty1a
dialup   tty2A
```

終端機類型實際上是 **/usr/lib/terminfo** 裡面的設定值，即專門用來控制終端機特性的資料庫。例如，**vt220** 的屬性可以在二進位檔 **/usr/lib/terminfo/v/vt220** 中被找到。預設的情況下，變數 **TERM** 是根據讀取檔案 **/etc/ttytype** 來被設定的。而其他的終端機特性可以用 **stty (3.5)** 來設定。



Note

如果一終端機是連接至一多連接埠介面卡，檔案 **inittab** 會有等同於連接埠數目的設定行。確認每個終端機的動作欄位有被啟動，否則終端機就不能使用。

### 22.13.3 使用虛擬終端機

如果您是在PC上使用UNIX或Linux系統，您可以在同一機器上經由多個「終端機」重複登入。這聽來有點不可思議，試試看 **[Alt][F2]** 或 **[Alt][F3]** 吧。您會在螢幕上看到 **login:** 的符號。UNIX提供了許多的虛擬終端機 (**virtual terminals**)，每一

個都可以利用按鍵 *[Alt]* 加上一功能鍵來存取。您可以登入這些虛擬終端機，並啟動個別的登入畫面，而使用相同或者不同的使用者名稱。

您可以在一個終端機上編輯程式，並在另一個終端機上編譯並執行它。這種技巧相當簡單。在一個終端機畫面執行您的編輯器（*vi*或*emacs*）並載入欲編輯的程式。儲存檔案但不要離開編輯器，然後切換到另一個終端機畫面上，進行編譯並執行它。指令行的指令只要輸入一次（假設您不是使用*Bourne shell*），之後您只需使用`!!`或`r`（重複上一個指令）即可。

當您像這樣啟動虛擬終端機時，您會發現您可以更有效率的作業，甚至比視窗環境下更快。這就是為什麼有些人只使用UNIX！

## 22.14 基本列印知識

印表機管理也是系統管理員一個重要的工作，而且在一個大型的系統下，通常也很耗時。UNIX提供了印表排程(*spooling*)控制的子系統，包含許多指令，可以確保列印工作依序進行。這包含了印表機佇列管理、新增或移除印表機、啟動或停止排程控制子系統，以及管理特定的印表機。這些指令在SVR4和Linux下都不相同。

當一使用者使用*lp*來列印檔案時，這個檔案被儲存在一個佇列中。SVR4下的排程控制機制，是由*lpsched*常駐程式來處理。當切換至多使用者模式時，這個常駐程式會被*rcn.d*目錄下的其中一個 *Scripts* 所呼叫啟動。而當列印的動作停止時，您也能用下面的這個方式來檢查是否*lpsched*仍在執行：

```
ps -e | grep lpsched
```

如果沒有在*rc2.d*或*rc3.d*目錄下，察看一下“S”開頭且包含字串*S\*lp* (*lpsched*) 的 *scripts*：

```
ls /etc/rc2.d/S*lp
```

末尾字“*lp*”指的是點陣印表機 (*line printers*)，現今已經被雷射和噴墨印表機所取代。不過，只有末尾字及基本的技術卻仍未改變。現今的印表機大部分是位元映射 (*bitmap*) 的設備，輸出是由數不盡的點所構成。一個印表機解析度為600 × 600 d.p.i. 代表能夠在每一英吋平方的面積上列印600個點。但UNIX系統並不是設計來支援處理這樣的印表機。

現在的印表機可以接受由頁描述語言 (*page description, language PDL*)格式化之後的輸入。今天，這些PDL選項僅剩下兩種，包括Adobe的*Postscript*以及Hewlett Packard的（惠普）*PCL*。*Postscript*是個程式語言，由許多小括號、大括號以及斜線來加以辨認。UNIX指令無法產生這種輸出，所以傳統的UNIX字元輸出不被*Postscript*印表機所接後。

為了要協調這兩種技術，輸出的字元資料要先經由介面程式來處理。這種介面

程式通常是一些指令 scripts，用來控制一些列印選項，比方列印標題頁、設定紙張的大小以及指定印表機控制碼。這種介面程式通常還會呼叫外部程式像 Ghostscript (gs) 或 netpr 來完成最後的列印。



Note

這種介面程式，可以想成是微軟視窗系統下的印表機驅動程式，當您安裝印表機時必須設定它們。如果您沒有正確地指定這些介面程式，印表機的字印動作就會不正確。



Linux  
注意事項

## lpd系統

列印工作排程，是經由 lpd 這個 Linux 下的常駐程式來完成，這是源自於 BSD 系統。要檢查這個常駐程式是否有在執行，可以用：

```
ps ax | grep lpd
```

如果沒有，察看 “S” 開頭且符合字串 **S\*lpd** 針（對 lpd）的 scripts，可能在目錄 **rc2.d** 或 **rc3.d** 下面：

```
ls /etc/rc.d/rc2.d/S*lpd
```

Linux 下有許多工具程式，可以產生 Postscript 輸出，比方 gs, a2ps 以及 enscript。利用 a2ps 由 ASCII 文字檔轉成 Postscript 輸出。反向轉換的話可以用 ps2ascii。

## 22.15 管理 SVR4 印表機

除了 lp 指令外，spooling 系統尚包含下列指令群：

- 使用者層級指令如 cancel（用以取消列印工作）及 lpstat（用以檢視列印佇列，queue）。而使用者只能刪除自己送出之列印請求。
- 管理層級指令如 lpadmin、lpsched、lpshut、accept 與 reject、enable 及 disable。

不幸的是這些指令並沒有一致的存放路徑位置。您可能會在 **/usr/lib** 中找到部分指令，而在 **/usr/sbin** 中找到另一部分的指令。整組列印應用指令中，最重要的是 lpadmin 程式。您能使用此指令新增、移除印表機，或者修改印表機設定。指令中通常會應用到的選項有

- 指定印表機名稱 (**-p**)。
- 定義印表機類別 (class) (**-c**)。
- 指派介面 script (**-e** 與 **-i**)。
- 指定印表機設備檔 (**-v**)。
- 訂定印表機型號 (model) (**-m**)。
- 設定預設印表機 (**-d**)。
- 刪除印表機 (**-x**)。

`lpadmin` 指令是將列印輸出的最後接收端，視為列印工作之最終目的地(destination)，而不是單純的直接指向印表機。因此，最終點有可能是一印表機或是同一類別(class)的印表機群。當您有一堆同型號的印表機時，劃分類別在處理印表工作上，會讓您相當的便利。譬如來說，當同時有三個列印工作要求同類別的印表機列印時，`lpsched` 程式會自動將那些工作分到同類別的各個印表機。假如所有的印表機都在忙線，列印工作會進行排序，以等待群組中最先空出來的印表機來服務。

`lp` 程式將所有的列印請求，暫存於 `/var/spool/lp/request` 目錄中。每一部被 `lpadmin` 程式登記的印表機，在此都建有目錄。不同於 BSD 系統，除非您執行 `lp` 指令時加上 `-c` 選項，否則目錄中不會複製請求列印的檔案。如果在列印請求排進佇列後你又修改檔案，`lp` 程式是以祈求方式 (the method of invocation) 來決定檔案是否會在最後列印時隨之更動。

而 `lp` 程式會產生 `/usr/spool/lp/SCHEDLOCK` 的鎖定檔，以避免第二支排序程式 `lpsched` 的實體被啟動造成衝突。因此如果列印工作被不正常中斷時，即如系統瞬間斷電，此檔就會被遺留下來。在此狀況下，當您要重新啟動 `lpsched` 程式時，要記得先移除此檔。

### 22.15.1 新增印表機

上述介紹的指令及支援的選項以「新增印表機」來瞭解其意義，是最恰當不過了。當您要新增印表機時，如 `lpsched` 程式仍在執行中，那要請您先以 `lpshut` 指令停止排序服務 (Solaris 系統不需執行此步驟)，之後才能以 `lpadmin` 程式安裝印表機：

```
# lpshut
Print services stopped
# lpadmin -p pr1 -m epson -v /dev/lp0
```

上列指令，係新增一設備檔名為 `/dev/lp0`，取名為 `pr1` 的 `epson` 型號之印表機。同一時間，會將此印表機的介面程式自 `/usr/spool/lp/model` 複製到 `/usr/spool/lp/interface` (Solaris 系統為 `/etc/lp/interfaces`)。印表機的標準介面程式是存放於 `model` 目錄中，安裝時程式會依您的指令將正確的介面程式複製到介面 (`interface`) 目錄中。因此 `lp` 程式所使用的介面程式是此複製版，而非直接應用原先系統提供的。

介面程式其實是由 `lpsched` 程式呼叫，包含如使用者名稱、列印份數、檔名及 `job-id` 等許多參數的 shell script。此 script 會將列印資料縮減、製作成接收列印請求之印表機可接受的格式後，送到標準輸出。然後此輸出會重新轉向到先前以 `-v` 選項設定的設備檔。

上述之指令設定，定義的是一部簡單的印表機。您也能加上 `-c` 選項，定義它屬於一個類別(class)。而當此類別原先不存在時，列印系統就會新建此類別：

```
lpadmin -p pr1 -m epson -v /dev/lp0 -c c1
```

印表機 `pr1` 屬於類別

您能以 **-p** 選項來新增印表機及修改印表機設定，而且許多 `lp` 列印子系統的指令，也是以此選項指定印表機名稱。現在您可以重新啟動 `lpsched` 程式，讓 `lp` 列印系統接收送到此印表機的列印請求，之後再啟動印表機：

```
# lpsched
Print services started.
# accept pr1
destination "pr1" now accepting requests
# enable pr1
printer "pr1" now enabled
```

*現在印表機已安裝完成，可供使用*

先前我們並沒有定義系統能使用的預設印表機（或預設類別），現在讓我們以 **-d** 選項來設定預設印表機：

```
lpadmin -d pr1
lpadmin -d cla1
```

*將 pr1 設為預設印表機  
將 cla1 設為預設類別*

當設定的介面程式運作不正常時，您能以 **-m** 選項更換它。假如介面程式運轉正常，在您新增同型號印表機時，您能吩咐 `lpadmin` 程式（以 **-e** 選項）直接由現有印表機複製其介面程式：

```
lpadmin -ppr1 -m HPLaserJet
lpadmin -ppr3 -e pr1
```

*採用 HPLaserJet 的介面程式  
採用 印表機 pr1 的介面程式*

在印表機沒有任何尚未處理完的工作時，您能以 **-x** 選項移除此印表機：

```
lpadmin -x pr1
```

*刪除印表機 pr1*



Note

列印時，系統首先會檢查 `LPDEST` 參數設定。當此參數未定義時，系統才會採用 `lpadmin -d` 所設定之印表機為預設印表機。假如您都沒有預設印表機的定義，`lp` 程式會拒絕所有的列印請求。

### 22.15.2 使用 `lpstat` 指令檢視印表機與列印工作狀態

`lpstat` 指令提供許多選項，以獲得印表機與列印工作的狀態資訊。如 **-r** 選項顯示是否排序程式依舊在運作中：

```
# lpstat -r
scheduler is running
```

當您不加選項時，`lpstat` 指令會顯示執行此指令的使用者送出的所有列印請求的狀態。系統管理者也能以此得到所指定的印表機上，所有等待列印工作的列表。

```
# lpstat -ppr1
pr1-323      romeo      345670      Jan 10 13:26      on laser
pr1-324      romeo      3659        Jan 10 13:30
pr1-325      juliet     23678       Jan 10 13:40
```

*也能使用 -p pr1*

列印工作狀態也能以使用者作排列 ( **-u** 選項 )，此外您亦能以 **-t** 選項檢視所有的狀態資訊：

```
# lpstat -t
scheduler is running
system default destination: pr1
device for pr1: /dev/lp0
pr1 accepting requests since Fri Dec 12 10:06:04 1997
printer pr1 waiting for auto-retry. available.
    stopped with printer fault
pr1-1                henry                1897    Dec 12 10:37
```



Note

列印請求排序的運作是以 **lpshut** 指令停止，而以 **lpsched** 指令啟動。當 **lpsched** 程式重新啟動時，所有被中斷的列印工作都會重頭再開始。

### 22.15.3 接收(accept)與回絕(reject)：佇列(Spooling)控制

許多指令能控制佇列之運作。在列印工作繁忙或印表機故障時，您會需要求助這些指令幫忙處理。您必須要能控制排入佇列的輸入，及將列印工作轉向到另外一個印表機。

您能以 **cancel** (請參閱 6.16.2 節) 指令取消任何列印工作。然而當佇列運作太過繁忙時，您應該以 **accept** 與 **reject** 指令控制到佇列的輸入。**reject** 指令能禁止列印請求再進入佇列，而加上 **-r** 選項能註明回絕的原因；當使用者嘗試要列印工作送到已設為 “rejected” 的印表機時，會看到下列訊息：

```
# reject -r "Spooler very busy" laser epson
destination "laser" will no longer accept requests
destination "epson" will no longer accept requests
```

相反的 **accept** 指令則是允許該印表機佇列運作。因此每當系統新增印表機時，就需要執行此指令。不過，要注意的是不管是 **accept** 與 **reject** 指令都不能改變已存在佇列中的列印工作狀態。

### 22.15.4 啟動(enable)與停止(disable)

**accept** 指令允許印表機佇列運作，但不代表印表機就能進行列印。在新增印表機的同時，您尚須啟動印表機，而且只要啟動一次，它就能列印。在某些狀況下，您需要關閉印表機；舉例來說，比如印表機卡紙時；**disable** 指令就能幫您停止印表機上的列印工作。

```
disable -r "Jamming of paper; just a minute" lwriter
```

不同於 **reject** 指令，**disable** 指令不能停止已送到印表機佇列的工作，或中止正在列印的工作。所有被停止的印表機之列印工作，只有再以 **enable** 指令重新啟動印表機後，才會繼續列印。

### 22.15.5 使用 lpmove 指令轉移列印工作

當印表機被停止後，您能以 lpmove 指令將列印工作移到其他印表機上。它執行的模式有兩種：

```
lpmove pr1-321 pr1-322 laser
lpmove pr1 laser
```

*移轉這兩個列印工作到印表機 laser*  
*將所有在印表機 pr1 上的列印工作移到 laser*

在第二種模式下，lpmove 指令會自動在原印表機呼叫 reject 指令。因此，這也表示印表機 pr1 在轉移指令移轉它的所有列印工作後，就不能再接收列印請求。



Linux  
注意事項

### 管理 Linux 印表機

Linux 所採用的是舊式的 Berkeley 列印系統。相較於 AT&T 系統，它所提供的功能陽春許多，而且較難被理解。列印工作是以 lpr 指令（請參閱 6.16——Linux 注意事項）與 lpd 常駐程式所執行。比起 SVR4 同樣的指令集，Linux 系統提供較少之指令：

- 使用者層級指令如 lprm（用來取消列印工作）及 lpq（用來檢視列印佇列，queue）。
- 管理層級指令如 lpc，處理所有原來在 SVR4 系統上各自獨立的印表機與佇列之運作功能。

不同於 AT&T 系統上之 lpadmin 指令，Linux 系統並不提供指令列工具來安裝印表機。它是另外提供一選單介面軟體以設定印表機。您從這些前端工具輸入的大部分資料，會存入 **/etc/printcap** 的文字檔中。此資料庫中存有所有印表機的屬性，如：印表機名稱、介面 script、佇列目錄與設備檔等等。您當然能直接編輯此文字檔，但建議您在熟悉檔案中各變數用途與程式碼作用後，才這樣作。

當使用者傳送列印請求至佇列時，lpr 會先檢視 **/etc/printcap** 檔，找到佇列所存放檔案的目錄；之後就有一複製檔存在此處。而 queue 則是受到 lpd daemon 監視。當 lpd 發現有列印工作等待執行時，它會將它自己複製一份，也就是再產生一個新的 lpd process。這個新的 lpd 就會呼叫 **/etc/printcap** 中設定的介面程式來進行列印。

列印系統中，大部分指令支援 **-P** 選項（SYSTEM V 系統中是用 **-p**）來指定目標印表機。假如指令未加上此選項，列印工作會送到系統設定的預設印表機。

以 lpq 與 lprm 指令檢視印表機與列印工作狀態

lpq 會列示印表機的 queue。假如加上 **-P** 選項，顯示的是指定印表機的 queue；如果不加，預設會列出預設印表機的 queue。



```
# lpq
lp is ready and printing
Rank  Owner      Job  Files                                Total Size
active sumit     48   (standard input)                   1972 bytes
1st    sumit     49   searchlist.html                     61061 bytes
2nd    henry     50   (standard input)                   17746 bytes
```

由 `lpq` 輸出的第一個列印工作狀態為 “active”，代表 `lpd` daemon 正在執行中。否則您會看到它顯視為 “1st”。第三欄所顯示的是列印工作的工作編號 (job number)。不同於 `lpstat` 指令的輸出，`lpq` 會顯示列印檔名；但在某些情形下列印輸入是由管線 ( | 符號，pipeline ) 所提供，那就沒有檔名而會以標準輸入 (standard input) 表示。

您可以藉由 `lpq` 指令之 `-l` 選項，得到完整的輸出狀態列表：

```
# lpq -l
lp is ready and printing
sumit: active                [job 048uranus]
      (standard input)      1972 bytes
sumit: 1st                   [job 049uranus]
      searchlist.html       61061 bytes
```

`lprm` 指令能將列印工作自印表機的 queue 中移除。一般使用者也能使用此指令，但她只能移除屬於她自己的列印請求。然而當超級使用者執行此指令，並以 `-` 為參數時，印表機上所有的工作都會被刪除。此外 `lprm` 指令也能依所給的參數，以印表機或使用者為單位，刪除所有屬於該使用者的列印請求：

```
lprm -                        刪除所有 queue 中的列印請求
lprm -Phpl                   刪除印表機 hpl 上的列印工作
lprm henry                   刪除所有屬於 henry 的列印請求
```

預設印表機是先由 `PRINTER` 變數設定；假如未設定此變數，那會改採 `/etc/printcap` 檔中所設定的預設印表機。當沒有預設印表機時，列印請求一樣會被 `lpr` 回絕。

主要的列印資料庫：`/etc/printcap`

BSD 的 `lpd` 系統以 `/etc/printcap` 當存放印表機詳細規格的資料庫。此檔案與另一個存放終端機設定的 `/etc/termcap` 檔一樣，檔案結構稍艱澀難懂。如變數要加上 `:s` 或 `#s` 有時難以詮釋明白。因此除非您確實瞭解要修改的項目，否則建議還是交由前端工具 (如 Red Hat 中的 `printtool`) 來修改此檔，至少在剛開始的設定階段。

**printcap** 檔的內容區分成數段，各段中分別存放一個印表機的相關定義。定義的起始是印表機名稱，接著是以 | 符號把所有別名劃分開來。下面就是一部 HP Deskjet 500 印表機在 **printcap** 的設定片段：

```
lp|lwriter|cdj500-a4-auto-mono-300|cdj500 a4 auto mono 300:\
:lp=/dev/lp0:\
:sd=/var/spool/lpd/cdj500-a4-auto-mono-300:\
:lf=/var/spool/lpd/cdj500-a4-auto-mono-300/log:\
:af=/var/spool/lpd/cdj500-a4-auto-mono-300/acct:\
:if=/var/lib/apsfilter/bin/cdj500-a4-auto-mono-300:\
:la@:mx#0:\
:tr=:cl:sh:sf:
```

由上可知此部印表機有數個名稱：**lp**、**lwriter** 或 **cdj500-a4-auto-mono-300**。您能在 **-P** 選項後以上述中任何一個名稱傳遞列印請求。其餘幾行以冒號隔開的敘述，每一行分別儲存一項印表機的屬性設定。定義中除了最後一行外，每行最後都以 \ 符號來跳脫 new-line。

每一項印表機屬性都是以屬性變數(*variable*)= 設定值(*value*)的格式儲存。**printcap** 檔中記錄著許多印表機控制屬性的變數，如印表機設備檔名稱、佇列(*spool*)目錄、列印的紙張大小、過濾程式與印表機列印介面等。本節中，我們將只對一些較重要的設定屬性進行解說。

本例中的 **lp** 同時是印表機名稱及屬性變數，您必須由其前後文確認它所代表的真正意義。此處 **lp** 屬性變數設定印表機設備檔為接到並列埠的 **/dev/lp0**。Linux 系統提供三個並列埠（另兩個為 **lp1**、**lp2**），因此您要確認選到正確的連接埠。

**sd** 設定的是位於 **/var/spool/lpd** 中讓 **lpr** 轉存列印檔資料的佇列（*spool*）目錄。要注意的是，佇列目錄是以印表機名稱命名。對每個列印檔，**lpr** 會於佇列目錄中建立兩個以 **cf** 及 **df** 起頭的檔案。“**df**”檔的內容是列印檔未經過濾的資料，而“**cf**”檔包含此列印請求的控制資訊。

**if** 所設定的是輸入過濾器(*input filter*)。此處它採用存放於 **/var/lib/aps-filter/bin** 目錄下介面 *shell scripts*。輸入過濾器通常會呼叫 X 視窗介面為基礎的圖形程式（如 Ghostscript 的程式 **gs**）來列印圖形輸出。其餘列印屬性變數我們就不再討論。

佇列目錄中也包括一個鎖定檔(*lock file*)以避免同時有兩個列印工作搶奪同一個印表機資源。鎖定檔中包含目前正在列印工作的“**cf**”數字及該執行中 **lpd** daemon 的 **PID**。在每個列印工作完成後，鎖定檔就會被移除；由下一列印工作產生另一新的鎖定檔取代。假如列印中不正常斷電，您必須要自行移除此檔，方能繼續列印工作。

在修改 **printcap** 檔後，您必須要重新啟動 **lpd daemon**。在 **init.d** 目錄中的 **lpd scripts** 除了 **start** 及 **stop** 外還支援額外的 **reload** 與 **restart** 參數，可以重新啟動或終止服務。而您可以直接呼叫此 **scripts** 或是使用連結的 **rc scripts**：

```
/etc/rc.d/init.d/lpd restart
/etc/rc.d/rc3.d/S60lpd restart
```

或 *reload*  
以 *rc scripts* 執行

### **lpc**：印表機控制指令

**lpc** 指令能處理所有的印表機控制設定。該指令執行時不加任何參數，會出現 **lpc>** 提示符號，進入其控制介面。您能於此提示下，輸入所有的內部指令。這些內部指令被用來

- 啟動與停止佇列運作（內部指令：**enable** 與 **disable**）。
- 啟動與停止印表機（**start** 與 **stop**）。
- 啟動與停止各服務（**up** 與 **down**）。
- 控制印表機 **daemon**（**abort** 與 **restart**）。
- 檢視佇列與列印狀態(**status**)。
- 將某列印請求移到 **queue** 的最前面(**topq**)。

而在 **lpc>** 提示符號輸入 **help**，會顯示完整的內部指令列表。此外 **lpc** 也能在非互動模式下以內部指令為參數來執行。其中的某些指令功能在 **AT & T** 系統上也相當常見，因此我們只對它們作一簡單的介紹。

除了 **topq** 之外所有的指令，都能以印表機名稱當作可選擇的參數。假如執行時未提供，就會採用預設印表機。**System V** 使用者要特別注意 **lpc** 程式是以啟動(**enable**)與停止(**disable**)方式，控制佇列(**spooling**)及停止列印。下面我們將以 **shell** 命令列模式與 **lpc>** 提示符號來說明：

```
# lpc enable lwriter
lpc> disable lwriter
```

*Shell 提示字元*  
*lpc 的提示符號*

**start** 與 **stop** 內部指令控制列印工作執行（如同 **AT & T** 系統之 **enable** 與 **disable**），但多加了控制 **lpd** 的啟動與停止功能。如要同時控制佇列與列印運作，您就必須應用 **up** 與 **down** 指令。類似 **AT & T** 系統的 **reject** 指令，**down** 指令一樣能附加訊息當作輸入之參數：

```
up lwriter
down lwriter "Printer is down"
```

同時包含 **stop** 與 **disable** 功能

而您只能在限定的模式下更改列印工作之優先權；也就是說，您只能將列印工作移到 **queue** 的最前面，而不能移到其他地方：

```
topq lwriter 52
topq lwriter henry
```

將列印工作 52 移到最前面  
將 *henry* 的列印工作移到最前面

藉 **status** 指令能顯示您列印系統的目前狀態。任何使用者皆可使用此指令：

```
lpc> status lwriter
lwriter:
    queuing is enabled
    printing is enabled
    no entries
    printer idle
```



Tip

要立即終止目前列印中的工作，lpc 的 abort 即能符合您的需求。

## 摘 要

超級使用者帳號被用來作系統管理。此外，任何使用者也都能藉由 su 指令，獲得超級使用者權限。大多數系統管理程式存放在 **/sbin** 與 **/usr/sbin** 目錄中。超級使用者帳號的路徑(PATH)變數中，沒有包括目前目錄(current directory)。

管理者能修改任何檔案的屬性、終止任何的 process 和更動任何一位使用者之登入密碼。她也能設定系統時間、以 wall 指令同時對所有使用者廣播訊息、利用 calendar 程式提示使用者約會時程以及應用 ulimit 指令設定檔案大小限制等。

使用者帳號能以 useradd 指令新增、usermod 指令進行修改及以 userdel 指令自系統中移除。使用者的詳細資料是存放在 **/etc/passwd**、**/etc/shadow** 與 **/etc/group** 檔案中。登入密碼則是以加密方式存在 shadow 檔內。在某些系統上的使用者能以 chsh 指令自行更改登入 shell。

檔案的 *set-user-id* 位元(SUID)能允許使用者在執行此檔時，暫時獲得檔案所有者之權限。使用者能藉由使用這些指令來修改系統檔案或將資料寫入系統目錄中，而不是讓使用者直接動作。

一般檔案具有 *sticky* 位元時，表示此檔長期被放在置換區域(swap area)。當目錄含有此位元時，任何使用者可在於此目錄中新增、刪除屬於自己的檔案。許多 UNIX 系統目錄如 **/tmp** 與 **/var/tmp** 都設有此位元，這也就是為什麼使用者能對這些目錄讀寫，卻沒有違反安全控管的疑慮。

限定(restricted)shell 的設定，讓使用者不能更換他的目錄、執行其他目錄中的指令甚至不能新增檔案。在此情形下，系統指令要在使用者無法修改的目錄中，以建立連結的方式讓他使用。

init process 控制系統在指定的 run levels 中執行，以及在每部終端機上產生 getty，讓使用者能登入系統。init 由 **/etc/inittab** 檔讀取它的指令，並負責在指定的 run level 上執行對應的系統 daemons(processes)。Run level **2**或 **3** 代表系統

在一般正常的多使用者操作模式，而1或s則是代表在單人使用(single-user)環境。

關機程式shutdown在關閉系統電源前，先終止所有的 processes，並將所有檔案系統卸下。此外，您也能以 init 指令加上 run level 為參數（0或6）將系統關機。Linux 系統可同時按下 [Ctrl][Alt][Del] 三鍵來進行關機。

軟碟片藉由 format 或 fdformat 指令來格式化。dd 程式則是利用 raw device進行複製軟碟或磁帶資料。UNIX 系統提供一整套的指令來操作 DOS 軟碟片。這些指令在System V 中是以 dos 開頭，Linux 中則是以 m 開始。

cpio 程式由標準輸入取得一組列表檔案，並集合它們成為一個備份資料檔。它非常適合 find 指令加上 | 符號形成管線來使用。它使用一些主要的選項，例如將資料複製到儲存媒體時，要加上(-o)選項、回存時用(-i)選項以及(-it)選項顯示存放之資料。另外在預設狀況下，除非加上 -u 選項，不會覆蓋舊有檔案。而萬用字元能被用來回存一組的檔案。

tar 程式特別適用於備份整個目錄樹。它也提供主要選項，例如以(-c)選項複製資料到儲存媒體、從儲存媒體萃取資料用(-x)選項及以(-t)選項檢視儲存內容。此外它亦提供（-r 與 -u）選項，讓資料能附加在原有之備份資料中，及保存不同版本之檔案。而 GNU tar 程式尚支援（-z 與 -Z）壓縮選項。

du 指令能顯示每一使用者目錄樹的空間使用狀況。管理者能利用 find 指令，找出那些長時間不曾被使用過的檔案。而 find 指令更能與 xargs 合併使用強化其原有的特色，xargs只要執行指令一次或限定執行次數，不用在每次找到檔案就執行一次。

## 進階篇

要增強系統安全，管理者能以 passwd 指令可設定密碼使用限期(password aging)，強迫使用者於限定時間內更換密碼（-x選項）。使用者帳號可以被鎖定（-l 選項）或強迫使用者於第一次登入系統時更換密碼（-f選項）。

系統 processes 是由/etc目錄中之 rc scripts 所啟動，其餘 scripts 則是存放在 /etc的子目錄下。特定的 run level 要被執行的scripts會被分開存放在不同的目錄中。“K”字元起始之 scripts 是用來終止 processes，而以“S”起始的scripts 則是負責啟動它們。所有的 rc scripts 都在建立連結指向 /etc/init.d 目錄下對應的 script。

終端機定義是由 getty 從 /etc/gettydefs 檔取得。您能設定此檔，讓 getty 自動依序取得合適參數。預設的終端機類型，大多數系統都是存放於 /etc/tty-type 檔中。您能在您的 PC 上以虛擬終端機建立許多登入連線，之後可在各虛擬終

端機上執行程式，而能有各自獨立的顯示。

在 System V 系統上，`lpsched` daemon 負責監視與進行列印工作排序。`lpadmin` 指令是用來新增、修改及設定預設印表機。`lpstat` 指令則是顯示列印請求的狀態。`accept` 讓列印 queue 準備好，以接收列印請求；`reject` 則相反的禁止所有列印請求送進佇列中。印表機需要先啟動(enable)才能列印，但有時會需要被停機(disable)。在 queue 上的列印工作能以 `lpmove` 指令移轉到其他印表機。

Linux 系統採用 BSD 列印系統。`lpd` 是監視列印queue 的印表機 daemon。`lpq` 指令能顯示出目前排序(queue)狀況，而 `lprm` 指令則是將工作請求自 queue 中移除。印表機詳細規格設定存放於 `/etc/printcap` 檔。列印與佇列管理是屬於 `lpc` 指令負責。您能啟動與停止印表機 (`start` 及 `stop`)，及啟動佇列或關閉其運作 (`enable` 與 `disable`)。列印工作能以 `topq` 指令將它移到 queue 的頂端。

UNIX 列印程式傳遞列印資料給介面程式，即依據印表機設定將列印資料格式化的過濾器。過濾器通常會再呼叫外部程式，以進行實際的列印工作。

---

## 自我測驗

- 22.1 請問系統管理指令通常存放的目錄為何？
- 22.2 請問當超級使用者執行 `calendar` 指令時會出現什麼變化？
- 22.3 請問密碼存放於何處？
- 22.4 請問管理者如何對所有使用者廣播訊息？
- 22.5 請您新增使用者 `john`，設定 UID 為 **212**、屬於 `dialout` 群組。此外 `john` 尚希望使用 `bash` shell 及將目錄放在 `/home` 之下。
- 22.6 請問如何用超級使用者帳號，更動您使用的 shell？
- 22.7 請問系統核心程式(UNIX Kernel)的名稱為何？
- 22.8 請問您如何決定系統預設的 `run level`？
- 22.9 Linux 系統最快速的關機方法為何？
- 22.10 請問如何以 `shutdown` 指令立即關閉系統？
- 22.11 請問 `init` 的兩個重要功能為何？
- 22.12 請問 `run level 0` 與 **6** 有何不同？
- 22.13 當系統處於單人使用模式時，您能列印檔案嗎？
- 22.14 在修改 `/etc/inittab` 檔之後，您需要立即做哪些處理？

---

## 練習

- 22.1 請問當超級使用者執行 `passwd` 指令時，會出現什麼變化？

- 22.2 請問哪個變數會以讀取 `/etc/passwd` 檔的第一個欄位資料來設定？
- 22.3 telnet 有時不允許用 root 帳號登入。請問您如何從遠端主機上，修改一個只有 root 才能寫入的檔案？
- 22.4 請問為何系統將編碼的密碼部分，由 `/etc/passwd` 檔移至 `/etc/shadow` 檔中？
- 22.5 請問您如何不使用passwd 指令，防止使用者 kathy以她的帳號登入系統？
- 22.6 請問何種狀況下，使用者登入後不能更換目錄或在其使用者目錄中新增檔案？
- 22.7 請問使用者如何能用passwd指令更新 `/etc/shadow` 檔，即使他並沒有寫入權限設定？
- 22.8 請問在檔案權限設定欄位出現 `t` 字元代表為何？
- 22.9 請問如何限定一群使用者能讀寫同一目錄，但不能刪除他人之檔案？
- 22.10 請問如何獲得相同於超級使用者權限，來新增使用者？
- 22.11 請解釋此指令的用途：`find / -perm -4000 -mount -print`
- 22.12 請問 `run level 1` 意義為何？請舉出四個在此狀態下無法執行之動作？
- 22.13 請問系統管理員如何指定在上班時間的早上九點到晚上十點間，每小時自動監視、回報硬碟可用的空間？
- 22.14 請寫一 shell scripts 實現DOS 下 DISKCOPY 相同的功能。
- 22.15 請問在 SVR4及 Linux 系統上，您要如何將所有 HTML 檔案複製到 DOS 軟碟上？
- 22.16 假如您預定要備份今天工作的成果，首先在您登入系統後要做的事為何？要如何以 cpio 指令進行備份？
- 22.17 請將 `/usr/local/bin` 目錄以 tar 指令備份成一硬碟檔案，再以 gzip 程式進行壓縮。
- 22.18 以tar指令將遠端主機 *Jupiter* 的 `html doc` 目錄樹備份，並將備份資料檔 (archive)建立在您的本地硬碟上。請問何時此指令會無法使用？

## 進階篇

- 22.19 請問密碼使用限期(password aging)資訊存放於何處？您如何確認使用者每四星期都會更換密碼？
- 22.20 請以 passwd 指令，禁止使用者 kathy 繼續使用她的帳號。
- 22.21 請問啟動(start)與終止(kill)scripts 的運作意義為何？
- 22.22 假如終端機沒有顯示登入提示，您首先要檢查的部分是什麼？
- 22.23 SVR4 系統中存放速度及終端機設定的檔案為何？您如何確認終端機是以唯一的速度連接？

734 UNIX終極指南

- 22.24 請設計一檢視印表機 `daemon` 是否有正常執行的確認步驟，可應用在不論是 SVR4 或 Linux 系統中。
- 22.25 請解釋何謂介面 `script`？
- 22.26 請問要列印的資料是否會轉存到佇列目錄中？如果有，那又是存到何處？
- 22.27 即使 `lpsched` 正常運作中，您卻無法使用印表機 `laser1` 列印。您會先嘗試哪個指令來檢視問題？
- 22.28 請問 SVR4 系統中的 `accept` 與 `enable` 指令，被用在列印工作上的差異為何？



