

## » 本書特色

- 本書以左右兩頁為單位完成單一主題的闡述，讓讀者印象集中不分散。這也能方便讀者之後搜尋所需要的內容。
- 當然，本書內容除了解說之外，也盡量大量提供具體的指令輸入與執行範例。但有些環境並無法使用本書所介紹的工具，或是輸入指令後產生的結果有所不同，敬請讀者見諒。
- 此外，本書出現的 CUI 介面，是在書中亦有介紹的虛擬環境 VirtualBox 上建構。相較於 Windows 等作業系統，周邊設備的支援未盡完善，依據所使用的電腦等環境可能出現無法使用的情況。周邊設備具體的支援狀況請洽個別供應商。
- 附錄中收錄了 VirtualBox、CentOS、Ubuntu 的安裝步驟，與書中出現指令的主要選項（option）等資料，在讀完本書並實際作業時，敬請多加利用。

## » 對象讀者

本書以 Linux 初學者為對象。只聽過 Linux 這個名詞，以及工作場所中雖具備相關環境卻未曾接觸者，請務必翻閱本書。另外，對於嘗試要接觸工作場合中相關環境，或是希望在自己電腦中導入 Linux 卻感到挫折的讀者，本書應該也能有所助益。

## » 關於標記

本書的格式規範如下。

### 【範例與執行結果】

【範例與執行結果】

```
pwd
```

實際顯示於畫面之內容

結果

```
/home/shiori/data
```

### 【字體】

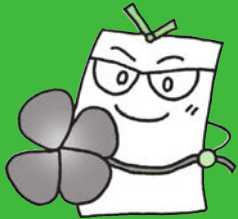
**黑體**：重要詞彙

List Font：指令輸入時實際使用的句子與詞彙

**List Bold Font**：指令輸入範例與執行結果中的重點

### 其他

本書以 CUI 介面中的指令輸入為基礎展開討論。有些發行版在安裝後會直接變成 GUI 介面，參考本書之前請先轉換為 CUI 介面。



學習 Linux

# 開始之前



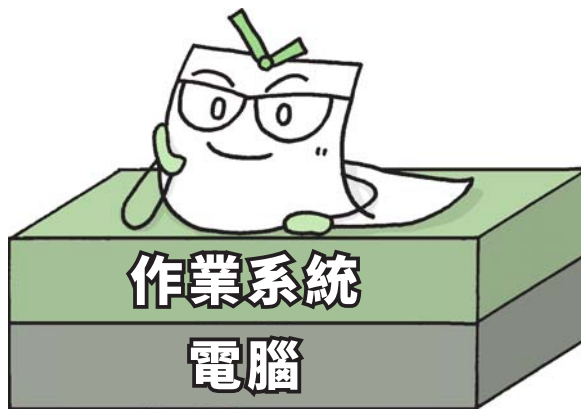
## 作業系統的角色

在日常生活中，我們常會聽到「Windows」和「Mac」等電腦詞彙。「Windows」和「Mac」所代表的是作業系統（OS，Operating system）的種類。

作業系統指的是讓電腦（不限於個人電腦，而是包含伺服器等所有計算機）運作所需要的軟體。Windows 和 Mac 會隨著版本更新而升級，所有有些人可能覺得作業系統可以輕易更新，或是沒有特別意識到這過於理所當然的存在。但作業系統與應用程式卻是截然不同。

文書處理軟體與電子試算表等都是透過作業系統與電腦連結。應用程式是我們因應目的執行操作時所需要的工具，這些工具需要有作業系統才能使用。就像是銀行的 ATM 需要作業系統，智慧型手機與平板也安裝有專用作業系統一樣。作業系統在電腦的使用上不可或缺，扮演著非常重要的角色。

各位將在本書中學習的 Linux 也是作業系統的其中一種。



## 關於 Linux

如今說到一般使用者最有機會接觸到的 UNIX，那就非 **Linux** 莫屬了。Linux 是由 Linus Torvalds 所開發的一種 PC-UNIX，他在開發當時是芬蘭的研究生。Linus Torvalds 所製作的是作業系統的「核心 (kernel)」(請參考第 8 頁) 部分，後來成為開源軟體，原始碼免費公開，現在世界各地的使用者也在持續改進中。

### » 開源軟體

將原始碼 (程式的根本) 公開，任誰都能自由使用，就稱為**開源軟體**。以企業所開發的軟體來說，為了確保利益，一般不會公開原始碼。也因此企業不會執行不符利益的開發，相對的也會在責任範圍內對開發軟體給予豐富的支援。相較於此，開源軟體只要受到使用者的支持，就會擁有迅速、有效改良，以及持續開發新功能等優點。但另一方面則可能無法提供周到的支援與服務。

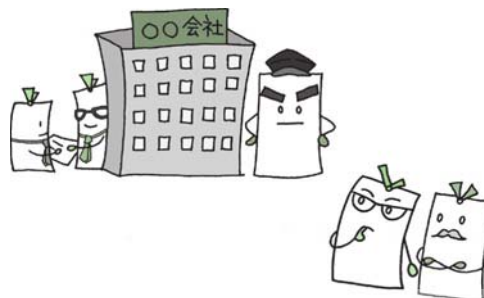
#### 開源軟體

- 只要從使用者的角度看來覺得方便，就能迅速實現。
- 容易得知使用者的評價
- 無法期待與企業同等程度的支援



#### 企業開發

- 以企業利益為優先
- 有時難以得知使用者評價
- 充沛的支援

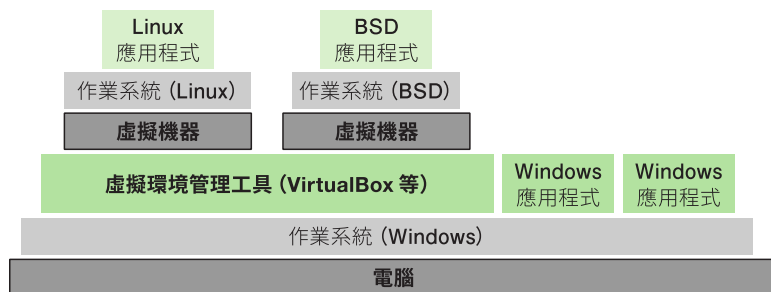




## 虛擬環境

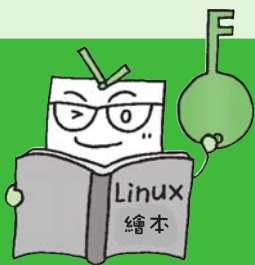
如果身邊沒有 Linux 的環境，就必須自己建立。除非有特殊情況，否則即便存取並使用了 Linux 伺服器，也不太會有機會獲得管理員的權限。然而，準備一台新電腦並導入 Linux 的過程相當繁複，也要花費購置的費用。這種情況下可以考慮在 Windows 中建置 Linux 的虛擬環境。

虛擬環境指的是在作業系統上模擬（emulate）電腦的硬體，以軟體的方式建立電腦。以這種方式建立的虛擬電腦就稱為虛擬機器（或是虛擬 PC、虛擬伺服器）。Linux 則安裝在這台虛擬機器上。由於虛擬環境容易營造，在實際應用中，也被使用於可接受低處理效能的情況。另外，由於虛擬環境是在電腦中重現電腦環境，建構虛擬環境的電腦必須具備許多的磁碟容量、記憶體，以及良好的 CPU 效能。如果電腦效能低落，可能會出現虛擬環境中的作業系統處理速度過慢，難以實際使用，或是根本就無法建構虛擬環境等情況，務必要注意。



只要效能足夠，在一台電腦中建立多個虛擬環境也沒問題。





## 第 1 章 學習重點！

# key



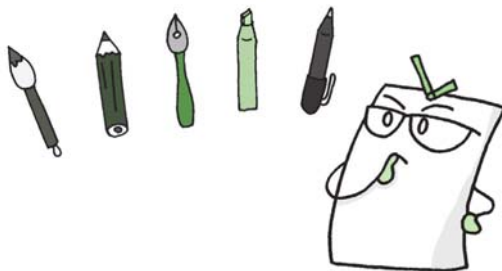
### Linux 的規則

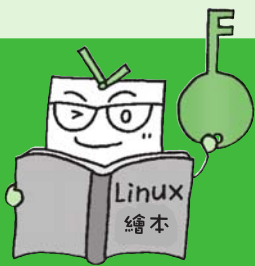
Linux 具有一些特別的操作規則。首先，請記得**登入** (log in 或 log on) 與**登出** (log out 或 log off)。Windows 系統只要開啟電源就可以自動登入，關機時也會自動登出，連關閉電源都自動完成。但是，Linux 系統對於「是誰從什麼時候開始使用？什麼時候結束？」的管理相當嚴謹。因此，登入與登出成為不可或缺的手續。此外，如果要關閉電源或是重新啟動，管理員也必須執行特別的指令。這是因為 Linux 原本是被設計為以多人使用為前提的伺服器作業系統，因此，才會讓一般使用者無法任意關閉電源。



### 核心與殼層

**核心** (kernel) 指的就是作業系統最基本的部分，**殼層** (shell) 則是讓作業系統更容易使用的介面。如果沒有殼層，就無法將我們的想法傳遞給核心。就像是寫信時要選擇原子筆、鉛筆或是鋼筆等書寫用具一樣，我們可以依據目的與使用方便性，從各種殼層中做選擇。





## 第 2 章 學習重點！

# key



## Linux 的基本操作

在第 2 章當中，除了操作 Linux 時的注意事項與基礎知識以外，會先介紹需要熟記的指令。

對於尚未熟悉 CUI 環境者，在操作上感到「困難」的原因，舉例來說有「在視覺上難以理解自己到底在什麼地方，及打算要做什麼」。在第 1 章也曾介紹，Linux 系統中，所有的檔案都儲存在從根目錄所衍生的目錄中，而各個目錄都是以樹狀結構的形式受到管理。

如果能像 Windows Explorer 一樣一邊看著樹狀結構一邊操作，那倒沒什麼問題，可惜這在 CUI 的環境中是行不通的。於是使用者必須要在自己的腦中一邊想像樹狀結構才能進行作業。

在 Linux 系統中不管是執行指令或者瀏覽檔案，全都需要使用**路徑**（通往目標資料的途徑）給予指示，因此「正確書寫路徑」就成了操作上的一大前提。本章將提供幾個例子來介紹路徑的書寫方式。請一邊在腦中模擬樹狀結構，以學習路徑的書寫方式。



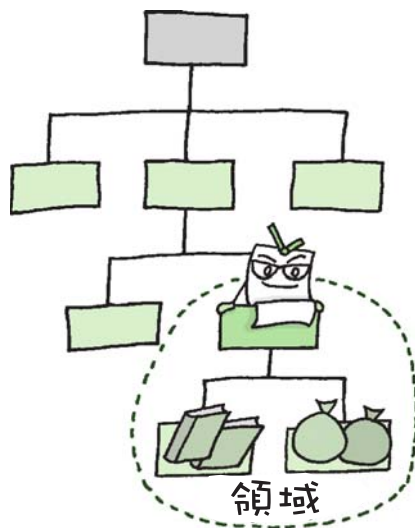


## 輸入指令的基礎概念

另外，本章的後半部將介紹幾個經常使用的指令。讓我們使用簡單的指令，從習慣 Linux 的操作開始嘗試吧！無論什麼事情，實際體驗都是很重要的。

只是，若不謹慎使用移動或是刪除檔案的指令，可能會導致問題產生。一開始可以先從瀏覽檔案與目錄的資訊開始嘗試。接下來也會介紹幾個搜尋指令，可以試著搜尋系統中有什麼樣的檔案與目錄。

一般來說，非管理員的使用者只能使用自己的家目錄。這是為了避免使用者在不經意下進到各個目錄，將其他使用者的重要資料刪除或改寫。基本上請記得一個概念，「除了管理員，不屬於家目錄及其「子目錄」的檔案，請盡量不要操作」。



1

開始使用  
Linux



2

基本的控制

3

掌握編輯器的  
運用

4

進一步運用  
Linux

5

管理系統與  
使用者

6

開始使用  
GUI

7

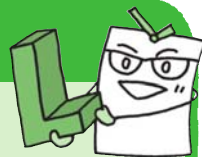
中文環境

8

進階操作

9

附錄



## ~萬用字元~

一聽到萬用字元心裡馬上就有個底的人，可能對於輸入指令相當熟悉，或是經常使用 Windows 的搜尋功能與搜尋引擎。

舉例來說，如果「想要找出所有檔名開頭為 a 的檔案」時，可以使用「\*（星號）」，寫法如下。

```
find a*
```

「a\*」的意思是，「如果第一個字母是 a，則後面是什麼字都可以」。「\*」這種「可以置換成任何文字」的特殊符號就稱為**萬用字元**，它具有像撲克牌鬼牌一樣的功能。

萬用字元除了「\*」之外還有幾個種類。例如「\*」表示任意 0 個以上的字元，而「?」表示的則是任意一個字元。因此輸入如下指令後，就會以 find 指令搜尋以 a 為開頭，2 字元的字串。

```
find a?
```

萬用字元可以附加在搜尋字串的前或後，也可以將其夾在字串之中（例如：`*sample`、`pict?data` 等）。

如果好好運用的話，萬用字元是相當方便的，可以把它記起來！

其他主要萬用字元

符號	功用	寫法範例	意思
[]	[] 中的任 1 字元	[dog]	d、o、g 這三個字
		[a-z]	從 a 到 z 的所有字母
[^] [!]	[] 中字串以外的任 1 字元	[^dog]	d、o、g 以外的文字
		[!dog]	
{ }	{ } 中個別的指定字串	{dog, rat}	dog、rat 這兩個字串





# vi 的基本操作 (2)

介紹刪除 (剪下)、複製、貼上的方法。



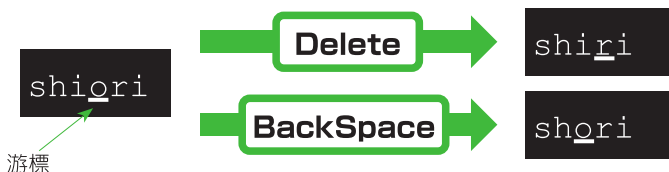
## 刪除 1 個字元 (剪下)

要刪除 (剪下) 1 個字時有幾種方法。

### » 不使用指令的方法

不使用指令，而是用 [Delete] 鍵與 [BackSpace] 鍵刪除。

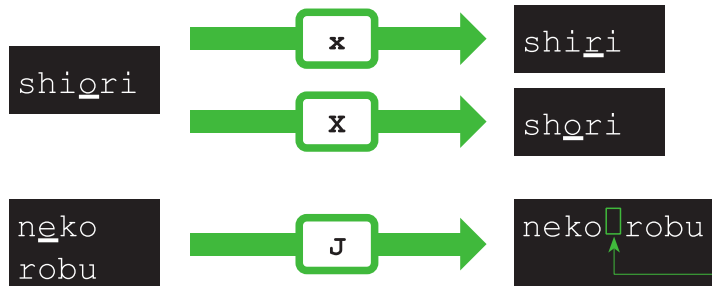
鍵	可用模式	作用	游標移動方向
Delete	插入模式 指令模式	將有游標的文字刪除一個字元	不移動
BackSpace	插入模式	刪除游標左邊一個字元	向左移動



### » 使用指令的方法

使用以下指令，每次可以刪除 1 個字。

指令	作用
x	刪除有游標的文字
X	刪除游標左側文字
J	取消游標所在行的換行



不要忘記改為指令模式喔！



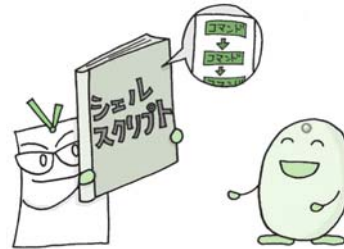
會產生半形空格



## 從「操作」端到「製作」端

在以多名使用者同時使用為前提的 Linux 系統中，即使只是取用一個檔案，對於「誰可以做什麼樣的使用」也需要確實管理。如果任誰都能自由改寫系統的設定檔，那麼後果可能不堪設想。因此，才會透過**權限**這樣的機制限制可以使用檔案的使用者。到目前為止，我們所學的操作是「執行一個指令並得到結果」。不過這樣一次就只能給出一個指令，無法進行複雜的操作。於是才會出現一種機制，能夠整合並執行數個指令，稱為 **shell script**。

shell script 是整合了一個以上指令的文字檔。如果事先寫入執行順序，只要給一次指令，就能執行多個動作。如果將經常執行的動作寫為 shell script，就可以省去重複執行的冗長程序。這個功能也能用於不同條件的複雜處理，不過本書將介紹的是基本寫法。



本章的最後將介紹學習 Linux 時的重要關鍵字之一，那就是**程序**（process）。在程式處理中，程序是 1 個單位的操作。Linux 藉由同時管理多個程序，實現了**多工處理**的系統。而程序管理在系統管理上也是相當重要的一環。

如同最開始的說明，之後介紹的主題內容將更深入。也會出現許多不熟悉的詞彙，就依照自己的步調慢慢閱讀下去吧！

1

開始使用  
Linux

2

基本的控制

3

掌握編輯器的  
運用

4

進一步運用  
Linux

5

管理系統與  
使用者

6

開始使用  
GUI

7

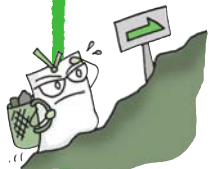
中文環境

8

進階操作

9

附錄



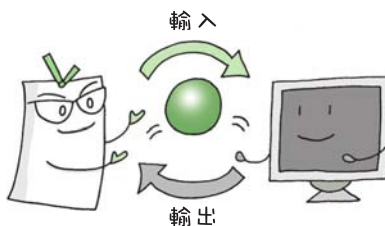
# 標準輸入與標準輸出

思考輸入與輸出。



## 輸入與輸出

將資料給予電腦，稱為「**輸入**」。而電腦回傳處理的結果，則稱為「**輸出**」。

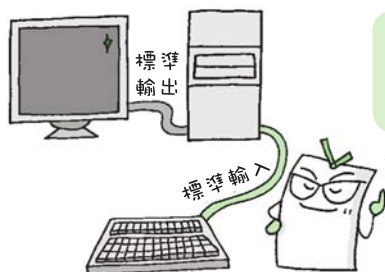


將資料輸入到指令中，所得到的結果也稱為輸出。



## 標準輸入與標準輸出是？

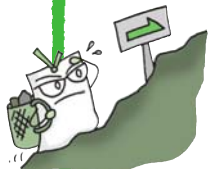
一般使用的輸入方法稱為**標準輸入**，輸出方法則稱作**標準輸出**。一般來說，標準輸入的裝置為鍵盤，標準輸出的裝置則為顯示器。



輸出、輸入時，最少會需要一組鍵盤與顯示器

### 參考

像 cat 與 grep 這樣指定查詢內容（檔名等）的指令，如果省略指定動作，通常會變成等待標準輸入（鍵盤）輸入資料的狀態（請參考 32 頁）。



# 權限 (1)

介紹權限的設定。

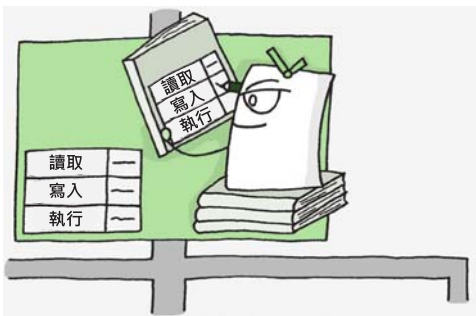


## 權限是什麼？

權限是使用者與群組對於檔案與目錄所擁有的權利。也稱為**存取保護模式**、**權限屬性**、**存取權限**。

### » 三種屬性

權限分為**讀取**、**寫入**、**執行**（包含移動與搜尋目錄等）這三種屬性。要分別對這三個屬性設定「可以」或「不行」。



### » 三種使用者類型

可以對以下三種身分個別指定讀取、寫入，與執行的屬性。

