

對本書的讚譽

即使在數位時代，《Linux 隨身指南》仍是每位 Linux 玩家案頭必備的書。它就像是我常常回頭參考，蒐集到書籤的線上手冊，但是更淺顯易懂、方便參閱。

—— *Abhishek Prakash,*
It's FOSS 共同創辦人

Linux 環境其中一個很棒的功能就是各式各樣的小工具以奇妙的方式結合在一起來解決問題。本書將其濃縮成容易理解的參考資料。即使是有經驗的老手也會在最愛的工具中重新發現被遺漏的面向與難以置信的選項。

—— *Jess Males,*
TriumphPay DevOps 工程師

這是超方便的參考手冊！它以某種方式做到了既全面又簡潔。

—— *Jerod Santo,*
changelog.com

首要之事

歡迎來到 Linux ！如果你是新手，本書能提供常見與實用指令的快速簡介和指南。如果你是老手，可以直接跳過簡介的內容。

本書有什麼內容？

本書是簡短的指南，而非完整的參考手冊。我介紹了 Linux 重要、有用的部分，讓你可以高效率地工作。然而我不會介紹每一個指令和選項（如果你的最愛被忽略了，很抱歉），也不會鑽研作業系統內部原理細節。簡短、甜蜜、必要 —— 這是我們的座右銘。

我會專注於指令，那些你在命令列上叫 Linux 做事所輸入的討厭小字。下面是一個計算 *myfile* 檔案文字行數的範例指令：

```
wc -l myfile
```

本書涵蓋大部分使用者所需之重要 Linux 指令，例如 **ls**（列出檔案）、**grep**（搜尋文字）、**mplayer**（播放聲音和影片檔）與 **df**（計算磁碟可用空間）。我只會簡單介紹圖形化環境，例如 GNOME 與 KDE Plasma，它們每一個都可以寫一整本的隨身指南。

我依功能整理了本書內容以提供簡潔的學習路徑。例如，為了協助你查看檔案內容，我一起介紹了許多檔案查看指令：看簡短文字檔的 **cat**、看

較長檔案的 `less`、看二進位檔的 `od` 等等。然後我會輪流解釋每個指令，簡短地介紹常見用法與選項。

我假設你能使用 Linux 系統，而且可以登入。如果不行，要在大部分電腦試用 Linux 也很容易。只要下載並安裝「Live」Linux 發行版到隨身碟上，並開機即可。例如 Ubuntu (<https://oreil.ly/ralRq>)、Fedora (<https://oreil.ly/Y3QGZ>) 和 KNOPPIX (<https://oreil.ly/Byqeu>)。

第四版有什麼新增內容呢？

新指令

本版增加了 50 個新指令，例如用於版本控制的 `git` 與 `svn`，文字操作的 `split` 與 `column`，檔案轉換的 `pandoc` 與 `ffmpeg`，套件管理的 `snap` 與 `flatpak`，酷炫之儲存管理的 `mdadm`、`lvcreate` 與 `zfs`，加密的 `gpg` 等等。

更清晰的架構

我以觀念重整本書章節，分成檔案、基礎系統管理、網路等主題。

舊指令再見

前一版有些指令現在已經過時了，像是 `write` 與 `finger`，或已經棄用了，例如 `ftp`。我用現代 Linux 系統的相關指令來替代。

本書指令慣例說明

本書介紹的每個指令都以標準的標題起頭。圖 P-1 是 `ls` 的標題，它是列出檔案屬性和名稱的指令。這個標題以簡單的格式示範了指令的一般用法：

```
ls [options (選項)] [files (檔案)]
```

表示在「`ls`」後輸入選項和檔名（如果需要的話）。不要輸入中括號「`[`」和「`]`」——它們只是表示其內容是選擇性的。斜體字意謂著要輸入自己的值，像是實際的檔名。如果你看到選項或引數間有垂直線，或許還以括號分組：

(檔案 (*file*) | 目錄 (*directory*))

這表示是選擇性的：可以提供檔名或目錄名稱當作引數。

```
ls      stdin  stdout -file --opt --help --version

ls [ 選項 (options) ] [ 檔案 (files) ]
```

圖 P-1 標準指令標題 (Standard command heading)

圖 P-1 的標準指令標題也包括指令的六個屬性，以黑色（支援）或灰色（不支援）列印：

stdin

此指令預設會從標準輸入讀取（也就是你的鍵盤）。請見第 22 頁的「輸入、輸出與重新導向」。

stdout

此指令預設會寫入標準輸出（也就是你的顯示器）。請見第 22 頁的「輸入、輸出與重新導向」。

-file

將單減號引數（-）當作輸入檔名使用時，會告訴指令要從標準輸入讀取，而非磁碟檔案。同樣地，如果是當作輸出檔名使用時，指令會寫入標準輸出。舉例來說，以下的 `wc` 指令會讀取檔案 *myfile* 和 *myfile2*，然後是標準輸入，再來是 *myfile3*：

```
wc myfile myfile2 - myfile3
```

--opt

雙減號選項（--）表示「選項結尾」：任何命令列上出現於其後的字串都不會被當作選項。有時候雙減號要用來和減號開頭的檔名一起運用，以免檔名被誤用作為選項。例如，如果你有一個名為 *-dashfile* 的檔案，指令 `wc -dashfile` 會執行失敗，因為字串 *-dashfile* 會被當作（不合法的）選項。請執行 `wc -- -dashfile` 以表示 *-dashfile* 是檔

名。如果指令不支援「--」，你仍然能夠將目前目錄路徑「./」置於檔名前，作為變通方法以讓減號不再是第一個字元：

```
wc ./-dashfile
```

--help

選項 **--help** 能讓指令印出使用方式說明的求助訊息，然後會退出指令。

--version

選項 **--version** 能讓指令印出版本訊息，然後退出。

指令、提示符號與輸出結果

Linux 命令列（或稱為 *shell*）在等待指令時，會印出稱為提示符號（*prompt*）的特殊符號。本書中，提示符號會是向右的箭號：

→

提示符號有各種形狀和大小，由你的 *shell* 如何設定來決定。你的提示符號可能是錢符號（\$），結合你的電腦名稱、使用者名稱和不同的符號（*myhost:~smith\$*）或其他東西。每個提示符號都表示一樣的事：*shell* 已經準備好接受下一個指令了。

當我在本書示範命令列時，有些部分要讓使用者自行輸入，其他部分則不用（像是提示符號和指令的輸出結果）。我用粗體表示要輸入的部分。有時候我會用註解來解釋怎麼回事：

```
→ wc -l myfile           在提示符號下要輸入的指令  
18 myfile                產生的輸出結果
```

你的好朋友，echo 指令

在許多範例裡，我會以 **echo** 指令印出資訊到螢幕上，我會在第 231 頁的「螢幕輸出」做正式解釋——它只會印出它的引數到標準輸出，而引數會先經過 *shell* 處理：

```
→ echo My dog has fleas
My dog has fleas
→ echo My name is $USER      shell 變數 USER
>My name is smith
```

必要觀念

什麼是 Linux ？

Linux 是自由、開放原始碼的作業系統（operating system，OS），是 Microsoft Windows 和 Apple macOS 的替代方案。Linux 驅動了網際網路上大部分的伺服器。它在每一台 Android 手機和 Chromebook，以及上百萬台網路裝置，例如路由器、防火牆和機器乳牛擠奶系統（我認真的）背後運作。它也在桌機和筆記型電腦運作得很好。

Linux 有四個主要部分，如圖 1-1 所示：

核心（*kernel*）

低階軟體。它能控制硬體和像是行程排程與網路這樣的基本功能。很少使用者會直接與核心互動。

應用程式

上千種用作檔案處理、文字編輯、軟體開發、網頁瀏覽、聲音、影片、加密、數學等你叫得出的程式。這些程式會和核心對話。在命令列執行的程式稱為指令（*command*）。

shell

一種執行指令和顯示結果的 Linux 程式。Linux 有各式各樣不同功能的 shell。本書聚焦於一個叫做 bash 的 shell，常常是使用者帳號的預設 shell。其他 shell 有 dash、fish、ksh（Korn shell）、tcsh（TC shell 或 T shell）、zsh（Z shell）和較少見的 busybox。雖然使用方法迥異，但所有的 shell 都有相似的功能。

圖形化桌面環境（graphical desktop environment）（選擇性的）

有視窗、選單、圖示、滑鼠支援和其他熟悉的 GUI 元件的使用者介面（UI）。較受歡迎的環境是 GNOME 和 KDE Plasma。大部分為 GNOME 打造的應用程式也可以在 KDE 執行，反之亦然。¹

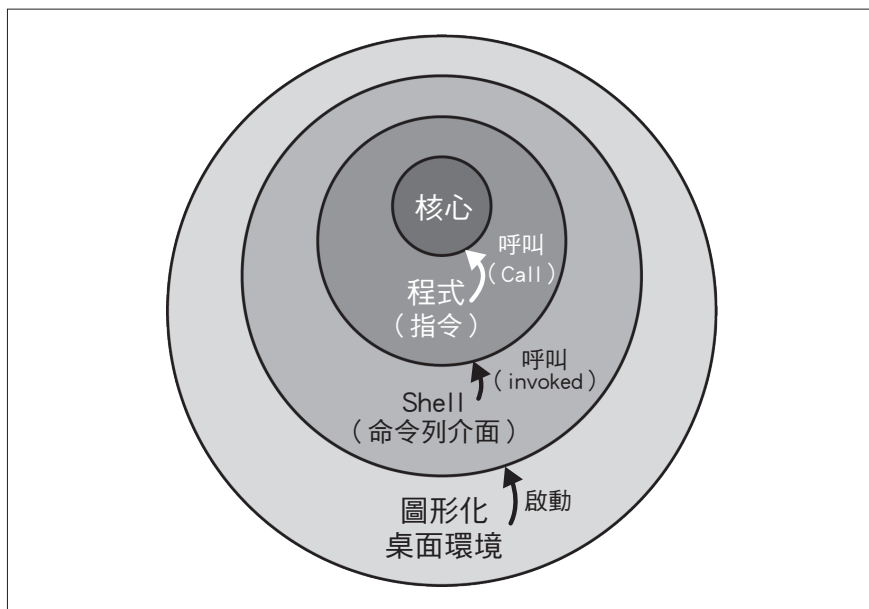


圖 1-1 Linux 在觀念上的四個主要部分。低階核心函式被程式呼叫（called），程式在 shell 裡被呼叫（invoked），shell 能由圖形化桌面啟動

1 GNOME、KDE 和其他環境都建構在共同的視窗系統上，一個是 X 或是 Wayland。要看你是用哪個系統，請執行 `echo $XDG_SESSION_TYPE` 指令。

本書聚焦於 Linux 的命令列部分，即 `shell` 及提供的程式。Windows 和 macOS 也有命令列介面（Windows 的 `cmd` 和 `powershell`，macOS 的終端機），但它們大部分的使用者都只用 GUI，可能從未看到或需要命令列。在 Linux，`shell` 是極為重要的。如果你使用 Linux 而沒有用 `shell`，你就錯過太多了。

Linux 能高度自訂，可滿足不同的需求和品味。每種變體叫做發行版（*distro*，「distribution」的簡寫）。所有發行版都共用某些核心組件，但是可能長得不同，包含不同的程式與檔案。熱門的發行版有 Mint、Ubuntu、Manjaro、Arch、Gentoo、Red Hat 和 OpenSUSE 等等。本書的核心內容應該能應用到每一種發行版。

啟動 Shell

`shell` 是從哪裡來的？有時候 Linux 會自動為你啟動一個。這時常是在你使用 `ssh` 或類似工具透過網路登入時。你看到第一件重要的事就是等待指令的 `shell` 提示符號。

其他時候，你必須手動啟動 `shell`。這在使用看不到 `shell`、充滿圖示和選單的圖形化桌面時很常見。在這種情況下，你需要在視窗裡執行 `shell`，稱為終端機（*terminal*）或終端機程式（*terminal program*）的 GUI 應用程式。第 4 頁的「Shell 與終端機」側欄會釐清 `shell` 和終端機的差異。

每一個有圖形化桌面的發行版至少會有一個終端機程式，但是你可能要找一下。搜尋名為終端機（Terminal）、Konsole、`xterm`、`gnome-terminal`、`uxterm` 或其他類似的東西，然後執行它以開啟一個終端機。也可以試著按下 `Ctrl-Alt-t`（按住 `Ctrl` 和 `Alt` 鍵，再按下 `T`），在某些環境下會開啟終端機。

Shell 與終端機

shell 是輸入純文字以啟動 Linux 指令的命令列介面。它會印出提示符號等待你的指令：

→

終端機是一個會開啟視窗並呈現執行中 shell 的程式，如圖 1-2 所示。它像是 shell 的圖形化包裝。終端機會加上選單、捲軸、複製與貼上和和其他支援 shell 的 GUI 功能。

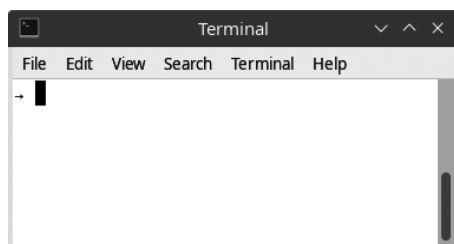


圖 1-2 開啟執行 shell 視窗的終端機程式

命令列暖身

為了讓你對 Linux 有感覺，這裡有十個簡單指令現在就在 shell 試看看。請精確地輸入它們，包括大小寫字母、空白和提示符號後所有的符號。在每個指令結尾按下 Enter。²

顯示 2023 年 11 月的月曆：

```
→ cal nov 2023
November 2023
Su Mo Tu We Th Fr Sa
    1  2  3  4
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30
```

2 如果你看到錯誤訊息「command not found (未找到指令)」，請別擔心：系統可能沒有安裝該指令。請見第 160 頁的「安裝軟體套件」。

檔案系統維護

使用磁碟與檔案系統

<code>df</code>	顯示已掛載的檔案系統上的可用空間。
<code>lsblk</code>	列出磁碟和其他區塊裝置。
<code>mount</code>	使磁碟分割區可存取。
<code>umount</code>	卸載磁碟分割區（讓它不可存取）。
<code>fck</code>	檢查磁碟分割區是否有錯誤。

Linux 系統可以有多個磁碟或分割區。在平常隨意的談話中，這些都被稱為裝置、檔案系統、磁區，甚至目錄。我會試著表達更精確一點。

磁碟（*disk*）是一種大容量儲存裝置，可以分成作為獨立裝置的分割區（*partition*）。磁碟和分割區在 Linux 系統上是 `/dev` 目錄中的特殊檔案。例如，`/dev/sda7` 可能是硬碟上的分割區。`/dev` 中的一些常見裝置有：

<code>sda</code>	第一個區塊裝置（block device），如 SCSI、SATA 或 USB 硬碟；分割區則是 <code>sda1</code> 、 <code>sda2</code> 、...
<code>sdb</code>	第二個區塊裝置；分割區則是 <code>sdb1</code> 、 <code>sdb2</code> 、... 同樣適用於 <code>sdc</code> 、 <code>sdd</code> 等等。
<code>md0</code>	第一個 RAID 裝置；分割區為 <code>md0p1</code> 、 <code>md0p2</code> ... 同樣適用於 <code>md1</code> 、 <code>md2</code> 等等。

fsck

`stdin stdout -file --opt --help --version`

`fsck [options] [devices]`

fsck（filesystem check，檔案系統檢查）指令會驗證 Linux 磁碟檔案系統，並根據要求修復其上發現的錯誤。**fsck** 會在開機時自動執行，或可手動執行。一般來說，在檢查裝置之前先卸載它，這樣就不會有其他程式同時在它上面執行：

```
→ sudo umount /dev/sda10
→ sudo fsck -f /dev/sda10
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/home: 172/1281696 files (11.6% non-contiguous), ...
```

當系統正常執行時，不能使用 **fsck** 來修復 **root** 檔案系統。請先在 Linux USB 隨身碟或其他救援媒體上開機，然後執行 **fsck**。

fsck 是 `/sbin` 中一組檔案系統檢查指令的前端程式，其名稱以「**fsck**」開頭。它僅支援特定類型的檔案系統，請以下列指令列出它們：

```
→ ls /sbin/fsck.* | cut -d. -f2 | column
cramfs      ext3        fat          hfsplus      msdos
ext2        ext4        hfs          minix         vfat
```

好用的選項

- A 依序檢查 `/etc/fstab` 中列出的所有磁碟。
- f 即使沒有明顯的錯誤，也要強制執行 **fsck**。
- N 列印將要完成檢查的描述，並在不執行任何檢查的情況下退出。
- r 互動式修復錯誤，在每次修復前先提示。
- a 自動修復錯誤（僅在你真的知道自己在做什麼時使用；如果不清楚，可能會嚴重破壞檔案系統）。

建立與修改檔案系統

mkfs 格式化（建立檔案系統）磁碟分割區。

resize2fs 擴大或縮小磁碟分割區。

e2label 修改磁碟分割區的磁區標籤。

磁碟分割和格式化等磁碟相關操作在命令列上可能很複雜。一般來說，對於任何比格式化單一分割區更複雜的事情，我建議使用圖形化應用程式，如 **gparted**。老實說，它更簡單，錯誤也更少。

儘管如此，我仍然會在命令列上執行一些快速而簡單的操作。一個是使用 **fdisk** 列出 `/dev/sda` 等儲存裝置的分割區：

```
→ sudo fdisk -l /dev/sda
Disk /dev/sda: 20 GiB, 21474836480 bytes, ...
:
Device      Start      End  Sectors  Size Type
/dev/sda1    2048      4095     2048    1M BIOS boot
/dev/sda2    4096   1054719   1050624   513M EFI System
/dev/sda3  1054720 41940991 40886272  19.5G Linux
```

或類似的 **parted**：

```
→ sudo parted /dev/sda -- print
```

另一個是匯出儲存裝置的分割區表以供保管（將它儲存在 USB 隨身碟或其他裝置上，而不是你正在處理的磁碟上！）。

```
→ sudo sfdisk -d /dev/sda > /mnt/thumb/sda.txt
```

稍後，如果你搞砸了有風險的分割區操作，就可以恢復分割區表（但要小心指定正確的磁碟裝置，否則將會覆蓋錯誤的分割區表）：

```
→ sudo sfdisk /dev/device < /mnt/thumb/sda.txt
```

以下指令也是不用圖形化工具對磁碟與檔案系統之相對基本操作。

mkfs

`stdin stdout -file --opt --help --version`

`mke2fs [options] device`

`mkfs.ext3 [options] device`

`mkfs.ext4 [options] device`

`mkntfs [options] device`

`mkfs.ntfs [options] device` ...和許多其他變化...

`mkfs` 指令系列能格式化 Linux 儲存裝置成各種檔案系統。儲存裝置通常是一個分割區，例如 `/dev/sdb1`。

警告

`mkfs` 會抹除儲存裝置。請確定提供的裝置名稱是正確的！

範例：

→ <code>sudo mkfs.ext4 /dev/device</code>	標準 Linux 檔案系統
→ <code>sudo mke2fs /dev/device</code>	標準 Linux 檔案系統
→ <code>sudo mkfs.ntfs /dev/device</code>	微軟 Windows 檔案系統
→ <code>sudo mkntfs /dev/device</code>	微軟 Windows 檔案系統

如您所見，大多數指令名稱都是以「`mkfs`」加上點和檔案系統類型來表示，如 `mkfs.ext4`。它們也可能有將檔案系統類型嵌入在「`mkfs`」中間的另類名稱（連結），例如「`ext`」檔案系統的 `mke2fs`。要列出所有在系統上安裝的此類指令，請執行：

```
→ ls /usr/*bin/mkfs.*
/usr/sbin/mkfs.ext2    /usr/sbin/mkfs.hfs
/usr/sbin/mkfs.ext3    /usr/sbin/mkfs.minix
/usr/sbin/mkfs.ext4    /usr/sbin/mkfs.msdos
/usr/sbin/mkfs.fat     /usr/sbin/mkfs.ntfs
```

好用的選項

- `-n` 試執行模式：不要格式化任何東西。只顯示將要做的事。
- `-L 名稱` 用指定名稱標記格式化的磁區，其長度可達 16 個位元組。
- `-b N` 將區塊大小設定為 `N` 位元組。

聲音與影片

<code>mediainfo</code>	列印多媒體檔案的詳細資訊。
<code>cdparanoia</code>	將聲音從 CD 擷取到 WAV 檔。
<code>lame</code>	轉換 WAV 為 MP3。
<code>id3info</code>	檢視 MP3 檔的 ID3 標籤。
<code>id3tag</code>	編輯 MP3 檔的 ID3 標籤。
<code>ogginfo</code>	檢視 OGG 檔的相關資訊。
<code>metaflac</code>	檢視和編輯 FLAC 檔的相關資訊。
<code>sox</code>	轉換不同的聲音檔格式。
<code>mplayer</code>	播放影片或聲音檔。
<code>ffmpeg</code>	轉換不同的影片和 / 或聲音檔格式。

有許多用於播放和編輯聲音與影片的圖形化介面 Linux 程式，但讓我們再次專注於命令列工具。

`mediainfo` `stdin stdout -file --opt --help --version`

`mediainfo [options] files`

`mediainfo` 指令能顯示影片和聲音檔的細節。

```
→ mediainfo clip.mp4
General
Complete name      : clip.mp4
Format             : MPEG-4
Format profile     : Base Media / Version 2
Codec ID           : mp42 (mp42/mp41/isom/avc1)
File size          : 1.11 MiB
Duration           : 23 s 275 ms
Overall bit rate   : 399 kb/s
:
```

`mediainfo` 有選項，但通常預設輸出就夠了。如果需要更多，請加上選項 `-f`（「full，完整」）來列印媒體檔相關的每個詳細細節。

cdparanoia

`stdin stdout -file --opt --help --version`

`cdparanoia [options] span [outfile]`

`cdparanoia` 指令可以從 CD 中讀取（擷取）聲音資料，並將其儲存在 WAV 檔（或其他格式：請參閱 `manpage`）中。常見的用法有：

`cdparanoia N`

擷取第 N 軌至檔案。

`cdparanoia -B`

擷取 CD 所有音軌至獨立檔案。

`cdparanoia -B 2-4`

擷取第 2、第 3 和第 4 軌至獨立檔案。

`cdparanoia 2-4`

擷取第 2、第 3 和第 4 軌至單一個檔案。

如果你存取磁碟時遇到困難，請試著執行 `cdparanoia -Qvs`（意思是「仔細地搜尋 CD-ROM 磁碟」）並尋找解法。

lame

`stdin stdout -file --opt --help --version`

`lame [options] file.wav`

`lame` 指令能將 WAV 聲音檔（例如，`song.wav`）轉換為 MP3 檔：

→ `lame song.wav song2.mp3`

它有超過一百個選項控制位元率（bit rate）、轉換其他格式、新增 ID3 標籤等等。

語言	程式	入門
C, C++	gcc,g++	man gcc https://oreil.ly/-kuj1
Java	javac	https://oreil.ly/-yvQR
.NET	mono	man mono https://oreil.ly/IVenL
Perl	perl	man perl https://oreil.ly/LKgDM
PHP	php	man php https://oreil.ly/JPmIL
Python	python	man python https://oreil.ly/vAunc
Ruby	ruby	https://oreil.ly/ifm2L

結語

雖然我已經介紹了許多 Linux 的指令和功能，但還有很多東西要學習。我希望你能繼續閱讀和探索 Linux 系統的能力。

為了提升你的 Linux 技能，請看看我的後續書籍《高效率 Linux 命令列學習手冊》。它超越了基礎知識，以大量實用的技巧和技術讓你使用 Linux 更快、更有效率。欲瞭解更多資訊，請訪視我的網站（<https://danieljbarrett.com>）。

