第八章 系統更新與擴充

8-1 系統核心簡介

8-1-1 何謂核心？

『系統核心』(System kernel) 是整部系統運作的中心，任何處理動作的運行，都必須透過它的指揮，它到底包含了哪些程式？相信也是大家最感到疑問的地方。讀者可由本書第 0 章所敘述的作業系統功能窺視其中端倪，諸如作業系統的週邊管理、記憶體管理、行程排序管理.....等等，都必須分別將它實現成可執行的程式，並將其組織成為一套完善的『系統程式』(System Program)，並常駐於主記憶體內，使其成為一部系統的『核心』。另一方面，也必須將一些較常用的硬體驅動程式 (如音效卡、網路卡、USB 介面卡等等) 嵌入於核心內，以備隨時被呼叫使用，而且『核心』程式與系統的屬性較有關聯。

Linux 可貴的地方，除了是免費的自由軟體之外，它還具有一套可組態規劃的核心，亦即同一套 Linux 作業系統可隨著系統需求而將它組態成網路伺服器、資料處理系統、多媒體工作站、路由器或網路交換器等等。當它被建構成不同系統環境時，所需的核心程式內容當然會有所不同。譬如，建構成一只路由器，核心程式的重點可能是網路驅動程式與路徑選擇演算法，至於虛擬記憶體或其他週邊設備 (如音效卡) 的驅動程式可能較不重要。

核心程式可區分為下列三大部分：

- ✧ **核心影像檔 (Kernel image file):** 已編譯完成的核心執行程式，大多與系統啟動較有關聯；當系統啟動時，『啟動載入器』(Boot Loader) 會將此影像檔存入記憶體內，並立即啟動執行。

- ✧ **核心模組 (Kernel module):** 系統會將某些特殊功能的程式建立成一套模組；組態系統

時，再選擇哪些模組要參與編譯成為核心的一部分。核心模組程式大多儲存於 /lib/module 目錄下。

✧ **initrd 影像檔：**『初始化記憶體磁碟』(Initial RAM Disk, initrd) 是將記憶體空間做為磁碟檔案系統。我們將某些核心模組儲存於 initrd 目錄上，即是 initrd 影像檔。

核心影像檔的內容大多與系統啟動較有關聯，至於建構系統的功能也大多由各個核心模組所構成，因此一般系統管理者甚少去變動或修正核心影像檔的內容，除非是直接升級核心版本。管理者想要組態一部較合適的核心，多半藉由增減或變更核心模組，譬如，建構一套網路伺服器，可能需要嵌入密碼系統 (Crypto) 較為合適。從另一個角度來看，一部電腦可能是許多不同硬體套件拼裝而成，至於需要拼裝哪些硬體週邊，會隨著電腦的應用環境而有所不同，譬如，一部資料處理主機與路由器兩者之間所拼裝的設備究竟有很大的差異。再說，每一只硬體套件都需要一套驅動程式，並由該週邊製造廠商所提供，如果所安裝的硬體套件需要隨時操作的話 (譬如，SCSI 硬碟控制或加速繪圖卡等等)，我們也希望它的驅動程式能嵌入核心，並隨時保存於主記憶體內。最簡單的做法，就是在啟動時將所需硬體的驅動程式，以核心模組方式，儲存於主記憶體內以備隨時被呼叫使用；又為了方便管理，將各種驅動程式以 initrd 方式儲存 (/proc/modules/ 目錄下)，管理者也可隨時觀察到底有哪些驅動程式被嵌入於核心內，我們則統稱這些核心程式為『initrd 影像檔』。

8-1-2 更新系統核心

當有新技術被發展出來，或更新其他硬體設備時，我們都希望作業系統的功能也能隨之提升，此時便有更新系統核心的必要。基本上，更新核心有下列幾種途徑可循：

✧ **修改核心原始程式，再重新編譯而成：**此方法最為透徹，但並非任何人都可以達成此目的。欲編寫核心程式必須具有相當的軟硬體功力才行，除非自行利用 Linux 系統去建置其他電腦設備，譬如製造路由器、交換器或其他網路設備(已超過本書範圍，不另介紹)；否則大多僅由發行者網站下載最新核心版本來更新而已(或稱核心升級，本書將介紹)。

- **嵌入或移除核心模組**：將已編譯完成的核心模組嵌入系統核心，或移除掉不需要的核心模組。
- **變更核心參數**：此方法為最簡單的更新核心方法，某些核心程式是依照參數來運作，如果改變了參數的內容，某些參數也會影響到整個核心的運作。
- **核心升級**：更新版本核心程式，雖然看起來很直接，但有些舊有的套件可能會因版本不合而無法執行。

以下分別介紹之。

8-2 管理核心模組

8-2-1 查閱核心版本 - uname

每一個 Linux distribution 都會將它所發行的核心編號，核心版本表示方法如下：

```
Major.Minor.Relase[-Custom]
```

主要包含四個項目，說明如下：

- Major：主要版本號碼。
- Minor：次要版本號碼。
- Release：第幾次修正發行號碼。
- Custom：發行廠商自行編號。

基本上，版本更新的次序是，經過幾次修正(Release)之後，再增加次要版本(Minor)；若再經過幾次次要版本更新，或系統有大變動時，才會增加主要版本(Major)。以 2.6.5-1.358 版本為例，則主要版本為 2(Major = 2)、次要版本為 6(Minor = 6)、第五次修正發行(Release = 5)、以及廠商編號為 1.358。

我們可利用 uname 命令查閱核心版本，命令格式如下：

```
# uname [OPTION]…
```

常用參數如下：

- -a ：顯示核心所有訊息，如 Linux linux-1 2.6.5-1.358 #1 Sat May 8 09:04:50 EDT 2004 i686 i686 i386 GNU/Lx。
- -r ：僅顯示核心的版本號碼，如 2.6.5-1.358。
- -s ：僅顯示核心名稱，如 Linux。
- -n ：僅顯示主機名稱，如 linux-1。
- -v ：顯示核心編輯的時間，如 #1 Sat May 8 09:04:50 EDT 2004。
- -p ：顯示核心的 CPU 版本，如 i686。
- -i ：顯示主機平台名稱，如 i386。
- -o ：顯示作業系統名稱，如 GNU/Linux。

操作範例如下：

```
[root@tsnien ~]# uname
Linux
[root@tsnien ~]# uname   -a
Linux tsnien.idv.tw 3.10.0-514.el7.x86_64 #1 SMP Tue Nov 22 16:42:41 UTC 2016
x86_64 x86_64 x86_64 GNU/Linux
[root@tsnien ~]# uname   -r
3.10.0-514.el7.x86_64
```

8-2-2 核心模組 - /lib/modules

當系統被建立之後，它的核心模組將被儲存於 /lib/modules/$(unmae –r)/kernel/ 目錄內，其中 $(uname –r) 為目前核心的版本號碼，操作範例如下：

```
[root@tsnien ~]# uname -r
3.10.0-514.el7.x86_64
[root@tsnien ~]# ls /lib/modules/`uname -r`/kernel
arch   crypto   drivers   fs   kernel   lib   mm   net   sound   virt
```

第一個命令（# uname -r）查閱目前所安裝的核心序號，再觀察該序號的核心模組內容如何（/lib/modules/2.6.5-1.358/kernel/）。系統會將屬性相同的模組歸納於同一目錄下，本書範例有下列目錄：

- arch 目錄：儲存與磁碟系統相關的程式模組。
- crypto 目錄：儲存密碼系統相關的程式模組。
- drivers 目錄：儲存硬體設備的驅動程式模組。
- fs 目錄：儲存與檔案系統相關的程式模組。
- lib 目錄：儲存安裝各個模組所需的庫存函數。
- net 目錄：儲存與網路服務較有關聯的程式模組。
- sound 目錄：儲存與音效較有關聯的程式模組。

我們可以觀察看看 crypto 目錄下有哪些程式模組，只是這些模組都已編譯成可執行檔，執行當中可能會使用到 lib 目錄下的庫存函數，範例如下：

```
[root@tsnien ~]# ls /lib/modules/`uname -r`/kernel/crypto
ansi_cprng.ko              crypto_null.ko      rmd160.ko
anubis.ko                  crypto_user.ko      rmd256.ko
arc4.ko                    cts.ko              rmd320.ko
async_tx                   deflate.ko          salsa20_generic.ko
authencesn.ko              des_generic.ko      seed.ko
....
```

8-2-3 查閱模組資訊 - modinfo

我們可利用 modinfo 命令觀察某一模組的訊息為何，其命令格式如下：

```
modinfo [ -a -d -l -p]   [-F field]   [module ...]
```

操作範例如下：

```
[root@tsnien ~]# modinfo /lib/modules/`uname -r`/kernel/crypto/des_generic.ko
filename:       /lib/modules/3.10.0-514.el7.x86_64/kernel/crypto/des_generic.ko
alias:          crypto-des3_ede-generic
alias:          des3_ede-generic
```

```
alias:          crypto-des3_ede
alias:          des3_ede
alias:          crypto-des-generic
filename:       /lib/modules/2.6.11-1.1369_FC4/kernel/crypto/des.ko
alias:          des3_ede
license:        GPL
description:    DES & Triple DES EDE Cipher Algorithms
vermagic:       2.6.11-1.1369_FC4 686 REGPARM 4KSTACKS gcc-4.0
depends:
srcversion:     EF1EE5113D8719DA384B3EE
....
```

由上述範例可以看出，des.ko 的版權擁有者（GPL），以及相關版本訊息。

8-2-4 查閱已載入模組 - lsmod

並非在 kernel 目錄下模組都已經載入系統，我們可利用 lsmod 命令查閱，目前到底有哪些模組已載入核心內，操作範例如下：

```
[root@tsnien ~]# lsmod
Module                  Size  Used by
xt_CHECKSUM             12549  1
iptable_mangle          12695  1
ipt_MASQUERADE          12678  3
nf_nat_masquerade_ipv4  13412  1 ipt_MASQUERADE
iptable_nat             12875  1
nf_nat_ipv4             14115  1 iptable_nat
nf_nat                  26147  2 nf_nat_ipv4,nf_nat_masquerade_ipv4
....
```

上述範例中，每一行表示一只已載入的模組，其中包含三個欄位，說明如下：

- ✧ Module 欄位：顯示模組名稱。
- ✧ Size 欄位：該模組載入記憶體之後，所佔用空間的大小（Byte 單位）。
- ✧ Used by 欄位：有哪些模組正在使用該模組。

8-2-5 載入模組 – insmod

如果需要再載入其他模組進入核心的話，也可利用 insmod(/sbin/insmod)命令來達成，若是自行載入模組，則必須注意到某些模組之間的關聯性，也就是說，除了載入所需的模組

之外，也必須載入它執行時所需的其他模組。操作範例如下：

```
[root@tsnien ~]# lsmod   | grep des_generic  【查閱 des_generic 模組是否已載入】
[root@tsnien ~]# insmod   /lib/modules/`uname -r`/kernel/crypto/des_generic.ko
                                          【載入 des_generic 模組】
[root@tsnien ~]# lsmod   |grep des_generic【查閱 des_generic 模組是否已載入】
des_generic                21379   0      【 des_generic 模組已成功載入】
```

上述範例，我們首先查閱 des_generic 模組是否已載入，發現沒有載入，則再利用 insmod 載入。

8-2-6 移除模組 – rmmod

除了可自行載入模組之外，也可利用 rmmod (/sbin/rmmod) 命令移除已載入的模組。如果模組有被其他模組呼叫使用時，系統將會拒絕移除，而必須先將原模組移除，才會允許移除被呼叫的模組。操作範例如下：

```
[root@tsnien ~]# lsmod   |grep des_generic  【查閱 des_generic 模組已載入】
des_generic                21379   0      【已載入】
[root@tsnien ~]# rmmod des_generic        【移除 des_generic 模組】
[root@tsnien ~]# lsmod |grep des_generic  【查閱 des_generic 模組是否已載入】
```

值得注意的是，如利用直接輸入 insmod/rmmod 命令載入或移除模組，系統重新開機之後，這些設定將不再存在，而必須重新輸入命令。如希望永久保存著，簡單的方法是將這些設定編寫到 /etc/rc.local 檔案內，往後每次開機，系統都將會自動執行該檔案內的命令。

8-3 變更核心參數

當系統啟動時，系統會依照所選定的 Runlevel，執行某些 RC Script，此時會建立一些相關聯的系統參數並儲存於 /proc 目錄下，而且將屬性相同的參數以同一檔案方式儲存。

8-3-1 系統參數 - /proc/sys/

系統參數大多記錄於 /proc/sys/ 目錄底下，在 /proc/ 目錄下也僅有此目錄下的資料可以修改，其他目錄或檔案大多僅供系統操作，並不允許管理者去變更它。一般 /proc/sys 目錄有下列次目錄：

```
[root@tsnien ~]# ls /proc/sys
abi  crypto  debug  dev  fs  kernel  net  sunrpc  vm
```

每一目錄儲存屬性相同的參數，各目錄屬性如下：

- debug ：儲存系統除錯（Debug）的相關訊息。
- dev ：儲存設備檔案（Device file）運作中的相關訊息。
- fs ：儲存檔案系統（File system）運作中的相關訊息。
- kernel ：儲存核心（Kernel）運作中的相關訊息。
- net ：儲存網路（Network）運作中的相關訊息。
- proc ：儲存行程（Process）運作中的相關訊息。
- sunrpc ：儲存 Sun RPC 子系統運作中的相關訊息。
- vm ：儲存虛擬記憶體（Virtual memory）子系統運作中的相關訊息。

上述目錄或子目錄下的檔案，並不一定會儲存資料，某些目錄或檔案是空閒著，有可能是該子系統還未被啟動，運行中的子系統並沒有可供修改的參數（如 proc）。

8-3-2 核心參數 - /proc/sys/kernel/

我們的重點是如何更新核心參數，其儲存於 /proc/sys/kernel/ 目錄下，範例如下：(本書範例)

```
[root@tsnien ~]# ls /proc/sys/kernel
acct                                panic_on_warn
acpi_video_flags                    perf_cpu_time_max_percent
auto_msgmni                         perf_event_max_sample_rate
```

```
bootloader_type                    perf_event_mlock_kb
bootloader_version                 perf_event_paranoid
cad_pid                            pid_max
....
```

上述中每一檔案儲存一個參數的內容，以本書範例所設定的『網域名稱』(Domain name) 為例，該參數 (domainname) 的內容如下：

```
[root@tsnien ~]# cat /proc/sys/kernel/domainname
(none)
```

如欲變更某一參數內容，只要直接修改其內容即可，大多利用 echo 命令就可達成，操作範例如下：

```
[root@tsnien ~]# echo "tsnien.idv.tw" > /proc/sys/kernel/domainname
[root@tsnien ~]# cat /proc/sys/kernel/domainname
tsnien.idv.tw
[root@tsnien ~]#
```

利用命令直接更新參數的內容，如果重新開機之後，所變更的內容會回復成原來的值，所以還是必須將所欲變更的內容填入 /etc/rc.d/rc.local 檔案內，每次開機才會被執行到)。範例如下：

```
[root@tsnien ~]# cat /etc/rc.d/rc.local
#!/bin/bash
# THIS FILE IS ADDED FOR COMPATIBILITY PURPOSES
#
# It is highly advisable to create own systemd services or udev rules
…
# that this script will be executed during boot.

touch /var/lock/subsys/local

echo "tsnien.idv.tw" > /proc/sys/kernel/domainname
```

8-3-3 參數操作命令 - sysctl

除了直接修改參數檔案內容之外，我們也可利用 sysctl 命令 (/sbin/sysctl) 修改參數內容，變更 hostname 參數的操作範例如下：

```
# sysctl -w kernel.hostname=csu_linux 【變更參數內容，等號前後不可以空格】
kernel.hostname = csu_linux
```

```
# sysctl    kernel.hostname                    【顯示參數內容】
kernel.hostname = csu_linux
# cat /proc/sys/kernel/hostname                【顯示參數內容】
csu_linux
```

8-3-4 核心組態檔案 - /etc/sysctl.d/

系統啟動時，會依據 /usr/lib/sysctl.d/、/run/sysctl.d/ 與 /etc/sysctl.d 等目錄下設定檔設定系統參數。/usr/lid/sysctl.d/ 是軟體開發者設定的參數，可能會被 /etc/sysctl.d/ 規劃檔重新設定，另外 /run/sysctl.d/ 是系統執行當然也可以將 sysctl 命令寫入 /etc/rc.d/rc.local 檔案內，每次重新啟動時會再執行當中，所設定的系統參數，優先權最高。一般系統管理者可以在 /etc/sysctl.d/ 下變更參數內容。

```
[root@tsnien ~]# ls /etc/sysctl.d
99-sysctl.conf
[root@tsnien ~]# cat /etc/sysctl.d/99-sysctl.conf
# sysctl settings are defined through files in
# /usr/lib/sysctl.d/, /run/sysctl.d/, and /etc/sysctl.d/.
#
# Vendors settings live in /usr/lib/sysctl.d/.
# To override a whole file, create a new file with the same in
# /etc/sysctl.d/ and put new settings there. To override
# only specific settings, add a file with a lexically later
# name in /etc/sysctl.d/ and put new settings there.
#
# For more information, see sysctl.conf(5) and sysctl.d(5).
```

每一行表示一筆參數的設定，在每行的最前面是井號（#）表示該行為註解使用。又設定變數等號（=）的前後允許空格。

8-4 核心升級

一般 Linux distribution 多半會週期性的發行核心升級版。新版本發行之後，經過許多使用者測試後，一定會發現許多小蟲（bug）或安全漏洞，此時發行者會重新修正核心內容。另一種情況，當新技術或硬體設備被發展出來時，也會針對這方面的需求修改其核心內容。基本上，如果沒有特殊需求，一般發行者都建議管理者不要一昧地追求新版本而更新，唯有

當安全疑慮時才建議管理者更新核心。以 Fedora core 為範例，更新系統有很多方法，下面列舉兩種較普遍使用的方法：

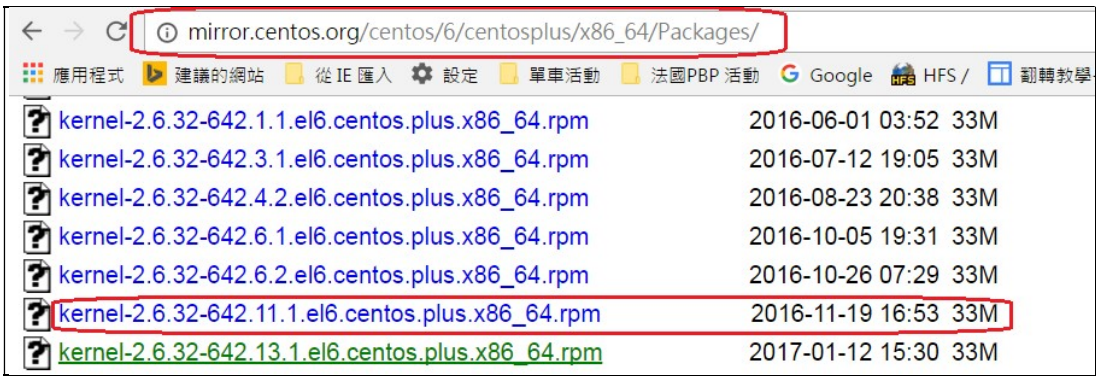
- RPM 套裝軟體管理工具。
- 利用 yum 核心升級工具。

以下分別介紹上述升級工具的操作技巧。

8-4-1 RPM 升級核心

首先須取得的最新核心版本的 RPM 檔案，可透過 Google 搜尋如下：

在鏡射網站上可以找出最新，依目前所找到最新版本是 kernel-2.6.32。接著必須將它下載到 /root 目錄下。(請利用 CentOS 桌面的 firefox 瀏覽，下載到使用者目錄後，再複製到 /root 目錄下)



其實新核心也算是非常危險的動作，萬一處理不當有可能造成整個系統崩潰。比較安全的方法，還是先將舊核心的影像檔備份起來，若新核心運作不正常時，還可利用舊核心挽救回來。備份舊核心程式，操作如下：(/boot 目錄下操作)

```
[root@tsnien ~]# cd    /boot
[root@tsnien boot]# ls | grep vmlinuz
vmlinuz-0-rescue-8649bc024a2b4013b7b7954b99f53a8e
vmlinuz-3.10.0-514.el7.x86_64
[root@tsnien ]# cp vmlinuz-3.10.0-514.el7.x86_64
vmlinuz-3.10.0-514.el7.x86_64_OLD
[root@tsnien boot]# ls | grep vmlinuz
vmlinuz-0-rescue-8649bc024a2b4013b7b7954b99f53a8e
vmlinuz-3.10.0-514.el7.x86_64
vmlinuz-3.10.0-514.el7.x86_64_OLD
[root@tsnien boot]#
```

✧ 步驟 3：安裝核心檔案，操作如下：(/root 目錄下操作)

```
[root@tsnien ~]# ls
anaconda-ks.cfg          kernel-2.6.32-642.13.1.el6.centos.plus.x86_64.rpm
initial-setup-ks.cfg
[root@tsnien ~]#
[root@tsnien ~]# rpm -ivh    kernel-2.6.32-642.el6.centos.plus.x86_64.rpm
正在準備...                ########################################### [100%]
```

完成後，再將新核心檔案複製到 /boot 目錄下。

8-4-2 yum 升級核心

利用 yum(Yellow-dog Updater Modified)自動更新核心版本，可說是最方便了。執行 yum 工具時，系統會自動到 Fedora Core 官方網站搜尋最新版本，如有還未更新的版本，則會自動下載並安裝，操作範例如下：(執行 # yum –y update kernel)

```
[root@tsnien ~]# yum -y update kernel      【執行 yum 自動更新】
Loaded plugins: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
 * base: ftp.tc.edu.tw
 * extras: ftp.tc.edu.tw
 * updates: ftp.tc.edu.tw
Resolving Dependencies
--> Running transaction check
---> Package kernel.x86_64 0:3.10.0-514.6.1.el7 will be installed
--> Finished Dependency Resolution

Dependencies Resolved
....
Complete!
```

```
[root@tsnien ~]# rpm -q    kernel                    【檢視目前所安裝的版本】
kernel-3.10.0-514.el7.x86_64
kernel-3.10.0-514.6.1.el7.x86_64
[root@tsnien ~]# ls    /boot                         【檢視目前啟動核心】
....
vmlinuz-0-rescue-8649bc024a2b4013b7b7954b99f53a8e
vmlinuz-3.10.0-514.6.1.el7.x86_64                    【此為新安裝的版本】
vmlinuz-3.10.0-514.el7.x86_64
vmlinuz-3.10.0-514.el7.x86_64_OLD
                                        【系統開機時，就有四個核心可供選擇】
```

8-5 軟體擴充

軟體可攜性高是 Unix/Linux 最為可貴的地方。無論哪一套 Unix-like 或 Linux distribution 所發展的軟體，大多可以容易的移植到其他系統上，如此也造就了軟體工業的快速發展。Unix/Linux 軟體可攜性高的原因是，核心程式大多是公開的，呼叫核心程式的介面也有其標準化，因此，利用該核心所發展出來的系統工具可攜性都相當高。另一方面，大部分程式語言的編譯器都可以安裝於 Unix/Linux 系統上，所以任何程式語言所發展的軟體，大多可以重新編譯安裝於另一系統上。譬如，利用一般程式語言（如 C、gcc、java 等等）並採用標準系統介面，所編寫成而成的套裝軟體，大多可以在各種 Unix 或 Linux 系統之間流通，只是這些套裝軟體多半需要重新編譯，這也是 Unix 系統上 Makefile 功能最強的地方。

一只套裝軟體必然包含了許多程式，這些程式的相依性也非常高，如何重新編譯程式可是一件棘手的事。還好我們可將軟體套件的編譯連結步驟編寫於 Makefile 內，而將此 Makefile 隨著所有軟體程式發佈，使用者只要執行該 Makefile 便可以重新編譯該軟體，甚至包含軟體安裝都可將它編寫在裡面，如此一來，其他使用者便可容易擴充其軟體。

雖然 Makefile 可以解決軟體擴充的問題，但任何一套軟體大多包含著許多小程式，即使遺漏其中一個小程式，也會造成軟體安裝失敗。RedHat 為了方便軟體擴充操作，於是發展出 RPM（Red Hat Package Manager）工具，它將每一軟體包裝成一只套件，壓縮整合於同

一檔案內，並發佈於網站上讓使用者下載；使用者下載後，只要利用 rpm 工具便可輕易安裝該軟體套件。雖然利用 rpm 工具可以快速的安裝軟體套件，但管理者還是需要到官方網站搜尋最新版本，Fedora 期望系統能像 Windows 一樣自動下載並安裝最新版本，因此又發展了 yum 工具；只是透過 yum 便能自動搜尋更新的版本，並自動下載安裝。本節將分別介紹 rpm、yum、Makefile 與 DNF 等四種軟體擴充方法。

8-5-1 原始檔案擴充 - makefile

在眾多 Unix/Linux 系統之間流傳的軟體套件，還是以原始檔案最為普遍，可攜性也最高。一般做法是將應用軟體（如 apache）的所有原始檔案以打包（如 tar 工具）並壓縮（如 zip 工具）方式，成為一個軟體套件。此軟體套件可經由網路或光碟片傳輸到另一系統上，該系統再經過解打包與解壓縮工具，恢復原來檔案系統架構，並經指定掛載於某一目錄之下。包裝檔案內必定會包含說明檔（如 README）與安裝檔案（Makefile），使用者閱讀說明檔後，即可依照 Makefile 檔案組態（發行者所製作的）來安裝該軟體套件。本書以 Apache（httpd）軟體套件作為範例，說明如何利用原始軟體擴充系統。首先到 Apache 官方網站（www.apache.org/dist/httpd）下載網頁伺服器的軟體套件，其檔案名稱為 apache_1.3.34.tar.gz，安裝步驟如下：

✧ **步驟 1**：解壓縮並展開軟體套件。利用 tar 工具展開套件後，它會將所展開的檔案系統掛載於套件名稱的目錄下，操作如下：

```
[root@Linux-1 ~]# tar xzvf apache_1.3.34.tar.gz
apache_1.3.34/
apache_1.3.34/ABOUT_APACHE
....
[root@Linux-1 ~]# ls -l
總計 17479
drwxr-xr-x    8 xbook games        1024 10 月  14 07:40 apache_1.3.34
-rw-r--r--    1 root    root     2468056    1 月    8 11:16 apache_1.3.34.tar.gz
.....
```

✧ **步驟 2**：查閱說明檔。由 README 檔案可以看出安裝該套件必須經過 4 個程序，如下說明：

```
$ ./configure --prefix=PREFIX
$ make
$ make install
$ PREFIX/bin/apachectl start
```

其中 PREFIX 是指定安裝位置，一般都安裝於 /usr/local/apache 目錄下

✧ **步驟 3**：設定安裝組態，操作如下：

```
[root@Linux-1 apache_1.3.34]# ./configure --prefix=/usr/local/apache
Configuring for Apache, Version 1.3.34
+ using installation path layout: Apache (config.layout)
Creating Makefile
Creating Configuration.apaci in src
Creating Makefile in src
+ configured for Linux platform
.....
```

✧ **步驟 4**：執行 Makefile 安裝軟體（make 與 make install），操作如下：

```
[root@Linux-1 apache_1.3.34]# make
===> src
make[1]: Entering directory `/root/apache_1.3.34'
make[2]: Entering directory `/root/apache_1.3.34/src'
.....
[root@Linux-1 apache_1.3.34]# make install
make[1]: Entering directory `/root/apache_1.3.34'
===> [mktree: Creating Apache installation tree]
./src/helpers/mkdir.sh /usr/local/apache/bin
mkdir /usr/local/apache
mkdir /usr/local/apache/bin
./src/helpers/mkdir.sh /usr/local/apache/bin
.....
```

✧ **步驟 5**：啟動軟體套件，操作如下：

```
[root@Linux-1 apache_1.3.34]# /usr/local/apache/bin/apachectl start
/usr/local/apache/bin/apachectl start: httpd started
[root@Linux-1 apache_1.3.34]# ps -ef |grep httpd
root      9682     1  0 15:01 ?        00:00:00 /usr/local/apache/bin/httpd
nobody    9683  9682  0 15:01 ?        00:00:00 /usr/local/apache/bin/httpd
.....
```

上述範例是利用 apachectl 啟動後，接著觀察其程序是否執行中；但系統重新啟動後，

將不會繼續執行，因此必須將該命令填寫於 rc.local 檔案內，也可利用 apachectl help 觀察此命令是否還有其他選項。

8-5-2 RPM 套件擴充 - rpm

RPM（Red Hat Package Manager）是由 Red Hat 公司所開發出來的，它是個開放性的程式套件管理系統，也相對有一個 rpm 工具來操作 RPM 套件，目前已有許多公司採用 rpm 來管理軟體套件。基本上，rpm 工具將某一軟體套件的所有相關程式，打包成一個 RPM 套件檔案；其他系統（或主機）取得該 RPM 套件後，只要利用 rpm 工具便可安裝該軟體套件。打包而成 RPM 套件的命名格式如下：(但也並非完全如此定名)

套件名稱 + 版本 + 次版本 + 平台

若套件檔案名稱為 telnet-server-0.17-25.i386.rpm，表示該套件名稱為 telnet-server、版本是 0.17、次版本為 25、平台為 i386。

(A) rpm 命令格式

rpm 命令格式如下：

```
$ rpm {-q|--query} [select-options] [query-options]
$ rpm {-V|--verify} [select-options] [verify-options]
$ rpm --import PUBKEY ...
$ rpm {-K|--checksig} [--nosignature] [--nodigest] PACKAGE_FILE ...
```

rpm 命令具有打包套件的功能，也可用來安裝軟體套件，但前者功能已超過本書範圍，有興趣的讀者，可到 RedHat 官方網站參考相關技術文件。這裡僅介紹如何利用 rpm 工具來安裝或管理套件，但 rpm 命令選項非常多，僅就幾個範例簡單介紹，方便容易了解。

(B) 查詢已安裝套件

顯示系統已安裝哪些套件，命令格式如下：

```
$ rpm –qa
```

操作範例如下：

```
$ rpm -qa
setup-2.5.25-1
……
```

(C) 查詢套件的資訊

查詢某一套已安裝套件的資訊，基本格式如下（選項 -q）：

```
$ rpm –q 套件名稱
```

✧ 查詢套件版本為：

```
$ rpm -q setup
setup-2.5.25-1
```

✧ 查詢套件內有哪些檔案的操作如下：

```
$ rpm -ql setup
/etc/bashrc
……
```

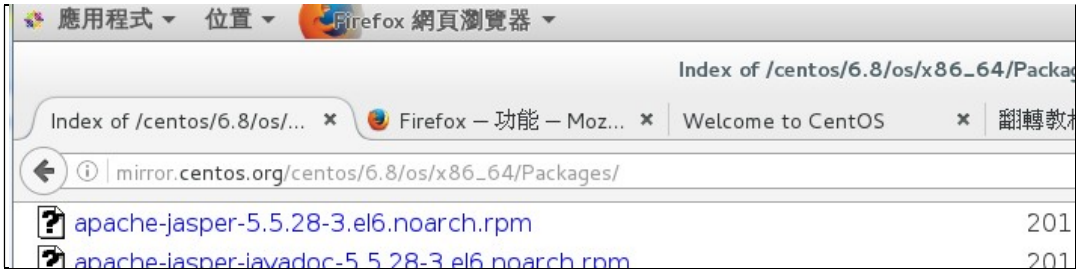
✧ 查詢某一檔案是屬於哪一個套件的操作如下：

```
$ rpm -qf /etc/profile.d
setup-2.5.25-1
```

(D) 安裝套件

由 CentOS/Gru 桌面開啟 firefox，搜尋如下：

下載 apache 套件，如帳號 student01 登入，則下載後，儲存於 /home/student01/下載/目錄下，如下：



將其複製到 /root 目錄下：

```
[root@tsnien ~]# ls /home/student01/下載
apache-tomcat-apis-0.1-1.el6.noarch.rpm
[root@tsnien ~]# cp /home/student01/下載/apach* .
[root@tsnien ~]# ls | grep apach*
apache-tomcat-apis-0.1-1.el6.noarch.rpm
```

另外，必須具有 root 的權限才可以執行安裝 rpm 的命令。命令格式如下：

```
$ rpm –ivh    套件名稱
```

✧ 安裝套件的操作範例如下：

```
[root@tsnien ~]# rpm -ivh apache-tomcat-apis-0.1-1.el6.noarch.rpm
正在準備...
################################# [100%]
Updating / installing...
   1:apache-tomcat-apis-0.1-1.el6
################################# [100%]
```

```
[root@tsnien ~]# rpm –qa  | grep apache
apache-tomcat-apis-0.1-1.el6.noarch
```

✧ 套件已安裝，而要重新再安裝一次的操作方式：

```
# rpm –ivh –replacepkgs apache-tomcat-apis-0.1-1.el6.noarch.rpm
```

(E) 套件升級

已安裝過的套件，升級版本的操作方式如下：

```
# rpm –Uvh apache-tomcat-apis-0.1-1.el6.noarch.rpm
```

(F) 移除套件

移除已安裝套件的操作方式如下：

```
# rpm  –e  apache-tomcat-apis-0.1-1.el6.noarch
```

8-5-3 線上擴充工具 - yum

當軟體套件更新速度很快的時候，隨時尋找最新版本的更新工作，也會造成很大的負擔，yum(Yellow-dog Updater Modified)則為解決此困擾的最佳工具。Yum 具有線上安裝、移除、與更新套件的功能，它能自動連結 RedHat 官方網站，搜尋及比對是否需要更新版本，且會自動判斷套件與系統相容性的問題。Yum 命令格式及選項說明如下：

```
yum [options] [command] [package ...]
```

常用選項（option）有：

✧ -y：內定詢問項，以『yes』回應。

✧ -d [number]：設定偵錯（debug）的等級（0 ~ 10），等級越低則偵測範圍就越小。

✧ -e [number]：設定錯誤等級（0 ~ 10）。

常用命令（command）有：

✧ install package1 [package2]：安裝套件。

- update [package1] [package2] [...]：更新套件。
- check-update：檢查需要更新的套件。
- remove | erase package1 [package2] [...]：移除或刪除套件。
- list [...]：列出套件。
- info [...]：顯示套件訊息。

(A) 設定連結網站

CentOS 將所有 RPM 套件儲存於一個稱為『儲藏庫』(Repository) 的官方網站，並在全世界各地區裝置許多映射 (mirror) 網站；基本上，映射網站會隨時由官方網站下載最新版本，以供區域性客戶需求並下載使用。使用 yum 工具之前，必須將所欲連結的網站位置設定與自己電腦最近的位置 (但也不一定)，才能加快下載速度。有關映射網站由 /etc/yum.repos.d/ 目錄下描述檔所規劃，如下：(**備註：一般 Linux Distribution 會將最新與最近區域的 mirror station 登錄於此目錄下，但各地區的 mirror 會隨時改變，因此所安裝的 Linux 版本不要太舊，否則會連結不上**)

```
[root@tsnien ~]# ls /etc/yum.repos.d
CentOS-Base.repo    CentOS-Debuginfo.repo    CentOS-Media.repo
CentOS-Vault.repo
CentOS-CR.repo       CentOS-fasttrack.repo    CentOS-Sources.repo
```

上述 5 個設定檔皆有效，管理者也可以自行建立 .repo 檔案，我們來觀察其中一個檔案的編寫方式，如下： (有興趣自行慢慢閱讀)

```
[root@tsnien ~]# cat /etc/yum.repos.d/CentOS-Base.repo
# CentOS-Base.repo
#
# The mirror system uses the connecting IP address of the client and the
# update status of each mirror to pick mirrors that are updated to and
# geographically close to the client.    You should use this for CentOS updates
# unless you are manually picking other mirrors.
#
# If the mirrorlist= does not work for you, as a fall back you can try the
# remarked out baseurl= line instead.
```

```
#
#

[base]
name=CentOS-$releasever - Base
mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&repo=os&infra=$infra
#baseurl=http://mirror.centos.org/centos/$releasever/os/$basearch/
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7
....
```

(B) 線上查詢套件

線上查詢可使用 (list avaliable) 的套件較為常用，但顯示出來的訊息可能非常的大，建議儲存於檔案內再查詢，或是透過 grep 工具篩檢較為快速，操作範例如下：

```
[root@Linux-1 yum.repos.d]# yum    list    available
Setting up repositories                              【連結並下載除藏庫資料】
updates-released          100% |=========================|   951 B      00:00
extras                    100% |=========================| 1.1 kB      00:00
base                      100% |=========================| 1.1 kB      00:00
Reading repository metadata in from local files          【除藏庫上可使用的套件】
Available Packages
BibTool.i386                                2.48-3.fc4                  extras
....
```

(C) 詢問可更新套件

可連結到儲藏庫網站查詢有哪些套件可以更新 (原安裝版本較舊)，yum 會下載並比較目前所安裝版本，操作範例如下：(# yum check-update)

```
[root@Linux-1 ~]# yum check-update
Setting up repositories
updates-released          100% |=========================|   951 B      00:00
......
```

(D) 線上安裝套件

無論套件是否已安裝，我們都可以利用 yum 要求重新安裝；yum 會事先上網查詢最新版本，並比較目前版本再來決定應該下載哪些檔案，並詢問是否需要重新安裝。以下範例是利用 yum 工具安裝 php 套件，操作範例如下 (# yum install php)：

```
[root@Linux-1 ~]# yum    -y install    php
Setting up Install Process
Setting up repositories
updates-released              100% |=========================|   951 B       00:00
…
```

(E) 移除套件

移除套件的操作方式如下：(移除 httpd，remove httpd)

```
[root@Linux-1 ~]# yum    remove    httpd
Setting up Remove Process
Resolving Dependencies
--> Populating transaction set with selected packages. Please wait.
---> Package httpd.i386 0:2.0.54-10 set to be erased
….
```

(F) 線上更新套件

線上更新套件應該是最常使用的，如果沒有指定要更新哪一個套件的話，就表示更新系統已安裝的所有套件。更新 setup 套件的操作如下：(# yum update setup)

```
[root@Linux-1 ~]# yum update setup
Setting up Update Process
Setting up repositories
updates-released              100% |=========================|   951 B       00:00
setup-2.5.44-1.1
```

(G) 查詢已安裝套件

查詢系統已安裝的套件，命令如下：(# yum list installed)

```
[root@tsnien ~]# yum list installed | grep setup
cryptsetup.x86_64                        2.0.3-5.el7          @anaconda
....
[root@tsnien student01]# yum list installed | more
Loaded plugins: fastestmirror, langpacks
Installed Packages
GConf2.x86_64                            3.2.6-8.el7          @anaconda
GeoIP.x86_64                             1.5.0-14.el7         @anaconda
....
```

8-5-4 新版線上擴充工具 - DNF

自從 CentOS 8 版本之後，RedHat 公佈 YUM 的下一代更新版本 DNF。但還是可以使

用 YUM，CentOS 7 也可以使用 DNF，但 CentOS 6 之前的版本則沒有提供此功能。基本上兩者都是線上套件管理工具，RedHat 公司表示 DNF 增加了許多功能也較完備， DNF 命令格式及選項說明如下：

(A) 安裝 DNF 套件

安裝 CentOS 8 系統時會自動安裝 YUM 與 DNF 套件，如沒有自行安裝命令如下：

```
[root@tsnien ~]# yum install dnf
```

安裝後，查詢版本如下：

```
[root@secureLab ~]# dnf --version
4.2.7
  已安裝：dnf-0:4.2.7-6.el8.noarch 於 西元 2020 年 03 月 03 日 (週二) 17 時 22 分 04 秒
```

(B) 查詢目前安裝套件

查詢目前系統安裝有哪些套件：

```
[[root@secureLab ~]# dnf   list   installed
```

(C) 查詢目前有哪些套件可安裝

查詢目前有哪些套件可供安裝：

```
[root@secureLab ~]# dnf   list   available
```

(C) 查詢軟體庫是否某套件

查詢某軟體套件是否可供安裝：

```
[root@secureLab ~]# dnf search httpd
上次中介資料過期檢查：1:13:31 以前，時間點為西元 2021 年 05 月 15 日 (週六) 09 時
07 分 33 秒。
========================= Name 精確符合：httpd =========================
httpd.x86_64 : Apache HTTP Server
```

```
==================== Name & Summary 符合： httpd ====================
centos-logos-httpd.noarch : CentOS-related icons and pictures used by
                          : httpd
keycloak-httpd-client-install.noarch : Tools to configure Apache HTTPD
                                     : as Keycloak client
python3-keycloak-httpd-client-install.noarch : Tools to configure
…
```

以上表示該套件存在可供下載安裝。

(D) 套件安裝/重新安裝/移除

- 安裝軟體套件如下：

```
[root@securelab ~]# dnf install httpd
上次中介資料過期檢查：1:20:46 以前，時間點為西元 2021 年 05 月 15 日 (週六) 09 時
依賴關係解析完畢。
.....
已安裝:
  httpd-2.4.37-30.module_el8.3.0+561+97fdbbcc.x86_64
  apr-util-bdb-1.6.1-6.el8.x86_64
  apr-util-openssl-1.6.1-6.el8.x86_64
  apr-1.6.3-11.el8.x86_64
  apr-util-1.6.1-6.el8.x86_64
  httpd-filesystem-2.4.37-30.module_el8.3.0+561+97fdbbcc.noarch
  httpd-tools-2.4.37-30.module_el8.3.0+561+97fdbbcc.x86_64
  mod_http2-1.15.7-2.module_el8.3.0+477+498bb568.x86_64
  centos-logos-httpd-80.5-2.el8.noarch

完成！[確實表示安裝成功]
```

- 重新安裝套件：

```
[root@securelab ~]# dnf -y reinstall httpd
上次中介資料過期檢查：1:28:13 以前，時間點為 ....
.....
已重裝:
```

```
  httpd-2.4.37-30.module_el8.3.0+561+97fdbbcc.x86_64
完成！[確實重新安裝成功]
```

- 移除已安裝套件：

```
[root@securelab ~]# dnf -y remove httpd
模組化的依賴關係問題：
.....
已移除:
  httpd-2.4.37-30.module_el8.3.0+561+97fdbbcc.x86_64
  apr-1.6.3-11.el8.x86_64
  apr-util-1.6.1-6.el8.x86_64
  apr-util-bdb-1.6.1-6.el8.x86_64
  apr-util-openssl-1.6.1-6.el8.x86_64
  centos-logos-httpd-80.5-2.el8.noarch
  httpd-filesystem-2.4.37-30.module_el8.3.0+561+97fdbbcc.noarch
  httpd-tools-2.4.37-30.module_el8.3.0+561+97fdbbcc.x86_64
  mod_http2-1.15.7-2.module_el8.3.0+477+498bb568.x86_64

完成！[確實移除成功]

[root@securelab ~]# dnf   list   installed
```

(E) 其他常用命令匯集

其他常用命令彙集如下：

功　　能	命　令　格　式
安裝套件	# dnf -y install httpd
重新安裝套件	# dnf -y reinstall httpd
移除套件	# dnf -y remove httpd
更新套件	# dnf -y update httpd
更新系統	# dnf update

檢查套件可更新	# dnf check-update
查詢套件可升級	# dnf upgrade
套件升級	# dnf upgrade httpd
自動移除多餘軟體	# dnf autoremove
清除快取空間	# dnf clean all
查詢軟體庫	# dnf repolist all [提供線上安裝之軟體庫]