日誌管理與套件管理

崑山科技大學資訊傳播系
蔡德明
(鳥哥, VBird)

分享指引

- 登錄檔
- 原始碼與Tarball
- Linux distributions提供的安裝/升級機制

登錄檔

系統登錄檔

- 登錄檔所記錄的資訊：
  - 事件發生的日期與時間；
  - 發生此事件的主機名稱；
  - 啓動此事件的服務名稱 (如 samba, xinetd 等) 或函式名稱 (如 libpam ..)；
  - 該訊息資料內容。
- 與登錄檔有關的服務與指令
  - syslogd → 主程式
  - logrotate → 進行登錄檔輪替的指令

系統登錄檔案

- 系統相關登錄檔：
  - /var/log/secure：登錄有『認證』資訊的紀錄
  - /var/log/wtmp：記錄登入者的訊息資料，可用 last 讀取
  - /var/log/messages：預設系統資訊登錄的檔案(非常重要)
  - /var/log/maillog 或 /var/log/mail/*：紀錄郵件存取或往來 (sendmail 與 pop3)的使用者記錄；
  - /var/log/cron：記錄 crontab 這個例行性服務的內容的！
- 其他服務的登錄資訊
  - /var/log/httpd, /var/log/news, /var/log/mysqld.log, /var/log/samba, /var/log/procmail.log

syslogd

- 設定檔 /etc/syslog.conf 語法
  - 服務名稱[.=!]訊息等級　　訊息記錄的檔名或裝置或主機
  - mail.info　　/var/log/maillog_info
- 服務名稱
  - # auth, authpriv：主要與認證有關的機制，例如 telnet, login, ssh 等
  - # cron：就是例行性命令 cron/at 等產生訊息記錄的地方；
  - # daemon：與各個 daemon 有關的訊息；
  - # kern：就是核心 (kernel) 產生訊息的地方；
  - # lpr：亦即是列印相關的訊息啊！
  - # mail：只要與郵件收發有關的訊息紀錄都屬於這個；
  - # news：與新聞群組伺服器有關的東西；
  - # syslog：就是 syslogd 這支程式本身產生的資訊啊！

syslogd(續)

- 訊息等級
  - 1. info ：僅是一些基本的訊息說明而已；
  - 2. notice：比 info 還需要被注意到的一些資訊內容；
  - 3. warning 或 warn：警示的訊息，可能有問題， 但是還不至於影響到某個 daemon 運作的資訊；
  - 4. err 或 error ：一些重大的錯誤訊息， 例如設定檔的某些設定值造成該服務服法啓動的資訊說明！
  - 5. crit：比 error 還要嚴重的錯誤資訊！
  - 6. alert：警告，已經很有問題的等級，比 crit 還要嚴重！
  - 7. emerg 或 panic：疼痛等級，意指系統已經幾乎要當機的狀態！ 很嚴重的錯誤資訊了。

syslog.conf

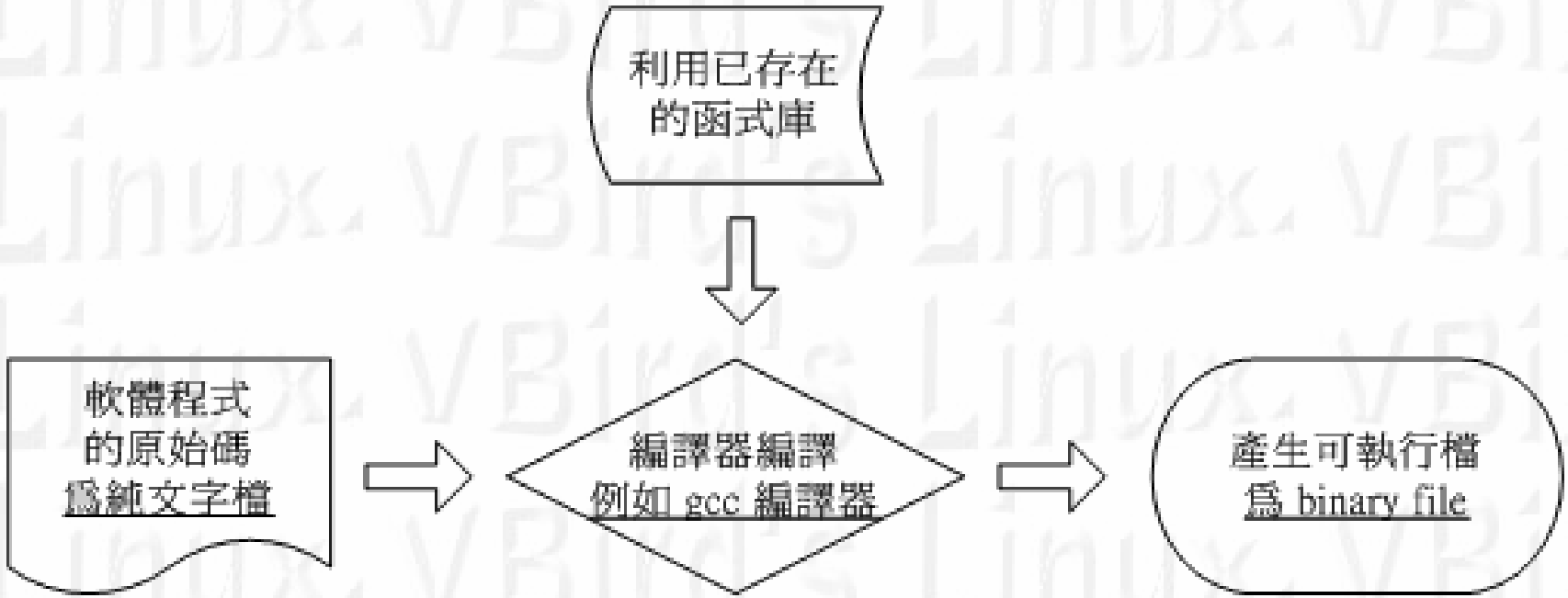
- 系統預設內容
  - *.info;mail.none;authpriv.none;cron.none /var/log/messages
  - authpriv.* /var/log/secure
  - mail.* -/var/log/maillog
  - cron.* /var/log/cron
  - *.emerg *
  - uucp,news.crit /var/log/spooler
  - local7.* /var/log/boot.log
- 非同步化：如上第三行最右邊，加上 – 可增加效能(先記憶在記憶體中，與檔案系統非同步)

syslogd相關

- 登錄資訊所需要的daemons
  - syslogd
  - klogd
- 讓主機變成登錄檔伺服器
  - 伺服器：
    - 啓動 syslogd 時，加入 –r 參數(或修改 /etc/sysconfig/syslog)
    - 啓動的埠口爲 514 (UDP封包)
  - 用戶端：在 /etc/syslog.conf 內，新增一行
    - *.*　　@serverIP

原始碼與Tarball

原始碼與編譯過程



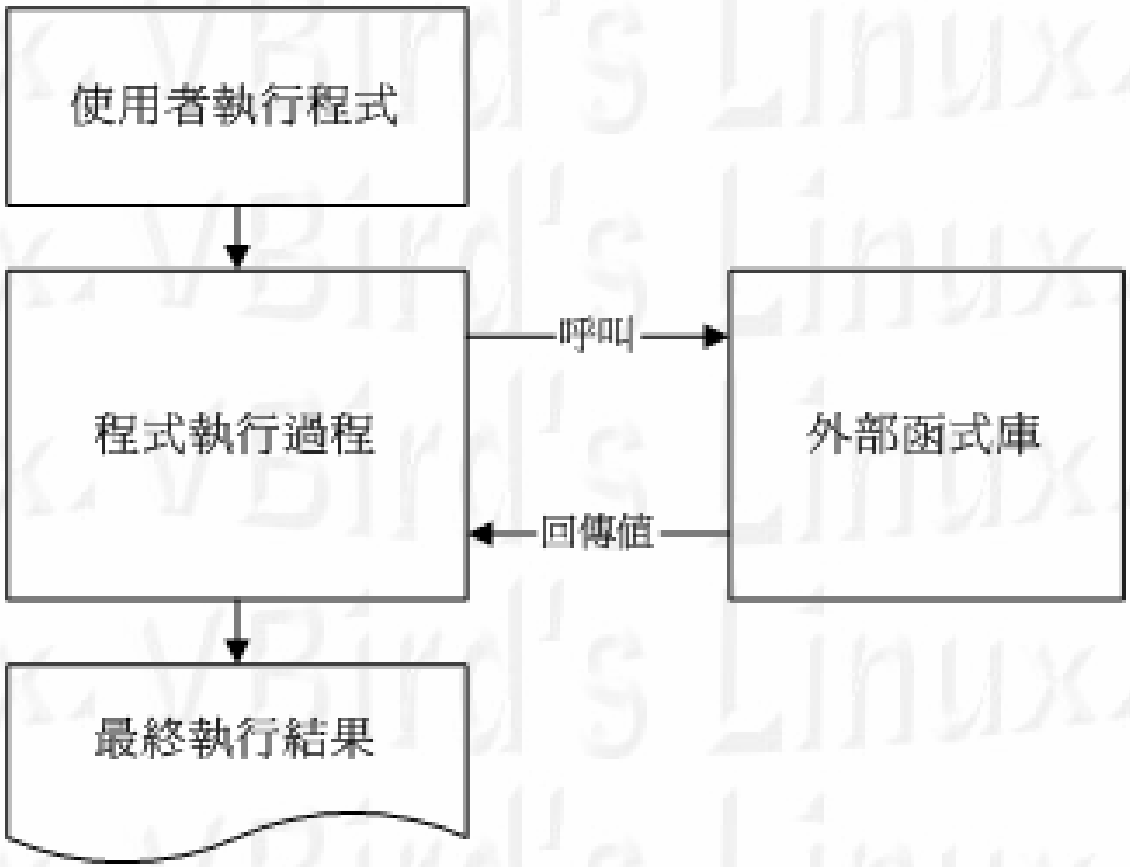
圖一、簡易的 gcc 編譯流程

- file /usr/bin/passwd
- /usr/bin/passwd: setuid ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), for GNU/Linux 2.6.9, dynamically linked (uses shared libs), for GNU/Linux 2.6.9, stripped

函式庫

- 函式庫種類
  - 動態函式庫
    - 程式編譯時，並未包含動態函式庫的程式碼，而是以『指向該函式庫所在的檔名』來呼叫使用
    - 通常附檔名爲libname.so
  - 靜態函式庫
    - 程式編譯時，將此函式庫的程式碼編譯進來，可獨立執行而不需要額外呼叫
    - 通常附檔名爲libname.a
- 函式庫放置的目錄
  - /lib, /usr/lib, /usr/local/lib

動態函式庫的呼叫示意圖



圖二、程式引用函式庫的示意圖

編譯器與編譯

- 編譯器
  - GNU C (gcc), fortran 等等傳統程式語言
- 以gcc編譯的示意
  - gcc –o program file.c
- 如果需要編譯一套軟體，該軟體內有1000個*.c呢？可使用make
  - make 可呼叫工作目錄下的 Makefile 檔案，並據以進行編譯的流程
- 驅動程式編譯尚須的訊息
  - 因爲與kernel相關性高，所以需要kernel-devel套件。

Tarball與安裝

- 原始碼釋出的狀態
  - 由於軟體開發針對不同的作業系統，故以原始碼的形態釋出
  - 爲節省網路頻寬，將原始碼以tar壓縮成*.tar.gz，稱爲tarball
- Tarball的安裝(通常流程)
  - 解壓縮，並查閱新增目錄的INSTALL/README
  - 執行./configure (檢查系統，並建立Makefile)
  - 執行make (開始進行編譯的行爲)
  - 執行 make install(將檔案放置到目錄樹)

Linux distributions提供的安裝/升級機制

Linux distributions

- 利用Tarball的困擾
  - 安裝不容易，而且需要make/autoconfig/kernel-devel等套件
  - 升級不容易，常需要移除再升級
  - 管理不容易，不容易找到所需要的套件資料
- 發佈商的手法：
  - 先在預設的環境下將軟體編譯起來，並打包
  - 利用簡易的指令讓用戶可以直接將軟體放置到正確的目錄樹
  - 建立資料庫，方便使用者查詢到軟體的資訊與檔案
  - 可利用線上升級機制直接 online 安裝/升級/移除等
  - 常見的機制：RPM, DPKG

RPM系統

- RPM：Redhat Package Manager
  - 支援者：RedHat/Fedora/Mandriva/CentOS…等
  - SRPM/RPM
    - xxxxxxxxx.rpm　<==經過編譯且包裝完成的 rpm 檔案
    - xxxxx.src.rpm　<==包含未編譯的原始碼資訊
  - RPM檔名設定
    - rp-pppoe-　3.1　-　5　.i386　.rpm
    - 套件名稱 版本資訊 釋出次數 硬體平台 附檔名
      - i386, i686, x86_64…

RPM的優點

- RPM 檔案本身爲已經編譯過的 binary 檔案，可以讓 client 端的使用者免除重新編譯的困擾；
- RPM 檔案在被安裝之前，RPM 會先檢查系統的硬碟容量、作業系統版本等，可避免檔案被安裝錯誤
- RPM 檔案本身提供套件版本資訊、相依屬性套件名稱、套件用途說明、套件所含檔案等資訊，便於瞭解套件；
- RPM 管理的方式使用資料庫記錄 RPM 檔案的相關參數，便於升級、移除、查詢與驗證。

RPM的使用

- 安裝：
  - rpm –ivh package.rpm →全新安裝
  - rpm –Uvh package.rpm →升級(若未安裝則安裝)
  - rpm –Fvh package.rpm →升級(僅升級有安裝者)
- 查詢：
  - rpm -qa
  - rpm -q[licdR] 已安裝的套件名稱
  - rpm -qf 存在於系統上面的某個檔案
  - rpm -qp[licdR] 未安裝的某個檔案名稱

RPM的使用(續)

- 查詢的細項

```
[root@linux ~]# rpm -qa
[root@linux ~]# rpm -q[licdR] 已安裝的套件名稱
[root@linux ~]# rpm -qf 存在於系統上面的某個檔案
[root@linux ~]# rpm -qp[licdR] 未安裝的某個檔案名稱
參數：
在查詢的部分，所有的參數之前都需要加上 -q 才是所謂的查詢！
查詢主要分為兩部分，一個是查已安裝，另一個則是查某個 rpm 檔案內容。
查詢已安裝套件的資訊：
-q  ：僅查詢，後面接的套件名稱是否有安裝；
-qa ：列出所有的，已經安裝在本機 Linux 系統上面的所有套件名稱；
-qi ：列出該套件的詳細資訊 (information)，包含開發商、版本與說明等；
-ql ：列出該套件所有的檔案與目錄所在完整檔名 (list)；
-qc ：列出該套件的所有設定檔 (找出在 /etc/ 底下的檔名而已)
-qd ：列出該套件的所有說明檔 (找出與 man 有關的檔案而已)
-qR ：列出與該套件有關的相依套件所含的檔案 (Required 的意思)
-qf ：由後面接的檔案名稱，找出該檔案屬於哪一個已安裝的套件；
查詢某個 RPM 檔案內含有的資訊：
-qp[icdlR]：注意 -qp 後面接的所有參數以上面的說明一致。但用途僅在於找出
            某個 RPM 檔案內的資訊，而非已安裝的套件資訊！注意！
```

RPM的使用(續)

- 驗證：
  - rpm -Va
  - rpm -V 已安裝的套件名稱
  - rpm -Vp 某個 RPM 檔案的檔名
  - rpm -Vf 在系統上面的某個檔案
- 反安裝與重建資料庫
  - rpm -e logrotate <==解安裝 logrotate 套件
  - rpm --rebuilddb <==重建資料庫

RPM系統的線上升級

- yum軟體
  - yum server
    - 提供資料庫，讓用戶端可分析軟體相關性
    - 提供RPM檔案，直接讓使用者下載安裝
  - yum 指令
    - 下載資料庫中的軟體相關性
    - 可以直接安裝/移除/升級軟體。
    - 範例：
      - yum update →全系統升級
      - yum install package →僅安裝package
      - yum groupinstall ‘groupname’