
Debianization

Rex Tsai
<chihchun@debian.org.tw>
OSDC. <http://www.osdc.tw/>
國科會 臺北
2006-04-09



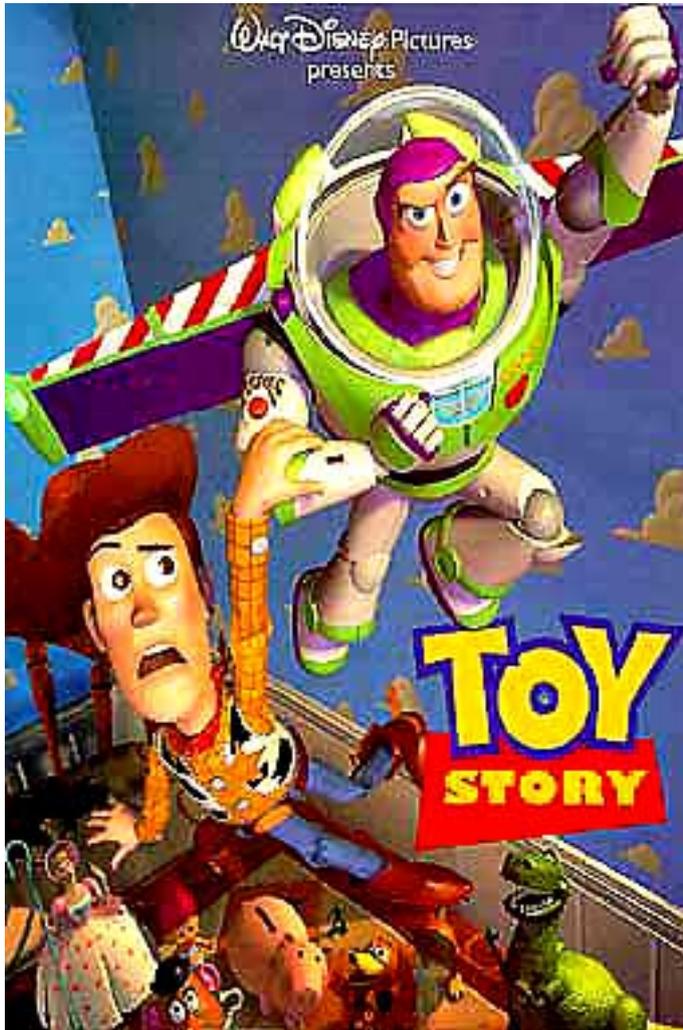
爲什麼 Debian 穩定？

3 個版本分支。

大量的規矩。

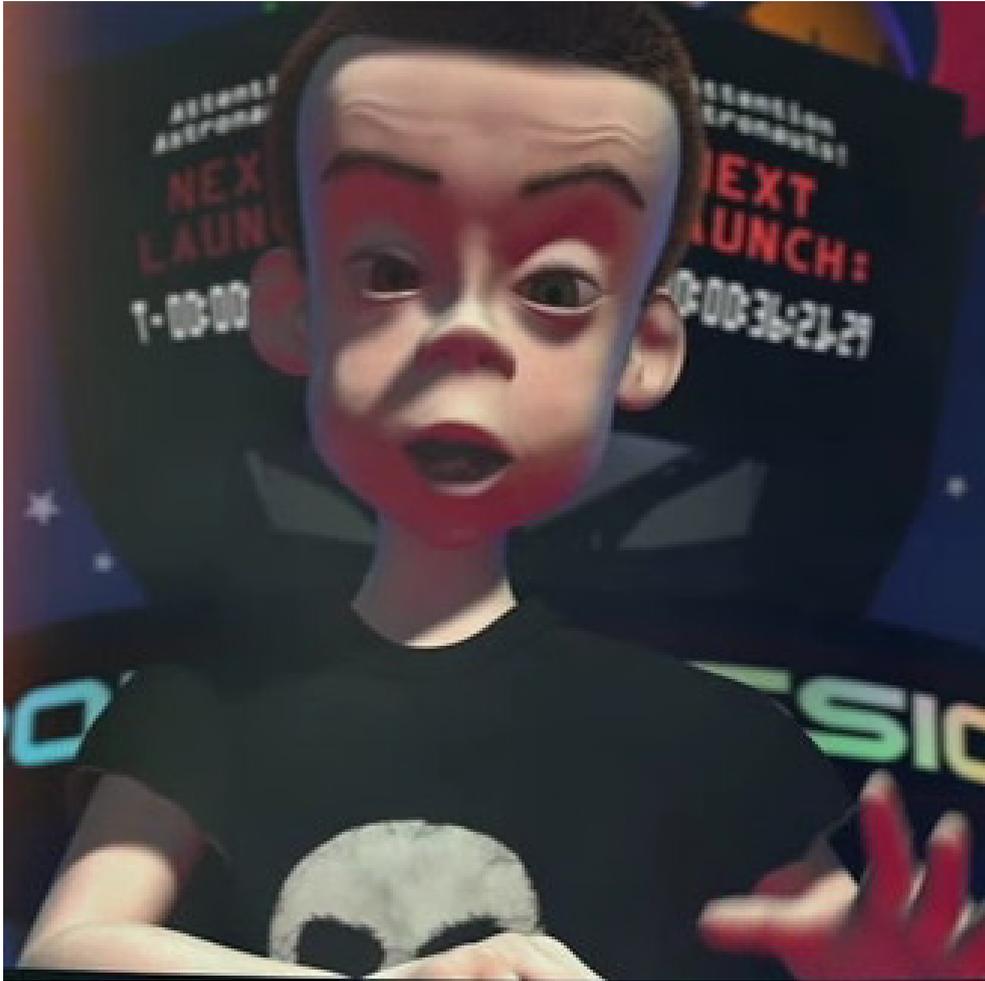
偏執狂開發者。

玩具總動員 (toy story)



名字 (codename) 來自電影玩具總動員 (Toy Story)

`sid' aka unstable

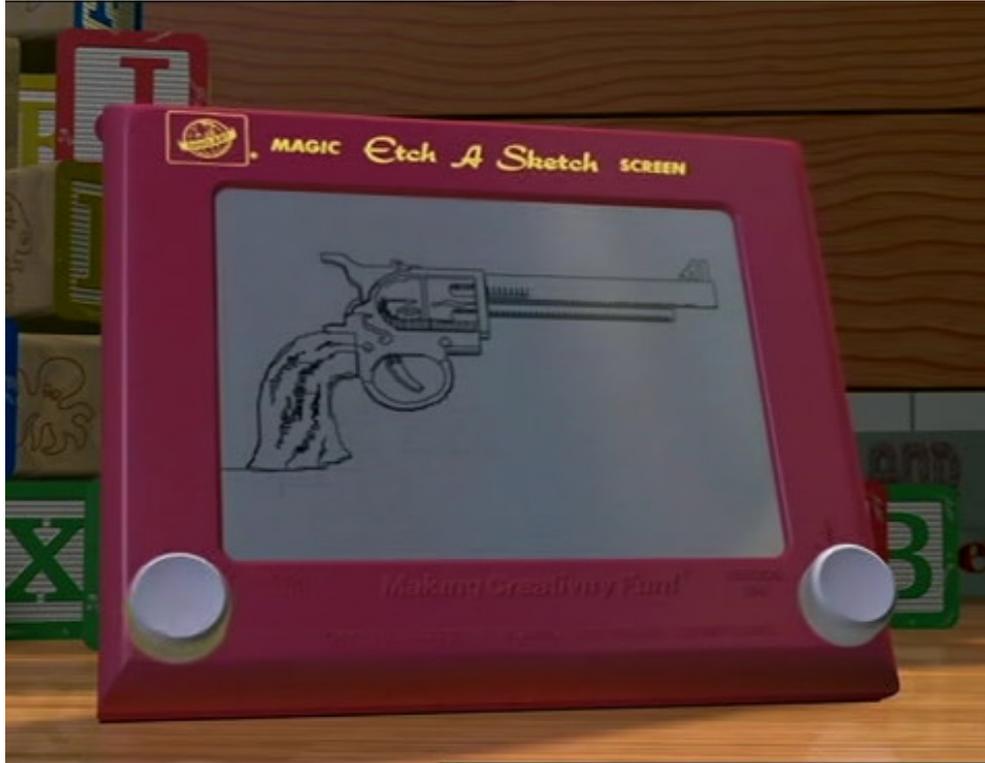


☻ 永遠都是 `sid'

☻ aka still in develop



`Etch' aka testing



- ☉ 一個軟體必須在 `unstable` 中依據上傳的急促性擺上 10/5/2 天，急促性是由開發者所定義，通常依據前版發生缺陷的嚴重程度。
- ☉ 可以在該套件自行指定支援的硬體平台中正確編譯並且在 `sid` 中沒有標注有『釋出關鍵瑕疵』(release-critical bugs)
- ☉ 在 `testing` 中必須有可用的相依套件，或者滿足同時安裝的套件不可因為版本衝突而破壞了 `testing` 中的套件可安裝性



debian

`sarge' aka Debian 3.1



🌀 2005-06-06 釋出

🌀 每個 testing 成為 stable 在釋出前，必須經過一段時間的凍結 (frozen)。

🌀 唯一的改變是安全性的加強，直到下一版穩定版釋出後一年內，都持續會有資訊安全的照料。



experimental



☻ Scud (有人建議)

☻ 這個分支所放的是最新的軟體，不穩定的開發版本。

平台

☉ Intel x86 / IA-32
("i386")

☉ Motorola 68k
("m68k")

☉ Sun SPARC
("sparc")

☉ Alpha ("alpha")

☉ Motorola/IBM
PowerPC
("powerpc")

☉ ARM ("arm")

☉ MIPS CPUs ("mips"
and "mipsel")

☉ HP PA-RISC
("hppa")

☉ IA-64 ("ia64")

☉ S/390 ("s390")

支援但尚未釋出

⊗AMD64

⊗ppc64

⊗SuperH (“sh”)

⊗armeb (big-endian ARM machines)

⊗m32r

套件數

已經超過 15490 個預先編譯好的套件。
35 個分類。

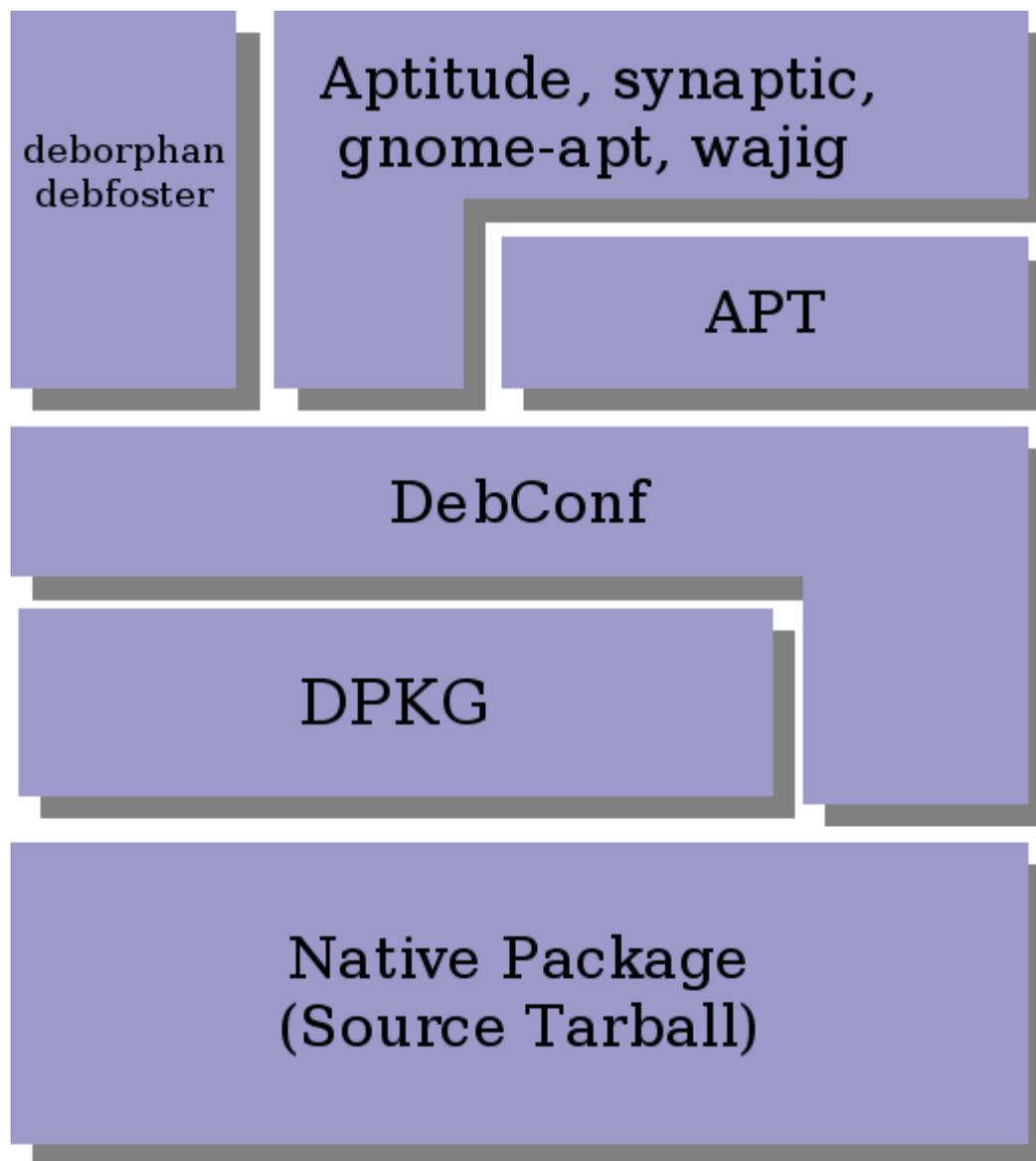
<http://packages.debian.org/unstable/>

與 debtags

<http://debian.vitavonni.de/packagebrowser/>



Debian Packages and Tools



原生軟體 (Native Package)

- ◎ 自由軟體社群有上萬種軟體可以使用，絕大部分都是以前原始碼釋出，而非預先編譯的執行程式，我們將其稱為 **tarball**。
- ◎ 這些 **tarball** 必須經過設定、編譯後，才成為可以使用的執行程式。
- ◎ 雖然部份的自由軟體計畫使用 **GNU-Style** 的 **autoconf/automake** 設定架構，但是仍有許多計畫透過自訂的編譯計畫檔 (**Makefile**) 或命令稿來設定編譯選項，因此使用者必須依據不同的軟體進行不同的編譯指令設定。

DPKG

④ Debian Package Management System

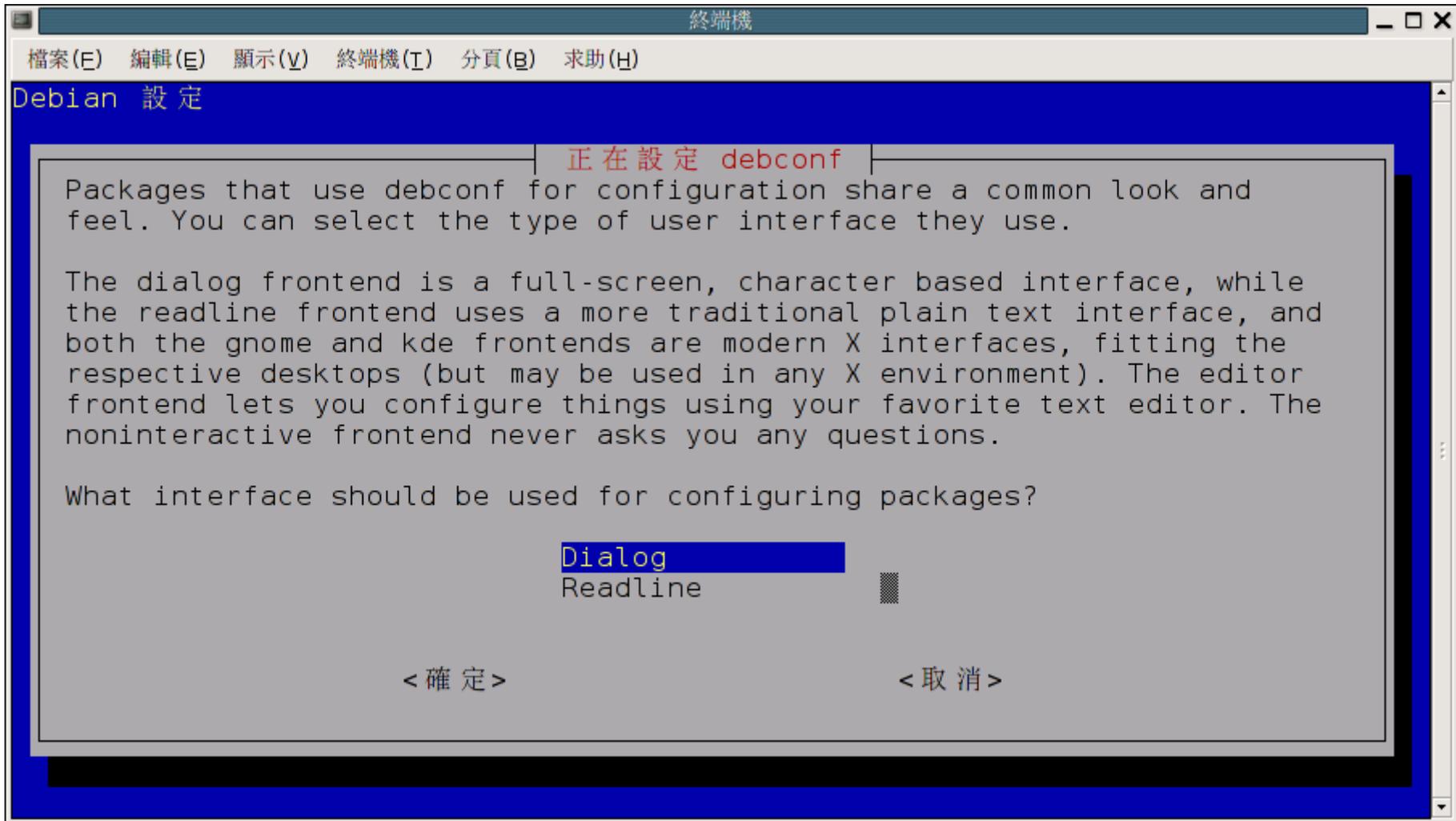
④ 用途就是包裝原始的原始碼，提供設定、及二進位執行程式的安裝系統。

- Binary Packages
- Source Packages

DebConf

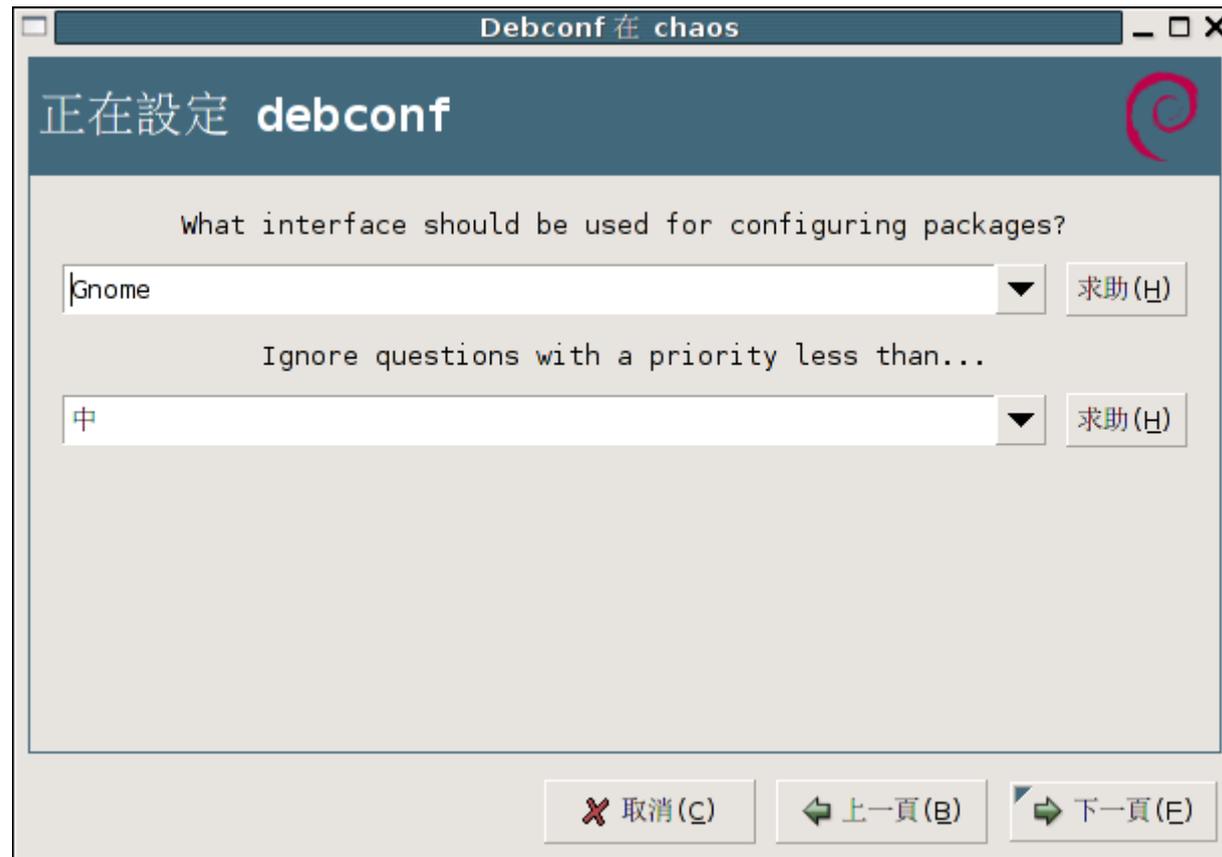
DebConf (Debian Package Configuration System) 是與 dpkg 相輔相成的系統，DebConf 提供統一的軟體設定介面，在你安裝之後，通常可以藉由 DebConf 直接設定配置軟體，因此絕大部分的軟體都是隨裝即用，而免再手動逐一編輯設定檔。

DebConf Interfaces



debian

DebConf Interfaces



Advanced Package Tool

apt (Advanced Package Tool) 是爲了取代 dselect 中未成熟的套件存取系統，提供網路存取與單一的網址結構，與妥善處理軟體依賴關係的函式庫。

由於自由軟體研發本身有擴展性特色，因此 apt 逐漸發展爲與 dpkg 有重複性功能，也提供安裝 / 移除功能。爾後，則有更多以 apt 爲基礎的使用者介面出現，dselect 本身也支援了 apt 作爲主要的套件存取基礎系統。

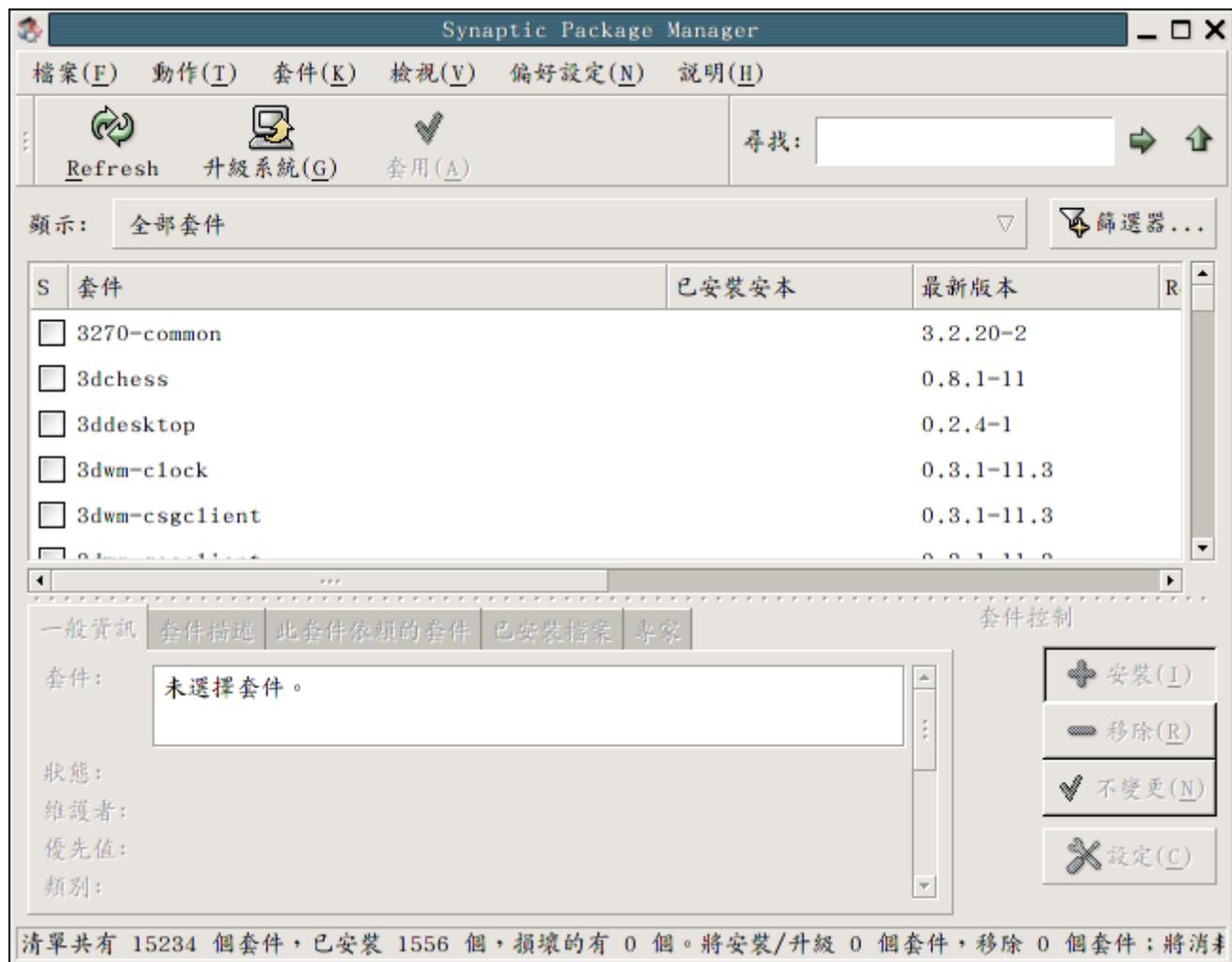


debian

```
Gnome Terminal
動作  復原  選項  查看  求助
f10 :      :退出  u:更新  g:下載/安裝套件
aptitu  復原  C-u
—\ 新
—— admin - 管理工具(安裝軟體, 管理使用者等)
—— base - Debian 基本系統
—— comm - 用於傳真數據機和其它通訊設備的程式
—— devel - 用於軟體開發的工具和程序
—— doc - 文件和查看文件的特殊程序
—— editors - 純文字編輯器和文書排版器
—— electronics - 用於電路和電子器材的程式
—— games - 遊戲, 玩具以及娛樂程式

這些套件在您上次清空“新”套件列表之後添加到 Debian
中。(從動作選單中選擇“忽略新套件”項以清空此列表)

復原最近一次或者一組操作
```



Rules

- ④ Debian Social Contract
- ④ Debian Free Software Guidelines (DFSG)

1. Debian 將始終是 100% 的自由軟體

我們以一份稱為『Debian 自由軟體指導方針』之文件來評斷某項成果是否為「自由」。我們許諾將保證 Debian GNU/Linux 系統與其所有元件將依照此文件之定義為「自由」。我們將支持在 Debian 上運行及開發非自由軟體的使用者，但是我們決不會讓這個系統依賴於任何非自由元件。



2. 我們將回報自由軟體社群

當我們編寫 Debian 系統的新的元件時，我們將其宣告為自由軟體授權。我們將力盡所能，編寫最好的系統，以利自由軟體得到最廣泛的使用及散布。我們將把對錯誤的修正、對程式的改善以及使用者的需求等等，反饋回「上游」、我們系統中所使用的軟體之作者。



3. 我們不會隱藏問題

我們將始終把我們整個的錯誤報告資料庫開放給公眾閱讀。由使用者線上提交的報告，將會立即出現在其他人的眼前。



4. 我們將優先考慮我們的用戶及自由軟體

我們由我們的使用者及自由軟體社群的需要所導向。我們將優先考慮他們的利益。我們將在哪種計算環境中支持我們的使用者的操作需要。我們將不受試圖在 Debian 系統上運行的商業軟體的支配，我們允許他人，在沒有我們的資金的參與下，製造包括 Debian 以及商業軟體的增值套件。為了達成這些目標，我們將提供一集成的、高質量的、100% 自由的軟體，而不附加任何可能阻止在這些方向的使用的法律限制。



5. 不符合我們的自由軟體之方針的程式

我們明瞭，某些我們的使用者需要使用不符合 Debian 自由軟體指導方針的程式。我們為這些軟體，在我們的 FTP 檔案中留出了“contrib”以及“non-free”目錄。在這些目錄下的軟體，並非 Debian 系統的一部分，儘管它們已被配置成可以在 Debian 下使用。我們鼓勵光碟製造商閱讀這些目錄下的軟體的許可證，以判斷是否他們可以在光碟中散布這些軟體。所以，儘管非自由軟體並非 Debian 系統的一部分，我們仍支持它們的使用，我們並為非自由軟體提供了基礎件受（諸如我們的程式錯誤追蹤系統以及通信論壇）。



debian

1 免費自由重複散布

Debian 中的任何元件之許可證不得限制任何一方將此軟體以一個匯集不同來源的程式之整合軟體系統的其中一個元件的方式，來用作銷售或送贈用途。該許可證不一定要求從是項銷售中索取使用費或其他費用。



2. 源碼

軟體需要附有源碼，並允許以源碼及已編譯的形式散布。



3. 衍生作品

許可證必須允許修改及衍生創作，並必須允許修改後的軟體和衍生出的軟體以原來軟體的許可證之同一條款散布。



4. 原作者源碼之完整

許可證 只有 在允許「修補檔案」 (patch files) 與源碼一同分發，以便在編譯時修改程式的情況之下，方可限制源碼不能以修改後的形式散布。許可證必須清楚表明用已修改的源碼編譯而成的軟體，是可以散布的。許可證可要求衍生軟體使用有別於原來軟體的名稱或版本號碼。（這是一個妥協，Debian 組織鼓勵所有作者不要限制任何源碼檔案或機械碼檔案之修改。）



5. 不得歧視任何人士或團體

許可證不得歧視任何人士或由多人組成的團體



6. 不得歧視任何範疇的活動

許可證不得限制任何人在某個範疇的活動中有關程式之使用。例如，許可證不得限制程式作商業或基因研究用途。



7. 許可證之散布

程式附帶的權利必須適用於所有被分發程式的人，並不需要他們履行額外的許可證。



8. 許可證不得特別針對 Debian

程式附帶的權利不得由該程式是否為 Debian 的一部分來決定。如果從 Debian 中取用一個程式，並根據該程式的條款使用或散布它，則被分發程式的所有人之權利與得到 Debian 系統的使用者的權利相同。



9. 許可證的規定不得污染其他軟體

許可證不得對其他與此軟體一同分發的軟體作出任何限制的規定。例如，許可證不得要求所有與它在同一媒體中一同分發的軟體都是自由軟體。



10. 許可證範例

"GPL"、"BSD" 和 "Artistic" 均是我們認為是「自由」的許可證例子。



Are Pine/qmail/ezmlml/djbdns free?

Are Pine/qmail/ezmlm/djbdns free?

© Pine/qmail/ezmlm/djbdns 是自由的嗎?

© 非也

- 因為不允許散佈修改過的二進位檔。



Is Phpnuke Free?

Is Phpnuke Free?

©PhpNuke 是自由的嗎？

©非也

- 作者希望每頁原始碼開頭都必須保留他的重要宣告（ **important notice** ），不得修改。
- 作者希望所有由 **PHPNuke** 處理過的網頁下方，都要加上他的版權宣告。



Is GNU Free Documentation License Free?

Is GFDL free?

© GNU Free Documentation License 是自由的嗎?

© 非也

- Invariant Sections

- 宣告為 Invariant Sections 的章節不得修改。

- Transparent Copies

- 文件原始碼必須以特定公開各式散佈

- Digital Rights Management

- 限制使用者不可以限制的機制散佈這些不限制的自由文件



debian

Is Debian Free?



Is Debian Free?

喂喂！別亂搞呀！！！！

軟體授權很重要

檢驗各種軟體授權是最重要的事情。
不可有模糊不清的條款。

否則請務必與“上游維護者”聯絡
或 `debian-legal` 論壇

Jargon

- ◎ “維護者 (maintainer)” 一字指的是製作套裝軟體的人。
- ◎ “上游作者 (upstream author)” 指的是編寫程式的人。
- ◎ “上游維護者 (upstream maintainer)” 是指非 Debian 計畫的維護程式的人。

軟體授權分級制度

- ◎ **main**: 最基本及主要且符合自由軟體規範的軟體 (packages)。
- ◎ **contrib**: 本身屬於自由軟體但多半卻是相依於非自由軟體。
- ◎ **non-free**: 不屬於自由軟體。
- ◎ **non-us** 這個分類裡頭的軟體牽扯到專利、加密 .. 等等問題，必須放至於美國以外地區。

政策

很重要

關於整個系統的各種檔案位置與設定方法

你必須讀他們！

Policies

⊙ Debian Policy Manual

⊙ Linux Filesystem Hierarchy Standard (LBS)

⊙ Menu System Policy

⊙ MIME Policy

⊙ Emacs policy

⊙ Java policy

⊙ Perl Policy

⊙ Python policy

⊙ Debconf Specification

⊙ Spelling dictionaries and tools policy



undocumented things

細節太多
更多事情甚至沒有寫在文件上

請加入社群

包套件第一件事情

WNPP

看看有沒人動手了，如果沒有就先告訴其他人你要開始幹了。

Jargon - WNPP

@Work-Needing and Prospective Packages

<http://www.debian.org/devel/wnpp/>

@<https://bugs.debian.org> and reportbugs

and Jargons

- ⊙ ITP - 'Intent To Package'
- ⊙ O - 'Orphaned'
- ⊙ RFA - 'Request for Adoption'
- ⊙ RFH - 'Request For Help'
- ⊙ RFP - 'Request For Package'

基本敘述

- ⊙Package name
- ⊙Version
- ⊙Upstream Author
- ⊙URL
- ⊙License
- ⊙Description

工具

- ④ **dpkg-dev** – 這個套裝軟體包裝檔括了在解開、製作、上傳 **Debian** 來源檔案包裝檔時需要用到的工具。
- ④ **dh-make** 和 **debhelper** – 在例子中建立我們的套裝軟體時需要使用 **dh-make** 來建立它的骨架，並且要使用到一些 **debhelper** 套裝軟體中的工具。
- ④ **devscripts** – 這個套裝軟體中包裝檔括了一些非常好的且很有用的腳本程序。

輔助工具

- ◎ **build-essential** - 這個套件包含一些基本的編譯用工具。您不一定需要，但是裝起來比較好。
- ◎ **fakeroot** - 這個工具使您可以模擬變成 **root** 使用者，這在建立套裝軟體的程序的一些部分是必要的。
- ◎ **lintian** - 一個可以幫助您檢查您的套件是否合乎 **Debian** 政策的軟體。

Initial "debianization"

```
@tar zxf hello*.tar.gz
```

```
@dh_make -e chihchun@debian.org.tw -f  
../hello*tar.gz
```

dh_make

```
debianization
檔案(E) 編輯(E) 顯示(V) 終端機(T) 分頁(B) 求助(H)
$ ls
hello-2.1.1.tar.gz
$ tar zxf hello-2.1.1.tar.gz
$ cd hello-2.1.1
$ dh_make -e chihchun@debian.org.tw -f ../hello-2.1.1.tar.gz

Type of package: single binary, multiple binary, library, kernel module or cdfs?
[s/m/l/k/b] s

Maintainer name : Rex Tsai
Email-Address   : chihchun@debian.org.tw
Date            : Sun,  9 Apr 2006 13:26:20 +0800
Package Name    : hello
Version        : 2.1.1
License        : blank
Type of Package : Single
Hit <enter> to confirm:
Done. Please edit the files in the debian/ subdirectory now. hello
uses a configure script, so you probably don't have to edit the Makefiles.
$ █
```



產生出 debian/

```
debianization
檔案(E) 編輯(E) 顯示(V) 終端機(T) 分頁(B) 求助(H)
$ cd debian/
$ ls
changelog      dirs           hello.doc-base.EX  menu.ex        rules
compat        docs           info                postinst.ex    watch.ex
conffiles.ex  emacsen-install.ex  init.d.ex          postrm.ex
control       emacsen-remove.ex  manpage.1.ex       preinst.ex
copyright     emacsen-startup.ex  manpage.sgml.ex    prerm.ex
cron.d.ex     hello-default.ex   manpage.xml.ex     README.Debian
$
```



debian/

🌀 重要的檔案

- “control”
- “changelog”
- “copyright”
- “rules”



debian/control

```
control (~/.workspace/dot/debianization/1/hello-2.1.1/debian) - GVIM
檔案(E) 編輯(E) 工具(T) 語法效果(S) 緩衝區(B) 視窗(W) 輔助說明(H)
[Icons]
1 Source: hello 來源程式包裝檔的名字
2 Section: devel 軟體分類
3 Priority: optional 使用者安裝系統時此套裝軟體的重要程度
4 Maintainer: Rex Tsai <chihchun@debian.org.tw>
5 Build-Depends: debhelper (>= 4.0.0), autotools-dev
6 Standards-Version: 3.6.2
7
8 Package: hello
9 Architecture: any 此軟體適合之硬體平台
10 Depends: ${shlibs:Depends}, ${misc:Depends} 相依性, 見下頁
11 Description: The classic greeting, and a good example
12 The GNU hello program produces a familiar, friendly greeting.
13 allows non-programmers to use a classic computer science tool
14 would otherwise be unavailable to them.
15 .
16 Seriously, though: this is an example of how to do a Debian pa
17 It is the Debian version of the GNU Project's 'hello world' pr
18 (which is itself an example for the GNU Project).
~
~
~
~
:set number 18,2
```



相依性 (Dependency)

⊗ Depends

- 除非把此套裝軟體所倚賴的所有其它套裝軟體安裝好，否則套裝軟體將不會被安裝。

⊗ Recommends

- `dselect` 或是是 `aptitude` 等前端工具在安裝你的套裝軟體的時候，它們會問你是否將與該套裝軟體以推薦的方式相關聯的套裝軟體一起安裝。

⊗ Suggests

- 在一個使用者安裝你的軟體時，所有的前端工具都會詢問他是否要安裝被建議的軟件包裝檔。



相依性 (Dependency)

⊙ Pre-Depends

- 它的要求比 **Depends**: 更强。除非它需要的套裝軟體已經安裝並且正確設定好，它才會被安裝。

⊙ Provides

- 當多個套裝軟體提供同一個功能時，可以定義一些虛擬的套裝軟體名稱。
- `/usr/share/doc/debian-policy/virtual-package-names-list.txt.gz`

衝突性 (conflicting dependency)

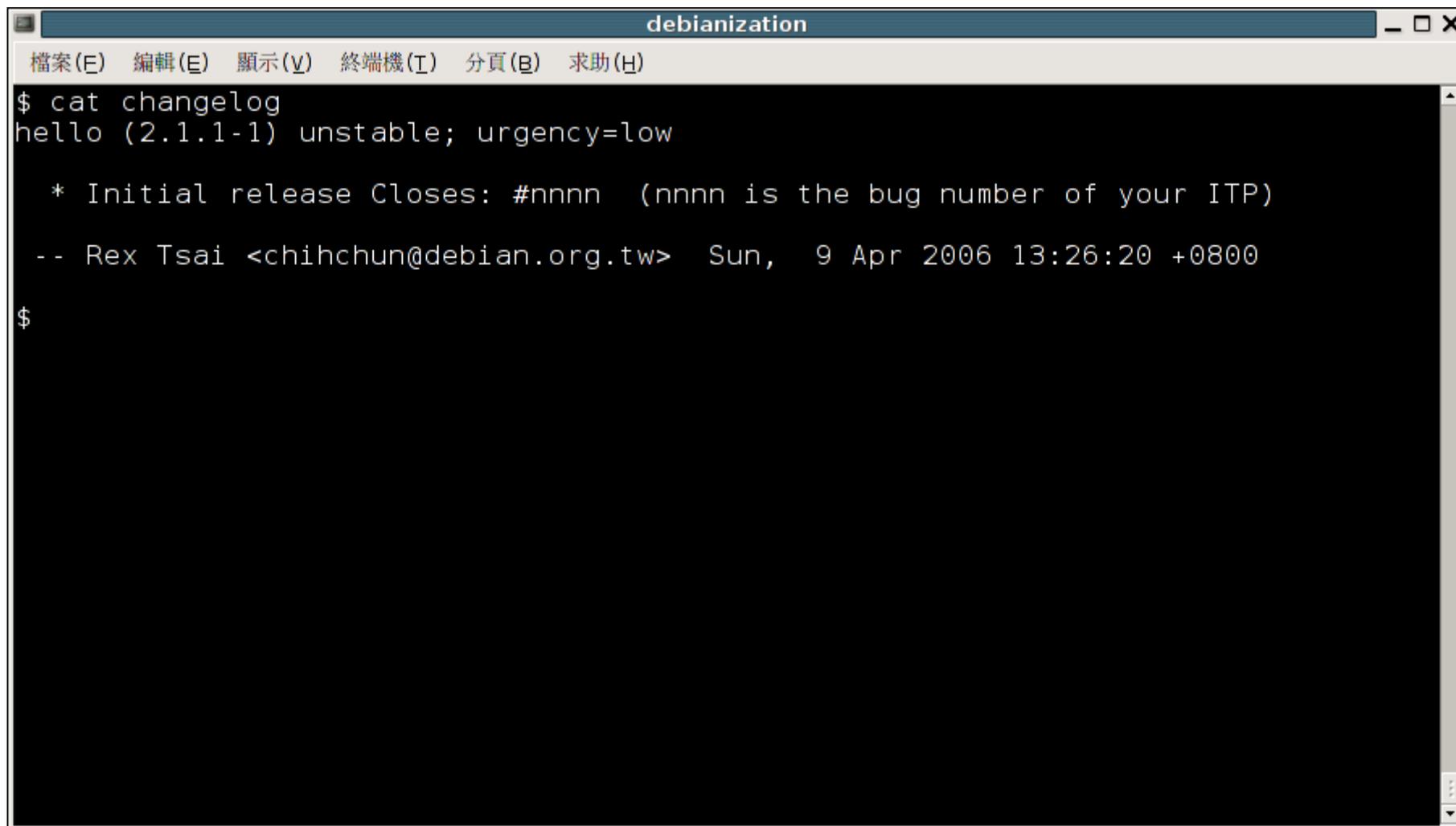
⊗ 衝突 (Conflicts)

- 除非與這個套裝軟體衝突的套裝軟體都已經被刪除了，否則它不會被安裝。

⊗ 取代 (Replaces)

- 當你的套裝軟體會取代一些其它套裝軟體的檔案或是是整個套裝軟體 (與 Conflicts: 聯用) 時，可以使用這個字串。

debian/changelog

A terminal window titled "debianization" with a menu bar containing "檔案(E)", "編輯(E)", "顯示(V)", "終端機(T)", "分頁(B)", and "求助(H)". The terminal shows the command "\$ cat changelog" and its output: "hello (2.1.1-1) unstable; urgency=low", "* Initial release Closes: #nnnn (nnnn is the bug number of your ITP)", "-- Rex Tsai <chihchun@debian.org.tw> Sun, 9 Apr 2006 13:26:20 +0800", and "\$".

```
debianization
檔案(E) 編輯(E) 顯示(V) 終端機(T) 分頁(B) 求助(H)
$ cat changelog
hello (2.1.1-1) unstable; urgency=low

 * Initial release Closes: #nnnn (nnnn is the bug number of your ITP)
-- Rex Tsai <chihchun@debian.org.tw> Sun, 9 Apr 2006 13:26:20 +0800
$
```



Tip: 你可以用 `debchange` 來協助編輯喔。

debian/copyright

```
debianization
檔案(E) 編輯(E) 顯示(V) 終端機(T) 分頁(B) 求助(H)
This is the Debian GNU prepackaged version of the FSF's GNU hello
utility. This package provides the classic greeting to the users of
Debian, as well as providing Debian's developers with an example of
how to package a GNU program. GNU `hello' was written by Mike Haertel,
David MacKenzie, Jan Brittonson, Charles Hannum, Roland McGrath, Noah
Friedman, Karl Eichwalder, and The King.

This package was first put together by Ian Jackson and it's currently
maintained by Santiago Vila. The source for this release was taken
from:

ftp://ftp.gnu.org/gnu/hello/hello-2.1.1.tar.gz

The changes were very minimal:

- Adding support for the Debian package maintenance scheme, by adding
  various debian/* files.

- Modifying hello.c so that /var/mail is searched first.

Program Copyright (C) 1992, 1993, 1997-2001 Free Software Foundation, Inc.
Modifications for Debian Copyright (C) 1994,1995,1996 Ian Jackson.

$ █
```



debian

Main building script : debian/rules

⊙ build

- 設定與編譯

⊙ build-arch (optional), build-indep (optional)

- 若與硬體相關則在此編譯

⊙ binary, binary-arch, binary-indep

- 包成 .deb 吧。
- 以 root 執行。

debian/rules

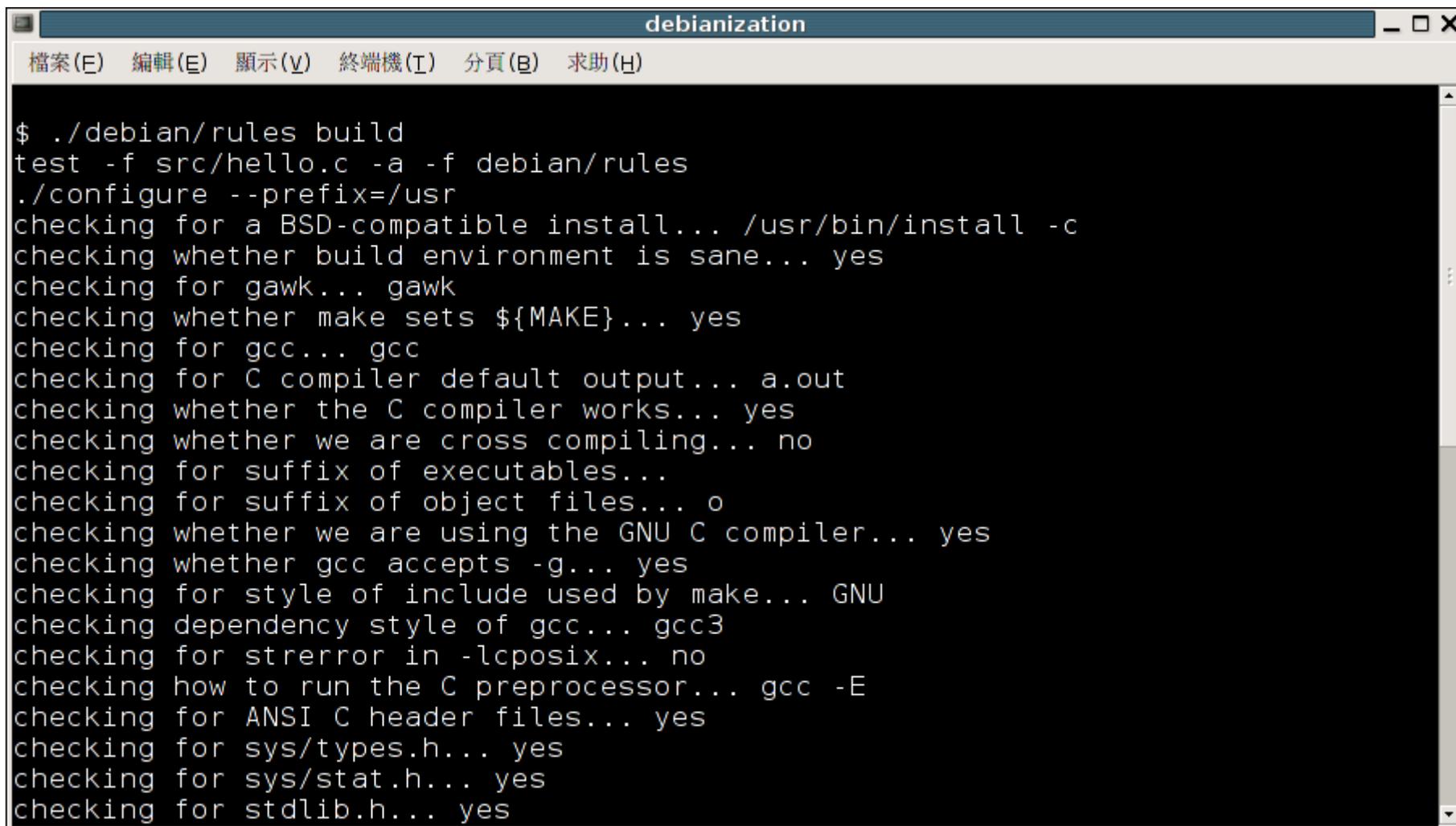
@clean

- 以 `root` 執行。
- 清掉上次的編譯與設定。

@get-orig-source (optional)

- 取得原本的程式碼。(通常沒有人用)

來編譯一下 `./debian/rules build`

A terminal window titled "debianization" with a menu bar containing "檔案(E)", "編輯(E)", "顯示(V)", "終端機(T)", "分頁(B)", and "求助(H)". The terminal displays the output of the command `./debian/rules build`. The output shows a series of checks for various tools and compiler options, all of which pass successfully.

```
$ ./debian/rules build
test -f src/hello.c -a -f debian/rules
./configure --prefix=/usr
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for gawk... gawk
checking whether make sets ${MAKE}... yes
checking for gcc... gcc
checking for C compiler default output... a.out
checking whether the C compiler works... yes
checking whether we are cross compiling... no
checking for suffix of executables...
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for style of include used by make... GNU
checking dependency style of gcc... gcc3
checking for strerror in -lcposix... no
checking how to run the C preprocessor... gcc -E
checking for ANSI C header files... yes
checking for sys/types.h... yes
checking for sys/stat.h... yes
checking for stdlib.h... yes
```

```
rules (~/.workspace/dot/debianization/2/hello-2.1.1/debian) - GVIM
檔案(E) 編輯(E) 工具(T) 語法效果(S) 緩衝區(B) 視窗(W) 輔助說明(H)
# made here.
binary-arch:    checkroot build
                $(checkdir)
                rm -rf debian/tmp
                install -d debian/tmp/DEBIAN $(docdir)
                install -m 755 debian/postinst debian/prerm debian/tmp/DEBIAN
                $(MAKE) INSTALL_PROGRAM="$(INSTALL_PROGRAM)" \
                    prefix=${PWD}/debian/tmp/usr install
                cd debian/tmp && mv usr/info usr/man usr/share
                cp -a NEWS debian/copyright $(docdir)
                cp -a debian/changelog $(docdir)/changelog.Debian
                cp -a ChangeLog $(docdir)/changelog
                cd $(docdir) && gzip -9 changelog changelog.Debian
                gzip -r9 debian/tmp/usr/share/man
                gzip -9 debian/tmp/usr/share/info/*
                dpkg-shlibdeps debian/tmp/usr/bin/hello
                dpkg-gencontrol -isp
                chown -R root:root debian/tmp
                chmod -R u+w,go=rX debian/tmp
                dpkg --build debian/tmp ..

define checkdir
```

50,2-9

79%

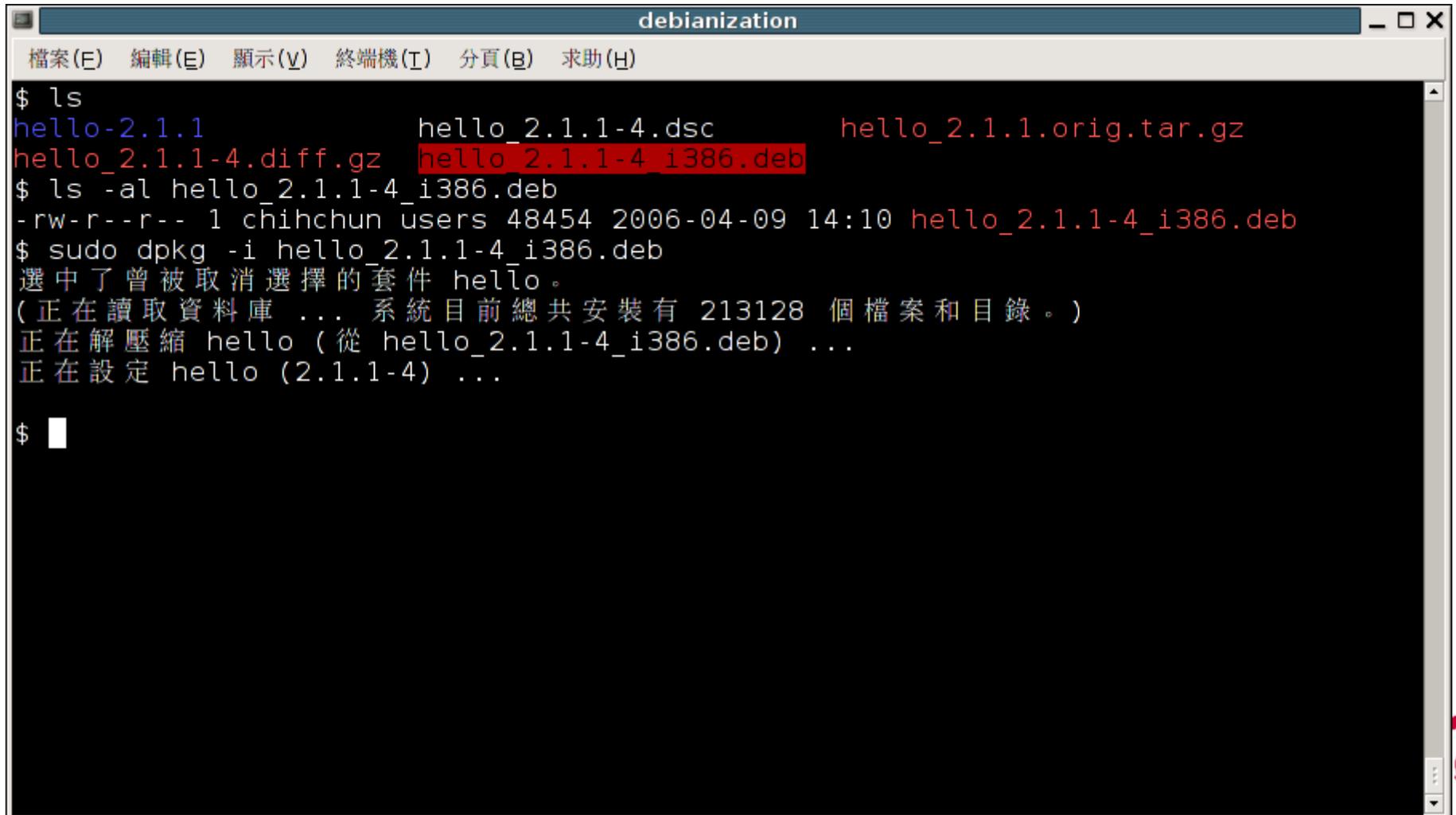


./debian/rules binary-arch

```
debianization
檔案(E) 編輯(E) 顯示(V) 終端機(T) 分頁(B) 求助(H)

$ fakeroot ./debian/rules binary-arch
test -f src/hello.c -a -f debian/rules
test $(id -u) = 0
test -f src/hello.c -a -f debian/rules
rm -rf debian/tmp
install -d debian/tmp/DEBIAN debian/tmp/usr/share/doc/hello
install -m 755 debian/postinst debian/prerm debian/tmp/DEBIAN
/usr/bin/make INSTALL_PROGRAM="install -s" \
            prefix=$(pwd)/debian/tmp/usr install
make[1]: Entering directory `/home/chihchun/workspace/dot/debianization/2/hello-2.1.1'
Making install in contrib
make[2]: Entering directory `/home/chihchun/workspace/dot/debianization/2/hello-2.1.1/contrib'
make[3]: Entering directory `/home/chihchun/workspace/dot/debianization/2/hello-2.1.1/contrib'
make[3]: Nothing to be done for `install-exec-am'.
make[3]: Nothing to be done for `install-data-am'.
make[3]: Leaving directory `/home/chihchun/workspace/dot/debianization/2/hello-2.1.1/contrib'
make[2]: Leaving directory `/home/chihchun/workspace/dot/debianization/2/hello-2.1.1/contrib'
Making install in doc
```

:D



```
debianization
檔案(E) 編輯(E) 顯示(V) 終端機(T) 分頁(B) 求助(H)
$ ls
hello-2.1.1          hello_2.1.1-4.dsc      hello_2.1.1.orig.tar.gz
hello_2.1.1-4.diff.gz hello_2.1.1-4_i386.deb
$ ls -al hello_2.1.1-4_i386.deb
-rw-r--r-- 1 chihchun users 48454 2006-04-09 14:10 hello_2.1.1-4_i386.deb
$ sudo dpkg -i hello_2.1.1-4_i386.deb
選中了曾被取消選擇的套件 hello。
(正在讀取資料庫 ... 系統目前總共安裝有 213128 個檔案和目錄。 )
正在解壓縮 hello (從 hello_2.1.1-4_i386.deb) ...
正在設定 hello (2.1.1-4) ...
$
```



debian

Tip: 其實你可以用 `dpkg-buildpackage` , 他會幫你跑那堆指令。

helper programs for debian/rules

沒有工具的協助

寫 `debian/rules` 很可怕



debhelper

```
rules (~/.workspace/dot/debianiz...llo-debhelper-2.1.1/debian) - GVIM
檔案(E) 編輯(E) 工具(T) 語法效果(S) 緩衝區(B) 視窗(W) 輔助說明(H)
binary-indep: install
# There are no architecture-independent files to be uploaded
# generated by this package. If there were any they would be
# made here.
binary-arch: install
    dh_testdir -a
    dh_testroot -a
    dh_installdocs -a NEWS
    dh_installchangelogs -a ChangeLog
    dh_strip -a
    dh_compress -a
    dh_fixperms -a
    dh_installdeb -a
    dh_shlibdeps -a
    dh_gencontrol -a
    dh_md5sums -a
    dh_builddeb -a
binary: binary-indep binary-arch
.PHONY: binary binary-arch binary-indep clean checkroot
40,15-22 底端
```



好啦，是比較簡單

但是
dh_* 是甚麼鬼

CDBS

Common build system for Debian packages

- 簡易、可讀、有效率！
- 自動處理 debhelper 與 autotools
- 維護者 (maintainer) 可以專注在真正的封裝問題上
- 延伸性



檢查一下 - lintian

規則該死多
用程式來幫忙檢驗有沒有明顯的錯誤

`lintian -vI`

然後呢？

- ◎找一個保證人幫你檢查然後上傳吧。
- ◎或是傳到 apt.debian.org.tw
 - 你可以用 `dput`, `dupload` 來幫忙上傳

繼續玩？

◎自己建 Repository ？

- dpkg-scanpackage / dpkg-scansource

◎給特定語言用的套件

- dh-make-perl - Create debian packages from perl modules
- dh-make-php - Create debian packages from perl modules

◎Voodoo Programming

- apt-build - frontend to apt to build, optimize and install packages

還要玩？

◎ 自動包客制化核心

- sourcerer-kernel-builder

◎ 版本控制你的套件

- cvs-buildpackage
- cvs-autoreleasedeb
- svn-buildpackage
- tla-buildpackage
- darcs-buildpackage

Questions ?



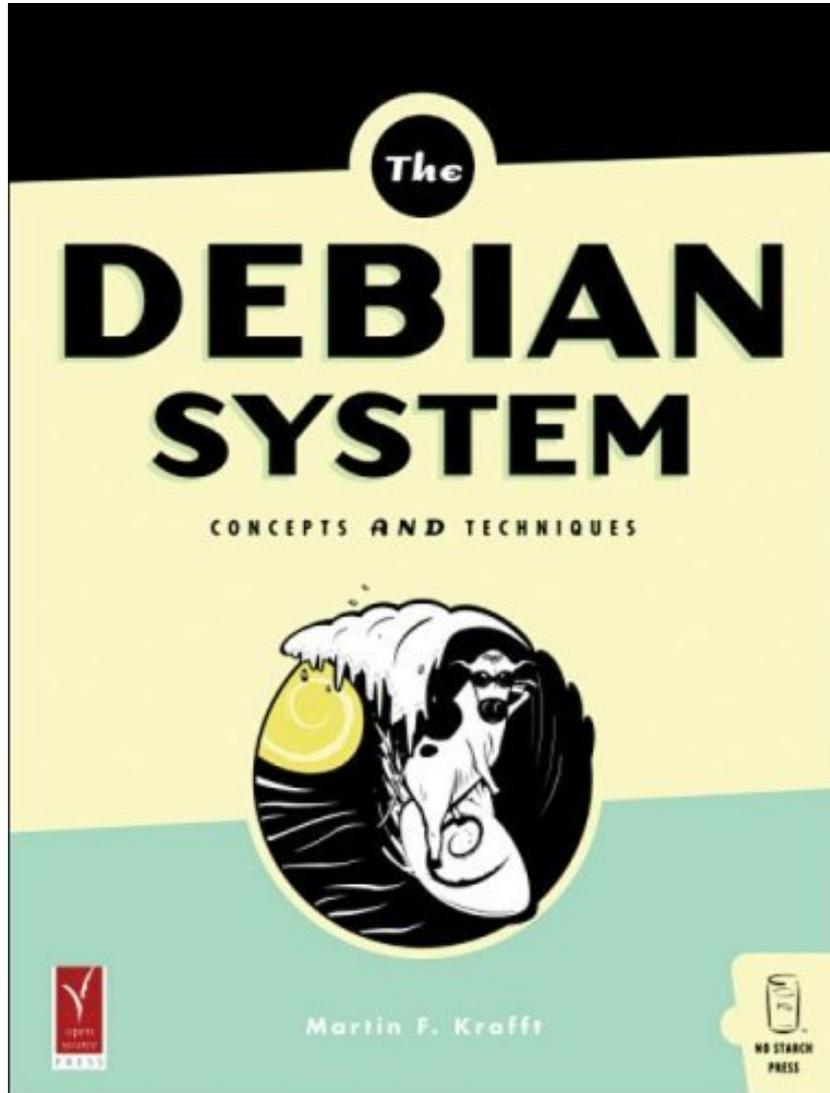
My Blog

Rex's blah blah blah

<http://people.debian.org.tw/~chihchun/>



References – Debian System



© Martin F. Krafft

© 1997 開始的資深
Debian 開發者

© <http://debiansystem.info/>

© <http://blog.madduck.net/>



References

⊙ Debian 套件管理系統

http://www.newzilla.org/2004/12/15/debian_package/

⊙ Debian Developers' Corner <http://www.debian.org/devel/>

⊙ Making a Debian Package

http://people.debian.org/~jaldhar/make_package1.html

⊙ Debian Policy Manual

<http://www.debian.org/doc/debian-policy/>

⊙ Debian Developer's Reference

<http://www.debian.org/doc/developers-reference/>

