

mirror DRBL apt repository

- 以 DRBL for RedHat 9 爲例

參考

→ <http://drbl.nchc.org.tw/redhat/desktop/drbl-cmd-list.html>

執行 `"/opt/drbl/sbin/drbl-create-RH-apt-enabled net 9 yes"`

設定 apache server

apt client 的設定

`/etc/apt/sources.list` 內容要這樣：

```
-----  
rpm http://HOSTNAME/redhat9-apt/ RedHat/apt os updates drbl drbl-test  
-----
```

其中的 HOSTNAME 是您那台主機的名稱。

APT client 設定檔例子

- `/etc/apt/sources.list`

- `rpm http://opensource.nchc.org.tw/redhat/apt
redhat/9/en/i386 os updates drbl drbl-test`
- `rpm-src http://opensource.nchc.org.tw/redhat/apt
redhat/9/en/i386 os updates drbl drbl-test`
- `rpm http://ayo.freshrpms.net redhat/9/i386 os
updates freshrpms`
- `rpm-src http://ayo.freshrpms.net redhat/9/i386 os
updates freshrpms`
- `rpm cdrom:[DRBL]/ RedHat/apt drbl drbl-test
firefly os updates`

APT

- apt-cache search - 搜尋套件
- apt-cache show - 顯示套件說明
- apt-get update - 更新本端套件資訊
- apt-get install - 安裝套件
- apt-get remove - 移除套件
- apt-get upgrade - 更新套件
- apt-get dist-upgrade - 更新整個套件
- apt-cdrom add - 將 apt-enabled CD 加入 list

kernel ? 爲何要重新 make kernel

- Kernel
 - Kernel 在 Unix 系統中，像是你的程式與硬體的一個仲裁者。首先，它爲所有執行中的程式 (程序) 做記憶體管理，並且確保它們都能夠平均 (或不平均，如果你願意) 的分享處理機的運算資源。此外，它還提供了一個良好的介面讓程式能透過它與你的硬體溝通
- make kernel
 - 比較新的核心一般來說
 - (1) 提供更多的設備驅動程式
 - (2) 能夠有更好的行程管理
 - (3) 它們能夠執行的比舊的版本更有效率
 - (4) 它們可能比舊的版本來的更穩定
 - (5) 而且它們修正了舊版裡的一些錯誤
 - 大部分的人是因爲他們需要那些新的設備驅動程式以及錯誤修正的部份而更新核心

何謂模組 (module)

- module
 - 它們是核心的一部分 (通常是設備驅動程式)，但是並沒有編譯到核心裡面去。它們被分別編譯，然後幾乎可以在任何時候將它們插入運作中的核心或從中取出。由於它的便利性，這已經成爲一種增加東西到核心裡去的較好方式。許多常用的設備驅動程式，例如 PCMCIA 驅動程式以及 QIC-0/40 磁帶機的驅動程式就是可載入模組。

手動編譯 kernel

- make mrproper
- make menuconfig (or xconfig)
 - 選選選 ... modules
- make dep
- make bzImage
- make install
- make modules
- make modules_install
- 檢查是否 lilo.conf 符合你的設定！

用 apt 安裝 binary kernel

- apt-get install kernel, apt 會列出可用的 kernel

- 例如：

- apt-get install kernel

Package kernel is a virtual package provided by:

kernel#2.4.20-20.9drbl 2.4.20-20.9drbl

kernel-smp#2.4.20-20.9drbl 2.4.20-20.9drbl

kernel-BOOT#2.4.20-20.9 2.4.20-20.9

kernel#2.4.20-20.9 2.4.20-20.9

kernel-smp#2.4.20-8 2.4.20-8

kernel#2.4.20-8 2.4.20-8

You should explicitly select one to install.

E: Package kernel has no installation candidate

- 確定您要的之後，執行

- apt-get install kernel#2.4.20-9drbl

在 Debian 裡重新 make kernel

- 完全手動法
 - 跟 Slackware、Redhat、Mandrake 及其他所有的 Linux 套件一樣。
 - 到 <http://www.kernel.org/> 抓取你要裝的 kernel 版本
 - 解開、config、compile、後安裝。
- 詳看 kernel source 解開裡面的 README 檔

Debian's Way

- 參考 "Debian 無痛起步 " 裡 #6.3 節
- 你需要的 packages :
 - Kernel source, of course.
 - kernel-source-2.4.18
 - Devel:
 - gcc, libc5-dev/libc6-dev, binutils, make, bin86
 - Interpreters:
 - awk(gawk or mawk)
 - Base:
 - gzip, shellutils, and grep
 - Misc:
 - Kernel-package

Optional, but always needed

- Make menuconfig:
 - ncursesX.X-dev(menuconfig)
- Make xconfig
 - TkX.X-dev(xconfig)

debian.README.gz

- apt-get install [kernel-source-xx.xx.xx](#)
- cd <kernel source tree> (ex: cd /usr/src/)
- unpack kernel-source-xx.xx.xx.tar.bz2
- make config (or make menuconfig)
- make-kpkg clean (you need "kernel-package")
- \$Get_Root make-kpkg --revision=custom.1
kernel_image
- dpkg -i ../kernel-image-X.XXX_1.0_<arch>.deb
- Reboot

一步做完

- `$cd /usr/src/kernel-source-2.4.18`
- `$make-kpkg --revision=mykernel.1 --config=menuconfig kernel_image`
- `$cd ../`
- `$dpkg -i kernel-image-XXXXXX.deb`
- DEMO

Why Debian's Way is Good

- 安裝 kernel-image 的 package 跟你在裝其他的 packages 一樣
 - `dpkg -i kernel-image-XXXX.deb`
- 多台一樣的機器可以在做好 package 之後，直接自動安裝升級 kernel ！
- 升級管理方便 ！

Patch 補丁

- 什麼是 patch?
- 到指定的目錄下：`$patch < xxxx.patch`
- 用 Debian 的方式如何 apply kernel patch?
 - 安裝你要的 kernel patch packages
 - 在前面所述的步驟前先設環境變數：
 - `export PATCH_THE_KERNEL=YES`
 - 在執行指令 `make-kpkg` 中加上 `--added_patches
xxx,000`
- 小心！有些 patch 彼此之間可能會衝突！可能要手動去改程式碼