

The Debian Way

Amministrazione del sistema, strumenti e indicazioni

Maurizio Lemmo

Novembre 2005

The Debian Way

- 1 Le premesse di questo intervento
 - Chi e cosa
- 2 Gli strumenti (1)
 - L'allegra famiglia update-*
- 3 Gli strumenti (2)
 - Interfaces
- 4 Gli strumenti (3)
 - kernel-package
- 5 Gli strumenti (4)
 - java-package

The Debian Way

- 1 Le premesse di questo intervento
 - Chi e cosa
- 2 Gli strumenti (1)
 - L'allegria famiglia update-*
- 3 Gli strumenti (2)
 - Interfaces
- 4 Gli strumenti (3)
 - kernel-package
- 5 Gli strumenti (4)
 - java-package

A chi

Si rivolge questo intervento?

Idealmente, a chiunque abbia installato o utilizzi, da relativamente poco, una Debian/Debian-like.

L'intenzione è quindi, quella di rivolgersi a chi **già** utilizza debian, ma non conosce alcuni strumenti di essa.

Speriamo.

A chi

Si rivolge questo intervento?

Idealmente, a chiunque abbia installato o utilizzi, da relativamente poco, una Debian/Debian-like.

L'intenzione è quindi, quella di rivolgersi a chi **già** utilizza debian, ma non conosce alcuni strumenti di essa.

Speriamo.

A chi

Si rivolge questo intervento?

Idealmente, a chiunque abbia installato o utilizzi, da relativamente poco, una Debian/Debian-like.

L'intenzione è quindi, quella di rivolgersi a chi **già** utilizza debian, ma non conosce alcuni strumenti di essa.

Speriamo.

Cosa e perchè?

Gli strumenti che andremo a presentare non fanno cose particolari, ovvero, non fanno molto di diverso da quello che già avete fatto tante volte in tanti altri modi.

Ma essi sono stati creati con il preciso scopo di automatizzare le parti noiose (è il caso di kernel-package), oppure di "standardizzare" (all'interno dei vostri sistemi) un certo modo di operare.

L'idea quindi non è quella di fare cose nuove. Molti di voi da anni compilano kernel, fanno link simbolici, etc.

Semplicemente, vi si presentano degli strumenti che (si spera) dovrebbero essere più comodi, o utili.

Debian è:

Debian è una distribuzione orientata all'amministratore di sistema.

Cosa e perchè?

Gli strumenti che andremo a presentare non fanno cose particolari, ovvero, non fanno molto di diverso da quello che già avete fatto tante volte in tanti altri modi.

Ma essi sono stati creati con il preciso scopo di automatizzare le parti noiose (è il caso di kernel-package), oppure di "standardizzare" (all'interno dei vostri sistemi) un certo modo di operare.

L'idea quindi non è quella di fare cose nuove. Molti di voi da anni compilano kernel, fanno link simbolici, etc.

Semplicemente, vi si presentano degli strumenti che (si spera) dovrebbero essere più comodi, o utili.

Debian è:

Debian è una distribuzione orientata all'amministratore di sistema.

Cosa e perchè?

Gli strumenti che andremo a presentare non fanno cose particolari, ovvero, non fanno molto di diverso da quello che già avete fatto tante volte in tanti altri modi.

Ma essi sono stati creati con il preciso scopo di automatizzare le parti noiose (è il caso di kernel-package), oppure di "standardizzare" (all'interno dei vostri sistemi) un certo modo di operare.

L'idea quindi non è quella di fare cose nuove. Molti di voi da anni compilano kernel, fanno link simbolici, etc.

Semplicemente, vi si presentano degli strumenti che (si spera) dovrebbero essere più comodi, o utili.

Debian è:

Debian è una distribuzione orientata all'amministratore di sistema.

The Debian Way

- 1 Le premesse di questo intervento
 - Chi e cosa
- 2 **Gli strumenti (1)**
 - L'allegria famiglia update-*
- 3 Gli strumenti (2)
 - Interfaces
- 4 Gli strumenti (3)
 - kernel-package
- 5 Gli strumenti (4)
 - java-package

update-*

Chiamo allegra famiglia update-* una serie di script e programmi, che hanno in comune due aspetti:

- iniziano con il **rassicurante** nome update-
- si occupano di aggiornare (l'avevo detto che era rassicurante) un singolo aspetto del sistema

update-*

Chiamo allegra famiglia update-* una serie di script e programmi, che hanno in comune due aspetti:

- iniziano con il **rassicurante** nome update-
- si occupano di aggiornare (l'avevo detto che era rassicurante) un singolo aspetto del sistema

update-*

Chiamo allegra famiglia update-* una serie di script e programmi, che hanno in comune due aspetti:

- iniziano con il **rassicurante** nome update-
- si occupano di aggiornare (l'avevo detto che era rassicurante) un singolo aspetto del sistema

La lista è lunga

E comprende diversi script. Ci soffermiamo su alcuni, che trovo particolarmente utili:

- update-alternatives
- update-inetd
- update-rc.d
- update-menus

...La lista ne comprende molti altri (alcuni presenti in presenza di alcuni pacchetti), a voi il gusto di scoprire un update-* che vi può servire in taluni compiti.

La lista è lunga

E comprende diversi script. Ci soffermiamo su alcuni, che trovo particolarmente utili:

- update-alternatives
- update-inetd
- update-rc.d
- update-menus

...La lista ne comprende molti altri (alcuni presenti in presenza di alcuni pacchetti), a voi il gusto di scoprire un update-* che vi può servire in taluni compiti.

update-alternatives

Definition

update-alternatives gestisce i link simbolici tra default in presenza di comandi con alternative

Diversi comandi/programmi hanno delle possibili alternative. E' il caso (ad esempio) di vi, x-term, rcp/rlogin, etc. In più debian fornisce dei "meta" comandi come x-window-manager, o x-www-browser per identificare (rispettivamente) il window manager di default o il browser grafico di default.¹

¹Per avere un'idea delle alternatives disponibili date un'occhiata a `/etc/alternatives`

uso di update-alternatives

L'uso più tipico di update-alternatives è:

Example

```
# update-alternatives --list vi
/usr/bin/nvi
/usr/bin/vim
```

uso di update-alternatives (2)

Seguito da:

Example

```
# update-alternatives --config vi
There are 2 alternatives which provide 'vi'.
Selection Alternative
-----
1 /usr/bin/nvi
*+ 2 /usr/bin/vim
Press enter to keep the default[*], or type selection
number:
```

update-inetd

Definition

update-inetd crea, cancella, abilita o disabilita un entry in /etc/inetd.conf

Sebbene editare inetd.conf sia una pratica abbastanza diffusa, questo script ha il pregio di offrire una visione più di alto livello, e come molti update-* ha il pregio di poter essere comodamente incluso in propri script di personalizzazione (maggiormente vero se avete a che fare con molte macchine e procedure che si ripetono nel tempo).

uso di update-inetd

Example

```
# netstat -lnt |grep daytime
tcp 0 0 *:daytime *: LISTEN
udp 0 0 *:daytime *:
# update-inetd --disable daytime
WARNING!!!!!! /etc/inetd.conf contains multiple
entries for
the 'daytime' service. You're about to disable these
entries.
Do you want to continue? [n] y
Ok, I'll continue ...
# netstat -lnt |grep daytime
#
```

update-rc.d

Definition

update-rc.d installa o rimuove i link agli script di avvio System-V (ovvero quelli in /etc/inet.d/)

E' un modo pulito ed efficiente di gestire i link di avvio dei demoni: crea/cancella tutti quelli necessari, con la priorità che decidete (oppure con un default ragionevole)²

²update-rc.d non avvia o ferma il demone.

uso di update-rc.d

Example

```
# update-rc.d -f postfix remove
Removing any system startup links for
/etc/init.d/postfix ...
/etc/rc0.d/K20postfix
/etc/rc1.d/K20postfix
/etc/rc2.d/S20postfix
/etc/rc3.d/S20postfix
/etc/rc4.d/S20postfix
/etc/rc5.d/S20postfix
/etc/rc6.d/K20postfix
```

uso di update-rc.d (2)

Example

```
# update-rc.d postfix defaults
Adding system startup for /etc/init.d/postfix ...
/etc/rc0.d/K20postfix -> ../init.d/postfix
/etc/rc1.d/K20postfix -> ../init.d/postfix
/etc/rc6.d/K20postfix -> ../init.d/postfix
/etc/rc2.d/S20postfix -> ../init.d/postfix
/etc/rc3.d/S20postfix -> ../init.d/postfix
/etc/rc4.d/S20postfix -> ../init.d/postfix
/etc/rc5.d/S20postfix -> ../init.d/postfix
```

update-menus

Definition

update-menus genera il sistema di menu Debian

I menu di norma sono legati al window manager. Questo impone la configurazione ad hoc degli stessi per avere menu aggiornati.

Debian introduce un sistema diverso: insieme al pacchetto "menu", viene generato (tramite appunto update-menus) automaticamente lo stesso e i window manager sono configurati per popolare i propri menu a partire da questo.

uso di update-menus

Non è previsto. Viene invocato, di norma, dal package manager ad ogni installazione/rimozione di pacchetti (per quei pacchetti che hanno delle entry da menu).

The Debian Way

- 1 Le premesse di questo intervento
 - Chi e cosa
- 2 Gli strumenti (1)
 - L'allegria famiglia update-*
- 3 Gli strumenti (2)**
 - Interfaces**
- 4 Gli strumenti (3)
 - kernel-package
- 5 Gli strumenti (4)
 - java-package

interfaces

Definition

`/etc/network/interfaces` è il file di configurazione per tutte le interfacce di rete

È un comodo "posto" dove impostare tutte le configurazioni di rete (che vengono poi date in pasto a `ifup` e `ifdown`), e dove è possibile fare diverse utili configurazioni.

Di prassi il file comprende diverse istanze "iface", "mapping", "auto".

interfaces

Definition

`/etc/network/interfaces` è il file di configurazione per tutte le interfacce di rete

È un comodo "posto" dove impostare tutte le configurazioni di rete (che vengono poi date in pasto a `ifup` e `ifdown`), e dove è possibile fare diverse utili configurazioni.

Di prassi il file comprende diverse istanze "iface", "mapping", "auto".

una prima configurazione

Example

```
auto eth0
iface eth0 inet static
    address 192.168.0.42
    network 192.168.0.0
    netmask 255.255.255.0
    broadcast 192.168.0.255
    gateway 192.168.0.1
```

E' un inizio...

un setup più complicato

Example

```
auto eth0
iface eth0 inet static
    address 192.168.1.42
    network 192.168.1.0
    netmask 255.255.255.128
    broadcast 192.168.1.0
    up route add -net 192.168.1.128 netmask
255.255.255.128 gw 192.168.1.2
    up route add default gw 192.168.1.200
    down route del default gw 192.168.1.200
    down route del -net 192.168.1.128 netmask
255.255.255.128 gw 192.168.1.2
```

anatomia

in sintesi, le istanze in interfaces sono composte da

1

```
auto <IFACE>
```

opzionale, viene avviata al boot del sistema

2

```
iface <IFACE> inet (inet6, o ipx...) <METHOD>
```

dove <METHOD> a seconda che sia static, loopback, dhcp, ppp offre ulteriori opzioni

anatomia

in sintesi, le istanze in interfaces sono composte da

1

```
auto <IFACE>
```

opzionale, viene avviata al boot del sistema

2

```
iface <IFACE> inet (inet6, o ipx...) <METHOD>
```

dove <METHOD> a seconda che sia static, loopback, dhcp, ppp offre ulteriori opzioni

anatomia (2)

Possiamo quindi utilizzare le opzioni pre-up, up, post-up, down, pre-down, post-down per eseguire appositi script o comandi³
Se utilizzate il pacchetto wireless-tools, avete a disposizione anche un wrapper a iwconfig, nella forma

3

```
wireless-essid Home  
wireless-mode Managed  
...
```

³in caso di fallimento di uno script, il comportamento di ritorno sull'interfaccia cambia, a seconda di quale momento viene scelto per l'esecuzione.

un setup ancora più complesso

Example

```
auto eth0 eth1
mapping eth0 eth1
    script /path/to/get-mac-address.sh
    map 11:22:33:44:55:66 lan
    map AA:BB:CC:DD:EE:FF internet
iface lan inet static
    address 192.168.42.1
    netmask 255.255.255.0
    pre-up /usr/local/sbin/enable-masq $IFACE
iface internet inet dhcp
    pre-up /usr/local/sbin/firewall $IFACE
```

anatomia (3)

mapping permette di assegnare e scegliere dei nomi logici a determinate interfacce fisiche.

Un insieme conveniente di script utili è in `/usr/share/doc/ifupdown/examples` (tra cui c'è `get-mac-address.sh`).

anatomia (3)

mapping permette di assegnare e scegliere dei nomi logici a determinate interfacce fisiche.

Un insieme conveniente di script utili è in `/usr/share/doc/ifupdown/examples` (tra cui c'è `get-mac-address.sh`).

The Debian Way

- 1 Le premesse di questo intervento
 - Chi e cosa
- 2 Gli strumenti (1)
 - L'allegria famiglia update-*
- 3 Gli strumenti (2)
 - Interfaces
- 4 Gli strumenti (3)**
 - **kernel-package**
- 5 Gli strumenti (4)
 - java-package

la gente compila il kernel

da un sacco di tempo. La compilazione del kernel è gioia e dolore di tutti gli utenti linux. kernel-package è un sistema comodo per aiutare questo processo, e fornisce diversi indubbi vantaggi tra cui:

- conveniente: non rischiate di dimenticare qualche passaggio
- permette di avere diversi kernel sulla stessa macchina, in modo pulito (sono pacchetti deb)
- permette di avere diverse versioni dello stesso kernel in modo comodo senza pasticci nei moduli in `/lib/modules`
- permette di creare diversi pacchetti (header, moduli, immagini, sorgenti)
- permette di usare i pacchetti deb "kernel-patch" senza patchare direttamente l'albero dei sorgenti del kernel

e altro ancora (usare il pacchetto altrove, per altre architetture, ecc.).

uso tipico di kernel-package

- installazione dei pacchetti necessari
- reperimento di un kernel
- configurazione
- creazione del/i pacchetti deb utili

uso (2)

I pacchetti necessari sono facilmente installabili con il solito apt-get, mediante il classico:

Example

```
# apt-get install debhelper modutils kernel-package  
libncurses5-dev fakeroot
```

uso (3)

Potete recuperare un kernel in modo classico (wget da kernel.org), recuperare i sorgenti che sono già pacchettizzati in debian (apt-get kernel-source-xxx). Per kernel-package non fa differenza. (è in grado di gestire le varie possibilità).

USO (4)

Potete usare il metodo di configurazione a cui siete abituati e nelle modalità con cui siete abituati (menuconfig, xconfig, editing manuale ;))⁴.

⁴make-kpkg in assenza di .config, utilizza uno a default, e in fase di compilazione viene invocato comunque oldconfig per gestire eventuali patch

uso (5)

si invocano i due comandi veri e propri:

Example

```
$ make-kpkg clean  
$ fakeroot make-kpkg --append-to-version  
-nomepersonalizzato --revision=1 kernel_image
```

notare che "-nomepersonalizzato" è una etichetta che identifica un dato pacchetto, così come -revision. Il risultato sarà un pacchetto tipo kernel-image-<VERSIONE>-nomepersonalizzato-386_1.Custom_i386.deb da dare direttamente in pasto a dpkg.

anatomia

l'uso tipico di `make-kpkg` (dopo il `clean`) è quindi

1

```
make-kpkg <gestione dei nomi e versioni> <altre opzioni>  
<target>
```

dove `target` è tipicamente `kernel_image`, ma può essere `kernel_headers`, o altro di utile.

Le opzioni che vale la pena citare (oltre a quelle viste) sono:

- `-initrd` per fare creare un `initrd.img`
- `-added-modules` per aggiungere moduli (da pacchetto) residenti in `/usr/src/modules`
- `-added-patches` idem per le patch pacchettizzate

mio cugino

Il compagno ideale di kernel-package è module-assistant. Questo pacchetto permette di scaricare/compilare/installare/ diversi moduli (non inclusi nel kernel), con un interfaccia comoda, interattiva, e **senza** ricompilare tutto il kernel.

Sebbene possieda una modalità non interattiva, il mio consiglio è di lanciarlo, per vedere le varie opzioni.

Se vi piacciono gli schermi che scorrono, lanciate invece "m-a la"⁵

⁵ m-a è un'abbreviazione per module-assistant

mio cugino

Il compagno ideale di kernel-package è module-assistant. Questo pacchetto permette di scaricare/compilare/installare/ diversi moduli (non inclusi nel kernel), con un interfaccia comoda, interattiva, e **senza** ricompilare tutto il kernel.

Sebbene possieda una modalità non interattiva, il mio consiglio è di lanciarlo, per vedere le varie opzioni.

Se vi piacciono gli schermi che scorrono, lanciate invece "m-a la"⁵

⁵ m-a è un'abbreviazione per module-assistant

The Debian Way

- 1 Le premesse di questo intervento
 - Chi e cosa
- 2 Gli strumenti (1)
 - L'allegria famiglia update-*
- 3 Gli strumenti (2)
 - Interfaces
- 4 Gli strumenti (3)
 - kernel-package
- 5 Gli strumenti (4)**
 - **java-package**

java-package

Con lo stesso spirito di kernel-package è stato prodotto questo java-package, per permettere una gestione più efficiente di installazione di ambienti java in Debian.

Java-package attualmente supporta:

- Sun js2e (jdk e jre) versioni 1.4 a 1.5.0
- Blackdown java-linux (jdk e jre) versioni 1.3 a 1.4.2
- Ibm js2e (jdk e jre) versioni 1.3.1, 1.4.2, e 1.5.0

make-jpkg

La creazione del pacchetto .deb, con make-jpkg è al limite del triviale, in sintesi:

Example

```
$ fakeroot make-jpkg jdk-1_5_0_05-linux-i586.bin
```

Il processo produce un pacchetto deb tipo sun-j2sdk1.5_1.5.0+update05_i386.deb da dare in pasto a dpkg

grazie

- A tutti gli intervenuti.
- All'organizzazione dell'evento.