

# Configurare un sistema Debian chroot

Spesso, essendo uno sviluppatore Debian, voglio provare i miei pacchetti prima di effettuarne l'upload. Ma, se i miei pacchetti dovessero avere dei brutti bug che provocano l'equivalente di `rm -rf /`, sarebbe brutta cosa fare i test sulla mia installazione di produzione.

Grazie ad una combinazione dei programmi chroot e debootstrap, è possibile installare un sistema Debian isolato in cui loggarsi ed installare pacchetti, senza il timore di distruggere il proprio sistema. Attualmente, sto usando una soluzione del genere per avere sia GNOME 2 che GNOME 1 installati tramite pacchetti Debian in 2 chroots separati, dove posso provare gli script per la transizione GNOME 1 --> GNOME 2.

## Prerequisiti

- chroot (parte del pacchetto coreutils in Debian `sarge` e `sid`)
- debootstrap
- login
- (*opzionale*) gdm

## Creare il chroot

Usando `debootstrap`, è possibile settare degli ambienti chroot per `woody`, `sarge`, o `sid`. Nel resto del documento si tratterà della creazione di un chroot `sid`.

1. Crea una directory, ad esempio `/sid/`. Se non scegli tale nome, assicurati di sostituire il nome da te scelto a `/sid/` nel resto di questo documento.

```
mkdir /sid/
```

2. Attualmente è necessario installare prima `woody` e successivamente effettuare l'upgrade a `sid`. `sid` cambia troppo rapidamente per essere efficacemente debootstrappabile (è la prima volta che vedi questo aggettivo!). Quindi, lancia `debootstrap`:

```
debootstrap woody /sid/ http://ftp.it.debian.org/debian/
```

3. Configura il file `/etc/fstab` del *tuo sistema stabile* per effettuare il mount del filesystem `/proc` del sistema chroot.

```
echo proc-sid /sid/proc proc none 0 0 >> /etc/fstab
mount /sid/proc
```

4. Configura il networking nel sistema chroot.

```
echo 127.0.0.1 localhost $(hostname) > /sid/etc/hosts
```

5. Ora sei pronto per usare il chroot! Usa il programma `chroot` per entrare nel nuovo sistema.

```
chroot /sid/ /bin/sh
```

6. A questo punto si può fare l'upgrade dell'installazione `woody` a `sid`. Puoi saltare questo

passaggio se vuoi semplicemente **woody**.

```
(echo deb http://ftp.it.debian.org/debian/ sid main; echo deb-src
http://ftp.it.debian.org/debian/ sid main) > /etc/apt/sources.list
apt-get update
apt-get dist-upgrade
```

7. A questo punto dovresti avere un sistema completamente funzionante, dove fare tutte le prove e i pasticci del caso senza il timore di danneggiare il tuo sistema Debian.

## Configurazione di **login**

Dare un semplice **chroot /sid/** è facile, ma così facendo vengono mantenute tutte le variabili d'ambiente che potresti non volere, ecc... Un approccio migliore è quello che consiste nel lanciare un altro processo di **login** su un terminale virtuale, dal quale loggarsi direttamente nella directory **chroot**.

1. Cerca le seguenti linee nel tuo **/etc/inittab**:

```
# Nota che in molti sistemi Debian la tty7 è usata dall' X Window System,
# quindi è consigliabile non modificarne la parte relativa.
#
1:2345:respawn:/sbin/getty 38400 tty1
2:23:respawn:/sbin/getty 38400 tty2
3:23:respawn:/sbin/getty 38400 tty3
4:23:respawn:/sbin/getty 38400 tty4
5:23:respawn:/sbin/getty 38400 tty5
6:23:respawn:/sbin/getty 38400 tty6
```

2. Aggiungi una linea di questo genere:

```
8:23:respawn:chroot /sid/ /sbin/getty 38400 tty8
```

L' "8" sta ad indicare il terminale virtuale da cui effettuare il nuovo login. Puoi sceglierne un altro se preferisci.

3. Ricarica l'init:

```
init q
```

## Configurare **gdm**

Vuoi lanciare **X** e **GNOME** dal tuo **chroot**? E' assolutamente possibile! L'esempio seguente permette a **GDM** di girare su un terminale virtuale 8; dovresti sceglierne un altro (9, ad esempio) se hai assegnato **login** al vt8, come nell'esempio precedente. A tutti i modi, ecco cosa bisogna fare:

1. (passa al tuo **chroot**, o con **chroot /sid/** o *settando login come sopra*)

```
apt-get install gdm gnome x-window-system
```

2. Copia il tuo **/etc/X11/XF86Config-4** dal tuo sistema principale al **chroot**:

```
cp /etc/X11/XF86Config-4 /sid/etc/X11/XF86Config-4
```

3. Loggati ancora nel **chroot**, e modifica **/etc/gdm/gdm.conf**. Nella sezione **[servers]**

troverai una linea come questa:

```
0=Standard vt7
```

Modificala come segue:

```
0=Standard vt8
```

Come visto prima, potresti sostituire ogni terminale di tuo gradimento al posto dell' "8".

4. Sempre dal chroot, lancia:

```
/etc/init.d/gdm start
```

Ora puoi facilmete passare dall'X Window System del chroot a quello del tuo sistema, semplicemente passando da un terminale virtuale all'altro; per esempio premendo `Ctrl-Alt-F7` o `Ctrl-Alt-F8`. Buon divertimento!

---

*Traduzione: Emanuele Rocca [emarocca@libero.it](mailto:emarocca@libero.it)*

Guida pubblicata su <http://linux.le.it>  
project by <http://russo.le.it>  
powered by Vito Russo