

4.2 - Compilare i kernel alla maniera Debian

Articolo di: Jonathan OxerTraduzione di: Marco Frattola
(aka sbarrax)

Sembra che Debian abbia una sua speciale maniera di fare per praticamente quasi tutto, inclusa la compilazione del kernel. Debian ha un sistema che vi permette di compilare il kernel e automaticamente costruirvi attorno un pacchetto Debian, consentendovi di installare poi un nuovo kernel così come installate un qualsiasi altro pacchetto. Ciò significa anche che potete impacchettare un kernel su una macchina e poi installare il pacchetto sulle altre macchine senza dover ricompilare o registrare tutti i fallimenti - ottimo se avete bisogno di aggiornare un buon numero di macchine simili!

Ottenere Le Sorgenti

Potete scaricare i sorgenti del kernel direttamente da kernel.org o da un mirror se preferite, ma naturalmente ci sono perfino i pacchetti delle sorgenti del kernel. Un veloce `apt-cache search kernel-source` vi restituirà una lista di pacchetti delle sorgenti del kernel tutti pronti da usare.

Prendendone uno ad esempio, potete semplicemente dare `apt-get install kernel-source-2.4.20` per ottenere le sorgenti scaricate sul vostro sistema. Finirete con l'averne un archivio in `/usr/src`, che è il posto in cui la maggior parte del lavoro del kernel viene svolto. Adesso potrete estrarre le sorgenti dall'archivio:

```
cd /usr/src
bunzip2 kernel-source-2.4.20.tar.bz2
tar xf kernel-source-2.4.20.tar
```

Tutto ciò vi lascerà con una directory `"/usr/src/kernel-source-2.4.20"`, pronta per la configurazione. Prima create un link simbolico ad essa chiamato "linux", così: `ln -s kernel-source-2.4.20 linux`

Se ci fosse già un tale link simbolico (che puntasse per esempio a delle vecchie sorgenti del kernel), dovrete prima cancellarlo. Adesso potrete ottenere il vostro albero delle sorgenti del kernel semplicemente digitando: `cd /usr/src/linux`

Strumenti che vi servono

Prima di configurare, compilare e impacchettare il vostro nuovo kernel vi dovrete assicurare di disporre di alcuni tools. Per installare i pacchetti principali: `apt-get install kernel-package libcurses5-dev`
Ciò farà sì che vengano installati una serie di altri pacchetti di supporto in base alle dipendenze trovate.

Metodi di configurazione

Ci sono parecchi modi di configurare un kernel prima di compilarlo, e tutti sono invocati come argomenti del comando 'make'. Assicuratevi di avere una shell aperta nella directory delle sorgenti del kernel (dovrebbe essere `/usr/src/linux` se avete seguito le mie precedenti indicazioni) prima di provarne uno qualunque di questi. Andando in ordine crescente di complessità, tali metodi sono:
`editing .config`

La reale configurazione è salvata nella directory dei sorgenti in un file di testo chiamato `'.config'`.

Se veramente volete (o se steste cercando per nome un'opzione trovata in un `'readme'`, del tipo `'CONFIG_PACKET'`) potete editare il file direttamente con un editor di testo come Vi o Anjuta. Ciò dovrebbe essere veramente molto raro.

```
make config
```

L'approccio più basilare, vi farà soltanto una sfilza di domande. una dopo l'altra.

Personalmente odio fare in questo modo, perché è sequenziale e dovete addentrarvi in una marea di opzioni di cui probabilmente non ve ne potrebbe interessare di meno. Approdate a questo modo soltanto come ultima spiaggia, proprio se le altre opzioni, per un qualche motivo, non vi fossero disponibili.

make menuconfig

Questo e' il modo piu' comune per eseguire la configurazione. Mostra un gradevole menu, guidato da tastiera, attraverso il quale potrete navigare usando le frecce, invio e la barra spaziatrice. Comunque sia, un colpo che potrebbe mettervi fuori gioco potrebbe essere il trovarvi ad aver bisogno delle librerie di sviluppo per Ncurses, non delle sole Ncurses. E' per questo che vi ho fatto installare il pacchetto libncurses5-dev qualche paragrafo fa' :-)

make xconfig

il metodo preferito per le macchine con X, e' molto simile al metodo precedente soltanto che e' interamente realizzato con un'interfaccia 'punta-e-clicca' attraverso la navigazione col mouse.

L'unica cosa di cui potreste aver bisogno per farlo funzionare e' tk8.3; se rilevaste errori che dicono di non trovare 'wish', lanciate: apt-get install tk8.3 e tutto dovrebbe sistemarsi.

Configurare il kernel

Qualunque metodo di configurazione scegliate, il passo successivo consta nell'esaminare le impostazioni per le differenti opzioni del kernel e fare le modifiche come richiesto.

C'e' un enorme numero di opzioni del kernel, quindi per aiutarvi a trovare meglio le cose, esse sono state raggruppate logicamente in differenti sezioni maggiori e minori. I diversi metodi di configurazione hanno ognuno il proprio modo di rappresentare questi raggruppamenti, ma dovrebbero essere tutti decentemente immediati da seguire.

Non entrero' qui nei dettagli della configurazione del kernel, perche' quello che vi puo' servire di configurare potrebbe variare enormemente da un caso all'altro e sulla Rete c'e' un'infinita di tutorial in proposito (incluso il Kernel How-To!) che spiegano cio' nel dettaglio. Essenzialmente, dovete esplorare i menu per trovare le opzioni e i moduli che vi serve di abilitare, e impostarli a off (non disponibile), on (compilato direttamente nel kernel) o modulo (compilato separatamente in modo da essere caricato quando necessario).

Una volta conclusa la navigazione attraverso le opzioni, uscite salvando avendo cosi' memorizzato le vostre scelte in un file di configurazione che usato dal compilatore.

Da notare un trucco: i dettagli della reale configurazione sono immagazzinati in un file chiamato '.config' nella directory sorgente del kernel. Una volta che abbiate finito di configurare il kernel a vostro piacimento, muovere verso un nuovo kernel potrebbe essere doloroso se avete iniziato dal nulla il processo di configurazione e quasi certamente vi stareste perdendo qualcosa di fondamentale - so che di solito mi capita! Per rendere le cose piu' facili, potete copiare il file .config nella directory delle sorgenti del nuovo kernel avendo cosi' portato automaticamente tutte le opzioni salvate col vecchio kernel. Quindi vi bastera' eseguire la configurazione semplicemente controllando che tutto sia come vogliate, salvare e procedere come prima. A patto che non stiate passando ad un tipo di kernel totalmente nuovo, questo trucchetto puo' risparmiarvi un sacco di tempo. Attenzione che, se state passando da un kernel ad un altro totalmente differente, come potrebbe essere dal 2.4.xx al 2.5.xx, a causa del fatto che molte opzioni saranno totalmente diverse, cionostante il vecchio file config potrebbe ancora funzionare, potrebbe risultare che alcune opzioni non siano attivate nel nuovo kernel.

Compilare e Impacchettare

A questo punto il kernel viene effettivamente compilato e posto in un pacchetto. Normalmente ci sarebbe molto lavoro da fare, ma grazie ai tools forniti dal pacchetto 'kernel-package' questo e' uno dei passi piu' semplici.

Se avete letto un kernel How-To avrete probabilmente visto una sequenza di comandi come 'make dep && make clean && make bzImage'. Non dovrete fare niente di tutto cio', ma soltanto digitare: make-kpkg kernel-image nella directory /usr/src/linux, ed esso si prendera' cura di tutto. Gli strumenti di impacchettamento del kernel prima compilano il kernel e i moduli come da file di configurazione generato, e poi costruiscono

un pacchetto Debian chiamato 'kernel-image-<version>_<architecture>.deb' al di fuori della directory delle sorgenti (ad esempio, in /usr/src).

Questa fase puo' durare un po' di tempo, dipendentemente dalla velocita' della vostra macchina e di quali moduli avete selezionato. Aspettatevi qualunque cosa, da 10 minuti a mezz'ora, tempo durante il quale vedrete un grande flusso di informazioni di debugging che probabilmente non vi interesseranno, a meno che qualcosa vada storto.

Installare il vostro Pacchetto Kernel

Ci siamo quasi! Quello che avete in mano adesso e' un kernel, compilato con le proprie opzioni personalizzate, e preparato come pacchetto Debian pronto da installare.

Installare il nuovo kernel e' adesso soltanto questione di usare dpkg come root, per installare tale pacchetto come ogni altro pacchetto Debian: `dpkg -i kernel-image-2.4.20_10.00.Custom_i386.deb`
Potra' esservi chiesto di creare un floppy di avvio che utilizzi quell'immagine di kernel; le dozzine di macchine che gestisco non hanno un singolo drive per floppy, percio' sono abituato a rispondere 'no' a questa domanda, ma potete rispondere 'yes' se preferite crearvene uno.

Successivamente vi chiederà se volete installare un blocco d'avvio usando l'esistente lilo.conf: se usate lilo come bootloader (che e' quello di default per Woody) allora rispondete 'yes'.

Per adesso quello che e' effettivamente successo e' soltanto che dpkg abbia copiato l'immagine del nuovo kernel in /boot dove i kernel sono normalmente messi, creato un link simbolico da /vmlinuz al nuovo kernel (potete verificarlo mediante `ls -l /vmlinuz`), modificato il vostro /etc/lilo.conf in modo che Lilo possa trovare il nuovo kernel, spostato il precedente kernel in modo che sia disponibile come ripiego nel caso il nuovo si pianti, e rilanciare Lilo per rendere effettive le modifiche.

Notate che a questo punto non state eseguendo il nuovo kernel: e' solo pronto per essere usato la prossima volta che riavverete la macchina. Installare un nuovo kernel e' una delle veramente poche occasioni in cui sia proprio necessario riavviare un sistema Linux.

Configurare il prompt di Lilo

Lilo permette di avere piu' kernel installati e selezionare quello che si vuole usare. Per agire sicuri, dovrete configurare Lilo in modo da visualizzare una lista di kernel disponibili in modo che se riavviate al nuovo kernel e scopriate di aver commesso un grosso errore potete semplicemente riavviare di nuovo e selezionare il precedente kernel dalla lista proposta da lilo.

Aprirete /etc/lilo.conf con un editor di testo, e assicuratevi di avere delle linee simili alle seguenti:

```
prompt
delay=100
timeout=100
```

Quelle linee ci saranno gia', ma commentate (e probabilmente con dei valori di delay diversi). Tutto cio' che fanno e' dire a Lilo di costituire una lista di kernel disponibili al boot, ed aspettare 10 secondi prima di procedere all'avvio del kernel predefinito.

Riavviare e Provare

E' ora della grande prova! Se state eseguendo X, uscite dalla sessione e riavviate. Quando Lilo si carichera', vi verra' presentata la lista di kernel disponibili, incluso il nuovo.

Selezionatelo e premete Invio.

Una volta avviato, potete controllare i messaggi di boot per vedere cosa ha caricato il nuovo kernel mediante i log di dmesg: `less /var/log/dmesg` Se tutto e' andato come ci si aspettava, congratulazioni! Avete appena compilato e installato un kernel personalizzato "alla maniera Debian".

Installare su altre macchine

Qua e' dove la convenienza di compilare il kernel alla 'Maniera Debian' diventa piu' lampante. Se volete installare il vostro kernel personalizzato su altre macchine, il processo e' molto semplice: semplicemente copiate il pacchetto .deb che avete creato sulla macchina di destinazione, ed installatelo usando `dpkg` esattamente come avete fatto prima. Semplice!

Non c'e' bisogno di compilare il kernel su tutte le macchine, o copiare le sorgenti, o nemmeno avere un compilatore installato su di esse. Tutto il lavoro duro e' stato fatto una sola volta su una macchina, e non c'e' bisogno di ripeterlo.

State pronti per il prossimo articolo de "L'Universo Debian". Verranno ancora fuori peculiarita' sulla gestione dei pacchetti, il caching dei pacchetti, e su FAI (Fully Automatic Installer, l' "installatore totalmente automatico")

Questo articolo e' un estratto dal nascente libro "The Debian Universe" (L'Universo Debian) di Jonathan Oxer. Maggiori informazioni disponibili al link <http://www.debianuniverse.com>

Jonathan e' il Fondatore e il Direttore Tecnico di Internet Vision Technologies, un'agenzia di sviluppo applicazioni web australiana con clienti in tutto il mondo. E' anche uno sviluppatore Debian, maintainer di apt-cacher, ed e' stato organizzatore di Debian Mini-Conf a Perth (AUS) nel gennaio 2003, in collaborazione con Linux Conf Australia dove ha presentato uno dei suoi scritti tecnici. Ha compiuto vari interventi a seminari sia aziendali sia governativi.

Il suo primo libro, "How To Hire A Web Developer And Stay Sane" (letteralmente: "Come impiegare uno sviluppatore web e rimanere sani"), sara' presto pubblicato, mentre il secondo ed il terzo ("How To Deploy Web Applications And Stay Sane" e "Disaster Proofing For Small Networks") sono gia' in corso d'opera. I suoi articoli sono stati tradotti per "el espirografo", la guida messicana per gli utenti Debian Gnu/Linux.

Tutorial prelevato da <http://linux.le.it> powered by <http://russo.le.it>