

Guida alla ricompilazione di Kernel Linux

Di Zorzan Urban Pawel (okno)



Compilare un kernel linux è un'operazione fondamentale per rendere più performante il nostro computer. Compilare il kernel significa dire al cuore del sistema operativo quali device sfruttiamo, i moduli che ci interessano, il tutto per alleggerire al massimo un kernel precompilato, che ha di base una miriade di moduli inutili caricati.

In più la compilazione del kernel adatta il cuore del sistema operativo perfettamente al nostro computer per sfruttare a pieno le sue potenzialità.

Innanzitutto è utile sapere precisamente l'hardware presente nel nostro computer in modo da velocizzare le scelte in fase di compilazione. Per fare ciò basta lanciare il comando :

```
# lspci
```

E scriverci le periferiche collegate al nostro computer.

Partiamo con il primo passo:

-Passo 1

Scaricare l'ultima versione del kernel disponibile e stabile!

quindi andiamo nella dir classica per i sorgenti :

```
# cd /usr/src
```

e poi mettiamo a scaricare il nuovo kernel :

```
# wget http://www.kernel.org/pub/linux/kernel/v2.6/linux-x.y.z.tar.bz2
```

Naturalmente dovete sostituire x y e z con la versione del kernel che state per scaricare!

-Passo 2

Estraiamo il file appena scaricato nella dir dei sorgenti linux:

```
# tar -xjvf linux-x.y.z.tar.bz2 -C /usr/src
```

-Passo 3

Assicuratevi di avere il compilatore caricato sul vostro linux!

Per controllare lanciate il comando :

```
# gcc -v
```

E dovrete vedere un output tipo :

```
Reading specs from /usr/lib/gcc/i486-slackware-linux/3.4.6/specs
```

```
Configured with: ../gcc-3.4.6/configure --prefix=/usr --enable-shared --enable-threads=posix --enable-__cxa_atexit --  
disable-checking --with-gnu-ld -- verbose --target=i486-slackware-linux --host=i486-slackware-linux
```

```
Thread model: posix
```

```
gcc version 3.4.6
```

Se questo non accade dovete prima di tutto installare gcc sul vostro sistema linux.

-Passo 4

Ora possiamo cominciare alla compilazione del kernel.

posizioniamoci nella directory con il nuovo kernel :

```
# cd linux-x.y.z
```

I comandi per la configurazione del vostro nuovo kernel ci sono vari tipi di menu e sono i seguenti :

```
#make config // Configurazione a domande consecutive (molto lungo e complicato)
```

```
#make xconfig // Configurazione basata sulle librerie QT in genere per Desktop KDE
```

```
#make gconfig // Configurazione basata sulle librerie GTK in genere per Desktop Gnome
```

```
#make menuconfig // Configurazione a menu testuale (semplice e intuitivo, il mio metodo di configurazione preferito)
```

Ora non vi resta che decidere come configurare il vostro kernel, controllate bene quello che abilitate e disabilitate, molti moduli sono vitali per il sistema operativo, comunque per ogni voce è presente il tasto HELP, se avete dei dubbi non esitate a premerlo.

-Passo 5

Compilazione del kernel.

Creiamo l'immagine compressa del nuovo kernel (può richidere diverso tempo a seconda del vostro processore):

```
# make
```

Compiliamo i moduli del nostro kernel (può richidere diverso tempo a seconda del vostro processore):

```
# make modules
```

Installiamo i nuovi moduli nel nostro sistema Linux.

Dobbiamo avere i privilegi di amministratore quindi a seconda della vostra distribuzione loggatevi come root:

```
# su -
```

```
# cd /usr/src/linux/linux-x.y.z
```

```
# make modules install
```

-Passo 6

Installiamo il nuovo kernel!

```
# make install
```

Verranno creati 3 file nella directory /boot per permettere al nuovo kernel di avviarsi. i file sono :

```
System.map-x.y.z
```

```
config-x.y.z
```

```
vmlinuz-x.y.z
```

-Passo 7

Creiamo l'immagine initrd, questo file contiene tutti i driver dei dispositivi per permettere al sistema operativo di avviarsi, non tutti i sistemi ne necessitano la creazione, ma per sicurezza è sempre meglio crearlo :

```
# cd /boot
```

```
# mkinitrd -o initrd.img-x.y.z x.y.z
```

-Passo 9

Ora non ci resta che istruire il nostro boot loader per avviarsi con il nuovo kernel.

Se utilizzate Grub seguite il Passo 9.1 per lilo invece il Passo 9.2.

-Passo 9.1

Editate il file /boot/grub/menu.lst :

```
# vi /boot/grub/menu.lst
```

Agiungete il seguente testo :

```
title Linux, kernel 2.6.25 Default
```

```
root (hd0,0)
```

```
kernel /boot/vmlinuz root=/dev/hdXX ro
```

```
initrd /boot/initrd.img-x.y.z
```

```
savedefault
```

```
boot
```

Attenzione a inserire la partizione corretta nella quale è installato il vostro kernel, quindi sostituite /dev/hdXX per esempio con /dev/hda1.

Se pensate che sia troppo complicato editare il file a manina lanciate il comando di update di grub e dovrebbe fare tutto da solo :

```
# update-grub
```

-Passo 9.2

Editate il file /etc/lilo.conf

```
# vi /etc/lilo.conf
```

In genere la configurazione generale della partizione in lilo.conf stà nella testata quindi a voi basta asaggiungere tra le opzioni di scelta del kernel le seguenti righe:

```
image = /boot/vmlinuz-x.y.z
```

```
label = linux
```

Opzioni aggiuntive :

```
vga = 791 // Avvia linux in un framebuffer a 1024x768
```

```
root = /dev/hdXX // Dichiaro in che partizione è presente il kernel
```

Con questo abbiamo terminato l'installazione del nostro nuovo kernel!

Ora non ci resta che riavviare:

```
# reboot
```

Spero di avervi fornito tutti i dati fondamentali per la compilazione del kernel.

Per informazioni e delucidazioni contattatemi all'indirizzo mail : mail@pawelzorzan.eu

La ripubblicazione di questo articolo è consentita ma solo citandone la fonte :

Zorzan Urban Pawel – <http://www.pawelzorzan.eu>

Ringraziamenti :

IceShock : www.iceshock.org

King-Infet : www.kinginfet.net