

Ricompilare il Kernel 2.4

Raoul Scarazzini – rascasoft@tiscali.it, <http://web.tiscali.it/rascasoft>
v 1.1, Mercoledì 09 13:45:02 CEST 2005

Questo articolo illustra come sia possibile configurare e compilare il Kernel di Linux in modo da avere un sistema ottimizzato per le proprie esigenze.

Introduzione

Tutti sanno che è il Kernel rappresenta il cuore di Linux, l'insieme cioè di tutti i programmi che consentono alla parte software di colloquiare con l'hardware. E' il Kernel a permetterci di avviare il sistema con il processo base "init" dal quale tutti gli altri processi nascono.

Ricompilare il Kernel risulta essenziale per avere un sistema operativo configurato ad-hoc con lo specifico hardware del nostro PC ed è inoltre un passo importante per capire meglio quello che è Linux e quale sia il suo maggiore punto di forza.

Prima di cominciare è bene però specificare come non sia detto che alla prima ricompilazione si possa già dichiarare "vittoria". Capire le dinamiche del processo di ricompilazione richiede applicazione e costanza, anche se una volta acquisita la necessaria dimestichezza ci si accorgerà che questo tipo di operazione sarà tanto semplice quanto "all'ordine del giorno".

Capire che hardware è installato nel nostro sistema

L'ideale sarebbe proprio prepararsi un bell'elenco che parta dal modello di processore, il modello di scheda madre, quello della scheda video e delle altre schede particolari (schede di rete, schede audio, etc. etc.).

Questa parte di lavoro (forse la più noiosa) è necessaria e fondamentale. Ricompilare un Kernel con impostazioni diverse da quelle reali della macchina sarebbe un lavoro stupido, inutile e quasi sicuramente non renderebbe stabile il nostro sistema.

Un buon aiuto potrebbe essere rappresentato dal comando :

```
# cat /proc/cpuinfo  
# cat /proc/pci
```

che visualizzano rispettivamente la tipologia di processore e la lista delle schede PCI presenti nel sistema.

Scaricare i sorgenti

Collegandosi al sito <http://www.kernel.org> sarà possibile scaricare l'ultima versione stabile del Kernel che al momento in cui scrivo è la 2.4.29 :

```
# cd /usr/src  
# wget http://www.kernel.org/pub/linux/Kernel/v2.4/linux-2.4.29.tar.bz2
```

Dentro a questo grosso file (più di 20 mega) sono contenuti i sorgenti di tutti i programmi che servono a far funzionare il nostro PC.

A questo punto potrebbe sorgere una domanda : Perché scaricare i sorgenti dal sito kernel.org e non utilizzare i sorgenti del Kernel che si trovano nella propria distribuzione ? La questione è opinabile. Il Kernel di www.kernel.org è quello ufficiale, solo la distribuzione Debian lo abbina così come lo si trova in internet, le altre lo modificano ad-hoc. Si possono usare entrambi i Kernel (anche perché quello personalizzato dalle distribuzioni è una derivazione di quello ufficiale), ma c'è una ragione in più per utilizzare il Kernel ufficiale :

Ricompilare il Kernel 2.4

se in futuro si avrà l'esigenza di applicare patch al Kernel, queste si rifaranno ai sorgenti ufficiali e pertanto saranno sempre applicabili.

Monolitico o modulare ?

Prima di effettuare la compilazione si dovrà fare una scelta : se creare un Kernel monolitico o modulare. Un Kernel monolitico è composto da una "immagine", in cui tutti i programmi necessari al funzionamento della macchina sono racchiusi in un unico file compresso che viene scompattato all'avvio e caricato in memoria. Con un Kernel modulare invece, si adotta la strategia di utilizzare un'immagine ridotta che si appoggia su tanti programmi denominati "moduli" che vengono caricati all'occorrenza.

Il confronto tra questi due sistemi è presto fatto : Da un lato avendo già tutto in memoria (Kernel monolitico) si favorisce la velocità a discapito dell'impiego fisico della memoria di sistema. Nel secondo caso invece (Kernel modulare) l'immagine è ridotta e questo comporta minor memoria impiegata in fase di avvio ed esecuzione e tempi di caricamento dipendenti dal numero di moduli necessari.

Generalmente si usano Kernel modulari (lo fanno tutte le più diffuse distribuzioni), ma nulla vieta di creare Kernel monolitici, soprattutto se si riescono ad ottenere immagini di dimensioni non eccessive.

Quello che creeremo insieme sarà un Kernel modulare, poiché questa tipologia presenta un'ulteriore vantaggio decisivo : se si ha la necessità di aggiungere una funzionalità, non è necessario ricompilare totalmente il Kernel, basterà solamente ricompilare la parte relativa ad i moduli. In macchine non recenti, dove una ricompilazione può durare anche ore, questo vantaggio è non è affatto indifferente.

Scompattare il kernel ed avviare il tool di configurazione menuconfig

Una volta scaricato l'unico file compresso del Kernel è necessario scompattarlo in una directory (che generalmente è /usr/src) con questo comando :

```
# tar -jxvf linux-2.4.29.tar.bz2
```

Questo comando creerà la directory di lavoro linux-2.4.29. E' prassi, oltre che molto utile creare un link simbolico a questa directory denominato semplicemente "linux" :

```
# ln -s linux-2.4.29 linux
```

Infine, c'è da fare un'ultima cosa prima di iniziare a configurare il Kernel : personalizzare la versione corrente in modo da renderla unica. Questo si ottiene modificando il file Makefile presente nella directory del Kernel :

```
# cd linux-2.4.29  
# vi Makefile
```

Le prime righe sono così composte :

```
VERSION = 2  
PATCHLEVEL = 4  
SUBLEVEL = 29  
EXTRAVERSION =
```

Per personalizzare la versione basterà assegnare un nome alla "EXTRAVERSION", ad esempio "rasca". Il Kernel, d'ora in avanti, sarà 2.4.29rasca.

Per quanto possa sembrare una sottigliezza, questa fase è molto importante, in quanto renderà unico il Kernel e ne faciliterà la convivenza con altri sullo stesso sistema.

A questo punto si può procedere con la configurazione eseguendo questo comando :

Ricompilare il Kernel 2.4

make menuconfig

Quello che dovrebbe apparire è il menu contenente tutte le voci di configurazione. Non c'è molto da spiegare su questo strumento, di semplice utilizzo : con "invio" si entra nelle relative voci, premendo la barra spaziatrice si modifica lo stato di un'opzione.

Ogni opzione ha essenzialmente 3 stati :

- vuoto, ossia non verrà considerato nella compilazione;
- * ossia verrà integrato nel Kernel;
- M, verrà configurata come modulo.

Configurare nel dettaglio

Una cosa importante da fare nel dubbio sul se e come settare un'opzione del Kernel è sicuramente consultare l'help. Per farlo basta spostare la freccia sulla scritta "help" e premere la barra spaziatrice. Qui si trovano importanti informazioni e consigli che soprattutto le prime volte risultano essenziali.

Code maturity level options --->

Partendo dall'alto, questa è la prima voce importante una volta eseguito `make menuconfig`. Se non si abiliterà (premendo spazio) l'opzione "Prompt for development and/or incomplete code/drivers" si potranno settare solo le opzioni relative ai moduli stabili, ossia che non sono in fase di sviluppo o sperimentali. Settare questa opzione è decisamente importante : molti moduli pur essendo ritenuti appunto "in fase di sviluppo" o "sperimentali" possono tranquillamente essere utilizzati.

Loadable module support --->

E' bene abilitare anche le tre opzioni di questo sottomenu, sono molto autoesplicative e per capire meglio la loro importanza, basterà fare riferimento all'help.

Processor type and features --->

In quest'area si inizierà a personalizzare davvero il Kernel, indicando il modello del processore e tutte le opzioni ad esso correlate. E' possibile includere senza problemi le due opzioni "Math emulation" e "MTRR (Memory Type Range Register) support" il resto va valutato in base all'hardware installato nella macchina. Se si è svolto un buon lavoro di "raccolta informazioni" si farà in fretta a settare ciò che interessa.

General setup --->

Presenta appunto le opzioni generali del Kernel. Già da questo punto si può fare in modo di alleggerire il Kernel, ad esempio per quanto riguarda la voce `pcmcia`, non selezionando nulla se non si dispone di un controller pcmcia nel sistema (nei PC desktop è generalmente sempre così).

Proprio da qui che parte una buona ricompilazione : eliminando cioè tutto ciò che non serve, alleggerendo cioè il carico di componenti da includere nel sistema. Si potranno comunque effettuare aggiunte in ogni momento. Le parti che vanno sicuramente abilitate sono quelle relative alla rete, ai tipi di BUS ed al System V. Se si possiede un portatile diventa fondamentale abilitare i moduli relativi all'APM, Advance Power Management BIOS support.

Parallel port support --->

Saltando la parte relativa all'MTD (da abilitare solo se si possiedono periferiche di questo tipo), si può procedere includendo come Moduli le parti "generali" relative alla porta parallela : "Parallel port

Ricompilare il Kernel 2.4

supporto, "PC-style hardware" e "Support foreign hardware". Le altre opzioni sono dettagliate e relative a tipi particolari di periferiche (nel dubbio, leggere l'help non guasta mai)

Plug and Play configuration --->

Anche per questa opzione si può abilitare senza problemi tutto ciò che ci viene presentato.

Block devices support --->

In quest'area andranno inclusi come moduli i supporti per i dispositivi a blocchi, generalmente una volta abilitato "Normal PC floppy disk support", "Loopback device support", "Network block device support".

Multi devices support (RAID and LVM) --->

A seconda della necessità di utilizzare dispositivi gestiti con software RAID o LVM, andranno selezionate le opzioni relative.

Networking options --->

Salvo esigenze particolari (come l'utilizzo iptables, per il quale occorrerà specificare i moduli relativi nell'area "IP: Netfilter Configuration") andranno bene i valori di default.

Telephony Support --->

Questo supporto andrà abilitato solo in caso si possieda dell'hardware specifico.

ATA/IDE/MFM/RLL support --->

Questo supporto va incluso come modulo (la maggior parte dei dischi appartiene a questa categoria). Le opzioni specifiche nel sotto menù "IDE, ATA and ATAPI Block devices --->" vanno considerate (come sempre) in base all'hardware che si possiede.

SCSI support --->

Questa parte andrà considerata se si possiedono periferiche scsi o periferiche che funzionano in emulazione scsi (tipo un harddisk esterno usb).

Fusion MPT device support --->

IEEE 1394 (FireWire) support (EXPERIMENTAL) --->

I2O device support --->

Da considerare solo in caso di esigenze di hardware specifico.

Network device support --->

All'interno di quest'area sarà necessario abilitare la compilazione del modulo relativo alla scheda di rete (se ne si possiede una). Si potrebbe necessitare dei moduli relativi al PPP, in modo da poter usare il demone pppd per effettuare connessioni ad internet.

Amateur Radio Support --->

IrDa (infrared) support --->

Ricompilare il Kernel 2.4

ISDN subsystem ---->

Old CD-ROM drivers (not -scsi, not IDE) ---->

Da considerare solo se si possiede l'hardware menzionato.

Input core support ---->

E' bene abilitare i moduli relativi a quest'area, torneranno utili in caso di tastiere o mouse usb, joystick e così via.

Character devices ---->

In questo gruppo tutto va configurato a seconda dell'hardware installato. Si potranno abilitare, ad esempio, i moduli relativi alla scheda grafica installata nel sistema.

Multimedia devices ---->

Da abilitare per utilizzare dispositivi Video o Radio supportati.

File Systems ---->

Fondamentale è il "Second extended fs support" e se si è scelto ext3 come filesystem di Linux, anche "Ext3 journalling file system support". E' bene compilare queste componenti come parte integrante del Kernel e non come moduli, mettendo un `*` anziché una `M` nella selezione.

Il resto va scelto a seconda delle esigenze.

Da abilitare nel sottomenu "Network filesystems" se si pensa di voler utilizzare NFS o SAMBA, i moduli relativi.

Console drivers ---->

Da abilitare se si ha la necessità dell'utilizzo della modalità "Framebuffer" all'avvio.

Tutti i moduli relativi al Framebuffer andranno compilati come parte integrante del Kernel (quindi settati ad `*`). Ciò è reso necessario dal fatto che se si caricano i moduli a macchina già avviata, la modalità framebuffer non si avvierà.

Sound ---->

Questa parte comprende i moduli relativi alla scheda audio. E' utile compilare anche le parti relative a "OSS sound modules" e "Loopback MIDI device support", in questo modo la parte audio del nostro sistema risulterà più completa.

USB Support ---->

Se si possiede una controller USB è bene abilitare le opzioni relative a questo supporto. In particolare bisogna prestare attenzione a "USB Host Controller Drivers" e abilitare lo stesso modello che si verifica di possedere. Per ogni periferica USB che si penserà di collegare, andrà abilitata la compilazione dei moduli relativi.

Bluetooth support ----> *kernel hacking* ----> *library routines* ---->

Da considerare solo se si è a conoscenza di quello che si sta facendo.

Ricompilare il Kernel 2.4

A questo punto è necessario uscire dal programma di configurazione ricordandosi di salvare le opzioni selezionate che saranno registrate nel file nascosto `œ.config` presente nella root directory dei sorgenti del Kernel.

Avviare la compilazione

Per avviare la compilazione, sarà necessario eseguire i seguenti comandi :

```
# make dep
```

necessario a risolvere le dipendenze

```
# make clean
```

per pulire residui di codice

```
# make bzImage
```

per creare l'immagine compressa di sistema

```
# make modules
```

per creare i moduli

```
# make modules_install
```

per installare i moduli. Nel dettaglio, questi verranno installati nella directory `/lib/modules/2.4.29rasca` dove ovviamente 2.4.29rasca è la versione del Kernel che abbiamo indicato in `/usr/src/linux/Makefile`.

Potrebbe essere conveniente includere tutti i comandi da svolgere in uno script, nominato, ad esempio, `kcompile.sh` e strutturato in questo modo :

```
#!/bin/bash
```

```
cd /usr/src/linux;
```

```
nice --20 make dep &nice --20 make clean &nice --20 make bzImage &nice --20 make modules &nice --20 make modules_install &echo -en "\n\naTutto ok ... \n" || ( echo -e "Compilazione fallita ... \n" &for i in `seq 1 50`; do echo -en "\a"; sleep 0,2; done );
```

Questo script eseguirà in sequenza i comandi di ricompilazione. In caso di successo effettuerà un singolo beep e riporterà la scritta "Tutto ok ...", in caso contrario, quindi in caso di un errore di compilazione, effettuerà 10 beep e la scritta "Compilazione fallita ...".

Per avviare il tutto, basterà settare per il file `kcompile.sh` abbia i permessi di esecuzione :

```
# chmod 755 kcompile.sh
```

e lanciare il comando :

```
# ./kcompile.sh
```

Avviare la nuova versione del kernel

Ricompilare il Kernel 2.4

Se la compilazione è andata a buon fine, rimangono da copiare l'immagine del Kernel (bImage) ed il file delle dipendenze (System.map) nella directory boot e fare in modo che questa possa essere avviata (aggiornando il bootloader) :

```
cp /usr/src/linux/arch/i386/boot/bzImage /boot/vmlinuz-2.4.29rasca
cp /usr/src/linux/System.map /boot/System.map-2.4.29rasca
```

Particolare attenzione va data al nome dei file nella directory di boot : System.map dovrà chiamarsi PER FORZA con il nome del Kernel in coda (System.map-2.4.29rasca, appunto) pena il probabile malfunzionamento del sistema.

Per aggiornare il boot-loader LILO basterà aggiungere queste righe al file /etc/lilo.conf :

```
image=/boot/vmlinuz-2.4.29rasca
label=Linux-2.4.29rasca
read-only
root=/dev/hda1
```

E lanciare il comando "lilo", mentre per aggiornare il boot-loader grub, basterà aggiungere queste righe in /boot/grub/grub.conf :

```
title Linux 2.4.29rasca
root (hd0,0)
kernel /boot/vmlinuz-2.4.29rasca ro root=/dev/hda1
```

A questo punto il sistema è pronto per essere riavviato. Sarà difficile aver tutto perfettamente funzionante alla prima ricompilazione, ma un buon punto di partenza potrebbe essere che il sistema si avvii. Le parti mancanti (moduli o semplicemente funzionalità) si potranno aggiungere in futuro con una nuova ricompilazione.

Conclusioni

In caso di problemi è sempre utile consultare i gruppi di google <http://groups.google.com>, tenere come riferimento i siti <http://www.kernel.org> per i sorgenti e la documentazione in inglese e <http://ildp.pluto.linux.it/HOWTO/Kernel-HOWTO.html> per un ottimo Kernel howto in italiano.

Se si riscontrano errori nel documento, si necessita di aiuto o semplicemente si vuole esprimere un commento questo è il mio indirizzo rascasoft@tiscali.it.

(C) by Raoul Scarazzini – <http://web.tiscali.it/rascasoft>. I testi contenuti in questo documento possono essere distribuiti e pubblicati liberamente se non si trae lucro dalla loro distribuzione e se non ne viene alterato in alcun modo il contenuto.