

Guida didattica alla compilazione del kernel Linux

Andrea Lazzarotto - <http://andrealazzarotto.com>

Questo lavoro è licenziato sotto una licenza Creative Commons Attribution Share-Alike

Il testo completo della licenza è consultabile all'indirizzo <http://creativecommons.org/licenses/by-sa/2.5/it/>



Disclaimer

La qui presente guida è concepita esclusivamente per fornire un'idea a scopo didattico di come si effettua la compilazione del kernel Linux. Non è pertanto da ritenersi adatta per ottenere informazioni su come creare un kernel seriamente funzionante e utile all'uso abituale.

Si comincia... Se siete così temerari da voler continuare l'avventura dopo l'avvertenza riportata mi complimento con voi. La guida è divisa in sezioni, che si possono seguire separatamente, l'importante è rispettare l'ordine giusto.

Scaricare i sorgenti

Naturalmente la prima cosa necessaria è l'archivio contenente il codice sorgente del kernel. Questo si scarica dal sito www.kernel.org ed è denominato kernel *vanilla*, ovverosia quello ufficiale, senza le modifiche fatte da alcuni membri della comunità (alcune sono molto famose e si trovano in svariate distribuzioni Linux).

Dal sito è possibile esplorare il server ftp o quello http. Entrambi contengono gli stessi file, e sono veramente molti! Ciò che a noi interessa si trova in <http://www.kernel.org/pub/linux/kernel/v2.6/>. Dentro a questa cartella del sito possiamo trovare gli archivi dei vari kernel del ramo 2.6. Una volta scelta la versione desiderata, poniamo ad esempio 2.6.27.9, è possibile scaricare l'archivio cercando il file *linux-2.6.27.9.tar.bz2*.

È indifferente scaricare il file omonimo con estensione *.tar.gz*, tuttavia esso è compresso in modo differente pertanto ha una dimensione leggermente maggiore. Si tratta solo di risparmiare un po' di tempo per il download.

Il file va scaricato in una directory a piacimento, qui si presuppone che esso sia stato scaricato nella cartella home dell'utente (che si può indicare in un terminale con ~).

Scompattare l'archivio

A questo punto è il caso di armarci del nostro amato terminale. Una volta individuata la directory in cui è presente il kernel compresso, è sufficiente assicurarsi di spostarci in essa con il comando `cd`.

```
cd ~
```

Ora bisogna estrarre il tutto.

```
bunzip2 linux-2.6.27.9.tar.bz2
```

```
tar xvf linux-2.6.27.9.tar
```

Ricordo che il primo comando sostituisce il file `.tar.bz2` il suo equivalente non compresso `.tar`, mentre il secondo estrae dall'archivio i file e crea la giusta struttura di directory, mantenendo il file `.tar`.

A questo punto si può entrare nella directory:

```
cd linux-2.6.27.9
```

La fase di configurazione

Per configurare i parametri con cui sarà compilato il kernel si esegue il comando:

```
make menuconfig
```

In caso nel sistema non siano installate le librerie `ncurses` (e relativi file di sviluppo) il precedente comando restituirà errore. Pertanto dovrete ricorrere ad una configurazione assai più spartana:

```
make config
```

La selezione dei parametri è un procedimento lungo e che richiede una certa ponderazione. La trattazione di questa parte richiederebbe svariate competenze nonché un ingente numero di righe di spiegazione. Ciò esula dallo scopo di questo testo. Ritengo comunque molto istruttivo il fatto che chi compila provi diverse configurazioni al fine di notarne le differenze e capire il funzionamento.

In sostanza si tratterà di decidere che cosa andrà compilato ed incluso nel kernel, che cosa non dovrà essere compilato, ed infine cosa andrà compilato ma utilizzato come modulo. Si vedrà meglio in seguito che cosa significa tutto ciò.

Finita la configurazione, possiamo procedere alla compilazione propriamente detta.

La compilazione

Prima di tutto vi avverto che in questo paragrafo ci saranno dei procedimenti che richiederanno molto tempo, per cui assicuratevi di avere il computer a disposizione senza interruzioni, nonché di potervi svagare in qualche modo finché il processore lavora!

Infatti è arrivato il momento! Un semplice comando stresserà il vostro processore per diverso tempo e compilerà ciò che avete impostato:

```
make
```

Il procedimento ha un tempo variabile, che dipende soprattutto dal vostro processore, comunque vi posso indicativamente dire che su un computer moderno medio ci mette circa un'ora. Naturalmente, salvo eventuali errori!

Immagine compressa, moduli e ramdisk

Il più è fatto. Ora è il momento che vi faccia una piccola digressione su come è composto il kernel Linux. Esso è stato progettato come monolitico, ovvero un grande file che contiene tutto. Successivamente è stato però aggiunto il sistema dei moduli.

Questo consente di evitare di gonfiare come un pallone il kernel che nella fase di boot viene caricato in ram. In questo modo, ciò che non è essenziale subito può essere caricato dopo.

In questo momento abbiamo compilato tutto ma non abbiamo ancora messo al loro posto né i moduli, né l'immagine del kernel (ovvero il grosso file principale). A dire la verità, quest'ultima dobbiamo ancora crearla. Per questo, si esegue:

```
make bzImage
```

Questo processo è decisamente più breve della compilazione, in quanto si tratta sostanzialmente di creare un archivio compresso. Fatto ciò, nella sottodirectory *arch/---/boot/* si troverà il file senza estensione *bzImage*. Al posto dei tre trattini, va la sigla che indica l'architettura per cui è stato compilato il kernel (nel nostro caso, quella che stiamo eseguendo). Per i sistemi a 32 bit, quelli più comunemente usati, è x86. In ogni caso basta provare ad entrare nelle directory e verificarne il contenuto (tra l'altro durante la creazione dell'immagine compaiono diverse righe di output che indicano l'architettura, per cui è difficile sbagliare).

Creato il kernel, è il momento di installare i moduli. Avendo dato in precedenza il comando *make* senza nessuno dei possibili argomenti, essi sono già stati compilati: è sufficiente installarli. Finora però vi ho fatti agire come utenti normali. Per installare qualsiasi cosa avete bisogno dei permessi di amministratore. Al fine di ottenere una console di amministratore date il comando:

```
su
```

Inserite la password dell'utente *root*. In alcune distribuzioni, in particolare quelle come Debian e derivate (esempio: Ubuntu), l'utente amministratore può essere disabilitato, e in alternativa è possibile che il vostro utente sia abilitato a operazioni privilegiate tramite autenticazione. Per verificarlo, eseguite (dopo aver fallito il comando di cui sopra):

```
sudo -s
```

In questo caso dovrete inserire la password del vostro account.

Una volta in possesso dei poteri richiesti, potete finalmente installare i moduli:

```
make modules_install
```

Ecco fatto. Manca solo un'ultima cosa: un ramdisk. Si tratta, come può suggerire il termine, di una porzione di ram che viene utilizzata come un disco. Nello specifico, questo avviene nella fase di avvio. Il kernel risiede nella ram fino a che non è disponibile il disco fisso reale del computer. Questa è la funzione del file denominato *initrd.img*.

Per crearlo, ora che siete amministratori, eseguite:

```
mkinitramfs -o /boot/initrd.img-2.6.27.9 2.6.27.9
```

Come potete intuire, in questo modo il file viene già collocato nella sede giusta, mentre il secondo parametro indica la versione del kernel di cui generare il ramdisk.

Ora tutto è stato compilato, ed i moduli sono già al loro posto. Manca solo da piazzare adeguatamente alcuni files, e poi abilitare il kernel.

Copia dei file

I file relativi al kernel devono essere posizionati nella directory `/boot` che contiene tutto il necessario per l'avvio del sistema operativo.

Saranno da inserire: l'immagine del kernel, un file che mappi il sistema, ovvero sia un registro degli indirizzi di memoria relativo a variabili e funzioni. Inoltre è consigliabile copiare anche la configurazione usata, sia perché può tornare utile nel caso sia funzionante, sia per analizzare eventuali errori successivi. Come accennato, il ramdisk è già al suo posto.

È possibile verificare il contenuto della directory al fine di avere un'idea sul da farsi:

```
ls -l /boot/  
totale 19972  
-rw-r--r-- 1 root root 507665 2008-11-21 00:46 abi-2.6.27-5-generic  
-rw-r--r-- 1 root root 91364 2008-11-21 00:46 config-2.6.27-5-generic  
drwxr-xr-x 2 root root 4096 2008-12-13 20:54 grub  
-rw-r--r-- 1 root root 8192633 2008-12-17 22:18 initrd.img-2.6.27.9  
-rw-r--r-- 1 root root 8192548 2008-12-13 20:54 initrd.img-2.6.27-5-generic  
-rw-r--r-- 1 root root 124152 2008-09-11 22:11 memtest86+.bin  
-rw-r--r-- 1 root root 1029585 2008-11-21 00:46 System.map-2.6.27-5-generic  
-rw-r--r-- 1 root root 1073 2008-11-21 00:48 vmcoreinfo-2.6.27-5-generic  
-rw-r--r-- 1 root root 2244304 2008-12-13 20:03 vmlinuz-2.6.27-5-generic
```

La parte rientrata indica l'output del comando.

I file nominati con un'altra versione del kernel sono relativi al kernel che sta eseguendo il sistema in uso. È da notare che volendo ci sono anche altri file che possono essere messi.

Per copiare il kernel vero e proprio nella sua destinazione eseguite:

```
cp arch/x86/boot/bzImage /boot/vmlinuz-2.6.27.9
```

Modificate il comando dato in caso utilizzaste una architettura diversa.

Il file di mappatura prima menzionato si chiama *System.map*. Lo potete copiare facendo:

```
cp System.map /boot/System.map-2.6.27.9
```

Ora rimane solo da copiare il file con la configurazione utilizzata:

```
cp .config /boot/config-2.6.27.9
```

Vi ricordo che i file che cominciano con il punto sono nascosti. Pertanto, per vederlo elencato nel terminale insieme al restante contenuto della directory, dovete fare:

```
ls -a
```

Ora che tutto è stato copiato, si può verificare controllando il contenuto della directory:

```
ls -l /boot/  
[...]  
-rw-r--r-- 1 root root 82474 2008-12-18 14:26 config-2.6.27.9  
[...]  
-rw-r--r-- 1 root root 8192633 2008-12-17 22:18 initrd.img-2.6.27.9  
[...]  
-rw-r--r-- 1 root root 892153 2008-12-18 14:25 System.map-2.6.27.9  
[...]  
-rw-r--r-- 1 root root 2244304 2008-12-18 14:22 vmlinuz-2.6.27.9
```

L'output è stato abbreviato. L'unica cosa che resta da fare è modificare il bootloader affinché permetta di avviare il sistema operativo col kernel appena compilato.

Modifica di Grub

Partirò dal presupposto che la vostra macchina utilizzi come bootloader grub, che è ormai presente in quasi tutte le distribuzioni. Nel caso usaste lilo, fate riferimento alla sua documentazione per le istruzioni.

Si apre con un editor di testo (sempre come amministratori) il file di configurazione:

```
nano /boot/grub/menu.lst
```

Si può scorrere con le frecce lungo le righe del file. Cercate la parte dove sono presenti le varie voci, qualcosa di simile a questo:

```
title Ubuntu 8.10, kernel 2.6.27-5-generic
root (hd0,0)
kernel /boot/vmlinuz-2.6.27-5-generic root=UUID=cbc39258-1c72-46b0-b398-3c6f81cc6574 ro quiet splash
initrd /boot/initrd.img-2.6.27-5-generic
quiet
```

```
title Ubuntu 8.10, kernel 2.6.27-5-generic (recovery mode)
root (hd0,0)
kernel /boot/vmlinuz-2.6.27-5-generic root=UUID=cbc39258-1c72-46b0-b398-3c6f81cc6574 ro single
initrd /boot/initrd.img-2.6.27-5-generic
```

Nell'esempio sono indicate due modalità di avvio, una normale multiutente, ed una monoutente da usare in caso di problemi. Il nostro scopo è copiare la modalità normale e modificarla per usare il kernel compilato. Pertanto, si può scrivere sotto alle due indicate, qualcosa del genere:

```
title Ubuntu 8.10, kernel 2.6.27.9 Compilato da me!
root (hd0,0)
kernel /boot/vmlinuz-2.6.27.9 root=UUID=cbc39258-1c72-46b0-b398-3c6f81cc6574 ro quiet splash
initrd /boot/initrd.img-2.6.27.9
quiet
```

Vi prego di prestare molta attenzione ai due parametri *root* (uno come riga di configurazione, e l'altro come argomento del parametro *kernel*). Essi indicano infatti la partizione che deve essere avviata. Non dovete certo copiare il mio esempio, invece assicuratevi di impostare in modo uguale a quanto avete già per le voci funzionanti!

Salvate ed uscite (rispettivamente con *ctrl + o* e *ctrl + x*). Ora non vi resta che riavviare e selezionare la voce relativa al vostro nuovo kernel, per verificare se davvero funziona ed avete fatto tutto bene. Divertitevi!

Fonti, spunti e approfondimenti

La maggior parte di questi url mi ha aiutato nello scrivere questa piccola guida. Alcuni sono invece semplicemente dei riferimenti utili per approfondire.

- Compilare un kernel linux <http://tinyurl.com/4tznmv>
- Ricompilare il Kernel <http://tinyurl.com/442zmd>
- Compilare il Kernel Linux 2.6.24 su Ubuntu, con il Thinkpad R61 <http://tinyurl.com/3fpz9p>
- Linux initial RAM disk (initrd) overview <http://tinyurl.com/28jcy2>
- System.map <http://tinyurl.com/4su9j5>

Per qualsiasi dubbio, chiarimento o informazione potete trovare una lista di tutti i modi per contattarmi all'indirizzo <http://andrealazzarotto.com/contatti>.