Saving Space mini-HOWTO

di Guido Gonzato, `mailto:guido@ibogeo.df.unibo.it` v1.0.1, 7 aprile 1999

Questo mini-HOWTO vuole dare dei consigli per far stare la tua installazione di Linux in uno spazio il più piccolo possibile. È rivolto specialmente agli utilizzatori di notebook. Traduzione di Mariani Dario, `darkpand@uni.net`

Contents

1 Introduzione

Ho un notebook su cui ho installato Linux, insieme a Windows 95 che era pre-installato. Ho ristretto la partizione di Windows a 500 Mb, facendo così spazio a una di tipo Linux da 240 Mb. Per quanto possa sembrare piccola, sono riuscito a installare un sistema Linux pressoché completo, basato sulla Red Hat 4.1 e su qualche trucco per liberare più spazio possibile.

Se hai bisogno di spazio, le indicazioni che troverai nella sezione seguente possono liberare una considerevole fetta del tuo hard disk. L'unico avvertimento che ti voglio dare è questo: non dare la colpa a me se qualcosa va storto! Utilizzeremo un paio di programmi che per me hanno funzionato bene, ma che sono pericolosi. Sei stato avvertito.

2 Requisiti di sistema

Hai bisogno di:

- Un sistema Linux completamente funzionante (qualunque versione dovrebbe andare bene);

- la comune utility di compressione `gzip`, o alternativamente `bzip2` che puoi trovare all'indirizzo `ftp://sunsite.unc.edu:/pub/Linux/utils/compress` ; questo strumento comprime meglio dello `gzip`, ma è anche un poco più lento e consuma più memoria;
- il compressore di eseguibili `upx`, le cui home page sono all'indirizzo `http://cdata.tvnet.hu/~ml/upx.html` e

  `http://wildsau.idv.uni-linz.ac.at/mfx/upx.html` ;
- il pacchetto `zlibc`, disponibile all'indirizzo `ftp://sunsite.unc.edu:/pub/Linux/libs/compression` . Si chiama `zlibc-X.X.tar.gz`, dove `X.X` è l'ultima versione.

Ci sono altri compressori di eseguibili. Uno è `gzexe` (dimenticalo), mentre `tcx` era probabilmente il migliore prima che `upx` fosse disponibile; ancora un altro è `tzx`, in teoria migliore di `tcx`. Il fatto è che ha rovinato tutto su un PC "di scorta", benché io non sia sicuro di aver fatto qualcosa di sbagliato. Ti conviene utilizzare `upx`: è sicuro, efficiente, e molto utile. Vai sul sicuro.

3 La procedura

3.1 Rimuovere i sorgenti del kernel

I sorgenti del kernel occupano più di 20 Mb, e potresti volerli rimuovere. Se è così, ti suggerisco di compilare un nuovo kernel adattato alla tua macchina una volta per tutte, e poi cancellare i sorgenti. Comunque, fai attenzione.

Non ti raccomando di rimuovere i sorgenti del kernel finché non sei sicuro che la tua macchina sia correttamente configurata. Inoltre, ti servono i #include del kernel per compilare i programmi in C. Pensaci due volte!

Se decidi di farlo, *non* rimuovere l'albero `include/linux` a meno che non sia sicuro che non compilerai mai applicazioni sulla tua macchina.

3.2 Togliere applicazioni

Ora, decidi di quali applicazioni hai *veramente* bisogno. Qualcuna può dimostrarsi ridondante: per esempio, sei sicuro di non poter vivere senza `emacs`? Puoi utilizzare `jed` al suo posto.

Sta a te decidere cosa vuoi tenere. Qualche consiglio generale:

- `gcc` è un pacchetto molto grande. È richiesto per ricompilare il kernel e tutte le applicazioni per cui non ci sono binari Linux pre-compilati. È necessario, ovviamente, se scrivi tuoi programmi in C o in Fortran (con `f2c` o `g77`); considera i tuoi bisogni prima di rimuoverlo. Altri compilatori come `lcc` sono buoni, ma non al livello di `gcc`. Ti suggerisco di tenerlo;
- X11 è terribilmente grande, ma è una bella cosa da avere. Se decidi di non potervi rinunciare, almeno cerca di farlo con il minimo indispensabile: solo l'X server giusto, un semplice window manager, solo un `xterm`, nessun font a 100 dpi, e così via;
- TeX e i pacchetti che lo accompagnano sono veramente molto grandi. Lasciar stare il LaTeX e utilizzare il TeX semplice fa risparmiare molto spazio; togliere i visualizzatori per X11 come `xdvi` e `ghostview` è possibile utilizzando `dvitty`, `dvivga` e simili. L'eliminazione del software dvi per X11 può rendere anche X11 ridondante;
- i giochi non sono mai "necessari".

3.3 Ridurre i binari

Cominciamo riducendo l'ampiezza dei binari. Spostati in `/usr/bin` e digita il comando

```
machine:/usr/bin# strip *
```

che toglierà i simboli contenuti nei binari. ripeti questo passo in `/usr/X11R6/bin/` e altre directory che potresti avere che contenegono eseguibili (non dimenticare di trovare i binari di TeX e `gcc`), ma *non* eseguirlo in `/sbin`, `/bin`, o `/usr/sbin/` se ti sta a cuore la tua installazione!

3.4 Comprimere con `upx`

Prima di tutto, installa `upx` e leggi la sua documentazione. Poi spostati in `/usr/bin` e avvialo con il comando `upx *`; comprimerà tutti gli eseguibili, includendo i suid (`tcx` non lo avrebbe fatto). Ripeti questo passo nelle altre directory come visto sopra.

Ricordati di comprimere gli eseguibili ogni volta che installi un nuovo pacchetto!

3.5 Gzippare

Ci sono molti altri file che possono essere compressi una volta e per sempre. Cominciamo con `/usr/doc/`; spostati in questa directory e digita il comando

```
machine:/usr/doc# find . -type f -exec gzip -9 {} \; 2> /dev/null
```

Ricorda di comprimere i documenti ogni volta che installi un nuovo pacchetto!

Ripeti questo passo nella directory contenente la documentazione per il TeX (sul mio sistema, `/usr/lib/texmf/texmf/doc/`). Se sei *veramente sicuro*, cancella del tutto queste directory.

Ora, installa `zlibc` e compilalo. Se il tuo sistema è come il mio, la compilazione fallirà lamentando la mancanza di una libc statica. Non ti preoccupare; troverai un file chiamato `uncompress.o` che è quello che ci serve. Spostalo in `/usr/local/lib/` e aggiungi questa linea al tuo `/etc/profile`:

```
export LD_ELF_PRELOAD=/usr/local/lib/uncompress.o
```

Ora, puoi comprimere con `gzip` non solo la documentazione, ma anche i file di dati: le applicazioni che li usano potranno utilizzarli ugualmente. In teoria, il trucco dovrebbe funzionare con la maggior parte delle applicazioni, ma in pratica il tuo grado di successo può variare. Il mio è stato piuttosto basso.

4 Un esempio dalla vita reale

Questo è il risultato ottenuto applicando i procedimenti di cui sopra ad una delle mie macchine. Prima del trattamento, `df` riportava 398.798 blocchi da 1024 byte utilizzati:

- non ho rimosso i sorgenti e gli header del kernel;
- ho disinstallato molte applicazioni e tutti i giochi, ma ho lasciato X11, sviluppo X11, sviluppo C e Fortran, Tcl/Tk, strumenti di rete, e poche altre applicazioni standard. `df` ha riportato 244.668 blocchi usati;

- ho eseguito `upx` in `/usr/bin`, `/usr/X11R6/bin`, `/usr/lib/texmf/bin/i586-linux`, e `/usr/lib/gcc-lib/i386-linux/2.7.2.1`: 226.270 blocchi usati;
- ho compresso la documentazione in `/usr/doc` e `/usr/lib/texmf/texmf/doc`: 198.745 blocchi usati.

Per riassumere, ho cominciato con 398.798 blocchi e ho finito con 198.745. Pensa alle cose che puoi mettere in questi 200.000 blocchi risparmiati! Avrei potuto risparmiarne ancora di più se avessi usato `bzip2` invece di `gzip`.

In media, se fai attenzione fin dall'inizio e installi solo le applicazioni necessarie, comprimere gli eseguibili e i documenti può farti liberare alcune decine di Mb. Su un notebook, ciò può essere veramente utile..

5 La fine

5.1 Copyright (in inglese)

5.2 Liberatoria

"Saving Space mini-HOWTO" è stato scritto da Guido Gonzato,

`REMOVE_MEguido@ibogeo.df.unibo.it` .

Questo documento è fornito "così com'è". Ho posto grandi sforzi nello scriverlo il più accuratamente che mi fosse possibile, ma utilizzi le informazioni contenute in esso a tuo rischio. In nessun caso posso essere responsabile per qualsiasi danno dovuto all'uso di questa opera.

I commenti sono i benvenuti. Per ogni richiesta, suggerimento, insulto, ecc., ritieniti libero di contattarmi.

Goditi Linux e la vita,

Guido `=8-)`