mini-HOWTO su come fare l'upgrade di una distribuzione Linux

Greg Louis, glouis@dynamicro.on.ca v1.11, 6 Giugno 1996

Suggerimenti sull'upgrade da una distribuzione Linux ad un'altra.

Contents

1 IMPORTANTE!!! Disclaimer e Copyright

Comments and questions may be directed to the author. Especially welcome, for use in future revisions, are accounts of successful upgrades of complex systems.

> *L'unica licenza valida è quella originale in lingua inglese. Di seguito ne trovate una traduzione abbastanza fedele che però non ha alcun valore.*

Potete rivolgere commenti e domande direttamente all'autore. Sono ben accette, per uso in versioni successive, relazioni su upgrade riusciti di sistemi complessi.

2 Cambiamenti dalla versione 1.1

- Aggiunta questa sezione
- Aggiunto il suggerimento di Zoltán Hidvégi su mtime e ctime. Grazie, Zoltán!
- Aggiunta la sezione Ringraziamenti

3 Introduzione

3.1 Come uccidere e resuscitare Linux

Lo scopo di questo documento è di dare dei suggerimenti per aiutarvi nella distruzione e riinstallazione di un sistema Linux. Non è affatto un libro di "ricette" sicuro, ma spero che servirà come indicazione su quello a cui si deve pensare, e sull'ordine in cui fare le cose. Mi sarebbe stato molto di aiuto se qualcun altro avesse scritto una cosa del genere prima del mio primo upgrade; quindi spero che sia di aiuto a voi, se dovete ricostruire una macchina Linux.

Non prendetelo come vangelo! Alcune cose cambieranno quasi sicuramente. Anche i nomi delle directory presenti nel documento possono essere diverse da quelle che dovete usare; alcuni hanno /usr/home invece di /home, ad esempio; altri la chiamano /u, ed altri (brivido :) mettono addirittura tutti gli utenti direttamente sotto /usr stessa! Non posso essere specifico per il vostro sistema, quindi ho semplicemente usato i nomi come sono nel mio.

Noterete anche che uso la distribuzione Slackware, e che assumo abbiate abbastanza RAM e spazio disco per installare i sorgenti del kernel Linux e compilare il kernel. Se il vostro sistema è diverso, alcune delle mie raccomandazioni non saranno applicabili, ma spero che potrete comunque trovare lo schema generale che vi possa essere di aiuto nel progetto di ricostruzione.

3.2 Perché farlo?

Buona domanda! Se si può evitare, non fatelo! (E questa è la raccomandazione più importante di tutta questa guida!!!) ...ma ci sono occasioni in cui non se ne può fare a meno.

Ad esempio, ho installato un hard disk da 4Gb e poi ho scoperto che Linux della Slackware 2.0 non sapeva che gli hard disk potessero essere più grandi di 2Gb, e si confondeva orribilmente; quindi ho dovuto fare un upgrade all'allora ultima Slackware 2.3. Quell'upgrade è stata un'esperienza estenuante, ed è parte della ragione per cui sto scrivendo queste note. Ho sbagliato praticamente tutto, e solo la fortuna ed il fatto che avessi un'altro Linux funzionante mi ha salvato dal disastro.

Un altro esempio: ho scoperto che non potevo compilare un kernel linux a.out funzionante con la serie 1.3, usando un'installazione esterna della Slackware 2.3 (un'altra macchina, non quella con cui ho impasticciato). Ho preso la palla al balzo, ho comprato la Slackware 3.0 su CDROM e ho convertito tutto a ELF. Stavolta la reinstallazione è andata meglio, grazie in parte all'esperienza precedente, ed è servita come spunto per la maggior parte delle idee che vi sto offrendo qui.

3.3 Devo per forza "distruggere e reinstallare"?

Stranamente è più sicuro. Se installate sopra un sistema Linux esistente, è probabile che abbiate una miscela di binari vecchi e nuovi, e in genere un vero problema per l'amministrazione. Ripulire il sistema del tutto, e poi rimettere solo quello di cui sapete avere bisogno è un modo drastico ma efficace di avere un risultato pulito. (Naturalmente stiamo parlando di installare una distribuzione per intero, non di aggiornare uno o due pacchetti! Il modo migliore per evitare una reinstallazione completa è, precisamente, di tenere aggiornati i singoli componenti – specialmente gcc e le sue librerie, e le binutils. Se quello che usate è ragionevolmente aggiornato, e potete tenercelo prendendo, e se necessario compilando, del nuovo codice di tanto in tanto, non c'è bisogno di un aggiornamento di massa).

Come nota Patrick Volkerding (anche lui raccomanda di togliere tutto per fare gli upgrade), installare ELF sopra un sistema funzionante a.out è il modo sicuro per provocare un disastro: almeno, se ne sapete abbastanza per provarci, non avete bisogno di questa guida!

Anche senza questa complicazione, però, sarebbe meglio ricostruire tutto da zero.

3.4 Quanto tempo ci vuole?

Naturalmente dipende dalla complessità del sistema, ma posso dire che, per l'aggiornamento andato bene (l'altro? – non chiedetemelo! :) ci ho messo circa dieci ore per i backup, sei ore per ricostruire il sistema per poter riabilitare i login, e un'altra mezza giornata più o meno per rimettere a posto le cose meno cruciali. Col passar del tempo mi accorgo che ci sono delle piccole cose che non sono come le vorrei – le aggiusto mano a mano che le trovo – ma in tutto dovrebbero bastare venti ore di lavoro per un progetto ragionevolmente complesso. Forse meno se state reinstallando da hard disk (io ho usato il CDROM) o più se dovete installare da floppy. Forse meno se avete un Pentium veloce, più se avete un 386. Avete capito il concetto.

E questo per quanto riguarda l'installazione. Ecco come impostarla, una volta deciso che deve essere fatta. Armatevi di forza d'animo, di caffeina o cose del genere, e:

4 Scrivete tutto quello che fate.

E' estremamente prezioso avere un registro di quello che avete fatto nel processo di preparazione e di applicazione delle modifiche. E' specialmente importante una lista dei backup che farete in preparazione della distruzione del sistema esistente.

5 Fate un backup completo del sistema esistente.

Generalmente parlando, i backup tendono a venir scritti su mezzi ad accesso sequenziale. Posto questo, non vorrete usare il backup completo per reinstallare un numero significativo di file: ne ha troppi che non volete. E' meglio creare dei piccoli backup di segmenti individuali che sapete che dovrete reistallare del tutto. Farò degli esempi più avanti.

Perché cominciare con un backup completo? Ci sono due ragioni principali: per prima cosa, nell'eventualità di una catastrofe nell'installazione del nuovo sistema, avrete modo di tornare indietro al punto di inizio con un minimo sforzo. Secondo, non importa quanto prepariate bene la nuova installazione, c'è una probabilità molto alta che uno o due file importanti vengano dimenticati; in questo caso la scomodità di dover estrarre questi file dal backup completo sarà preferibile a farne a meno.

Per risparmiare tempo e spazio, se avete ancora il mezzo di distribuzione per la vecchia versione Linux, potete salvare solo i file che hanno l'mtime o il ctime più recente della data dell'installazione originale.

6 Fare il backup di /etc e delle sue sottodirectory su uno o più floppy.

Questo è l'estremo opposto: non reinstallerete questi file (per la maggior parte, almeno), ma li paragonerete con quelli nuovi che vengono creati durante l'installazione. Perché? Perché i file nuovi potrebbero contenere informazioni che i vecchi non hanno, o potrebbero esprimere le stesse cose in modo diverso. I cambiamenti nei protocolli, l'aggiunta di nuovi strumenti o l'implementazione di nuove caratteristiche in strumenti esistenti sono tutte cose che possono apportare dei cambiamenti nel formato dei file di configurazione e negli script di avvio contenuti nell'albero /etc, e molto probabilmente dovrete immettere i vostri vecchi dati nei nuovi file per mantenere i formati nuovi e sfruttare i miglioramenti.

7 Fate backup separati di ogni gruppo di file che volete conservare.

Questa è la parte più variabile del lavoro, e tutto quello che posso fare è descrivere quello che ho fatto io nel mio sistema, nella speranza che servirà come guida. Sostanzialmente, dovete dare un'occhiata a tutte le directory che contengono:

- file che non fanno parte della vostra installazione linux standard, o
- file più recenti di quelli che installerete con la nuova installazione

e separate solo quei file che volete mantenere.

(Un'altra strategia possibile è di fare il backup di tutti i file con mtime o ctime più recenti del giorno della precedente installazione di Linux, come detto sopra, e recuperarli da lì. Se fate così, dovrete tener conto che la nuova distribuzione di Linux potrebbe contenere dei file ancora più recenti di quelli che avete salvato.)

Nel mio caso, ho finito per fare un file .tgz sul mezzo di backup per ciascuna di queste directory:

- /usr/lib/rn
- /usr/lib/smail
- /usr/lib/trn (il resto di /usr/lib era da reinstallare)
- /usr/local/src
- /usr/local/bin
- /usr/local/lib
- /usr/local/lpfont
- /usr/local/man
- /usr/local/sbin
- /usr/local/thot (c'erano altri file in /usr/local che non mi servivano)
- /usr/openwin
- /usr/src/lilo-17 (perché la mia Slackware nuova aveva ancora la versione 16)
- /usr/src/linux-1.2.13 (perché l'avevo configurato)
- /usr/X11R6/lib/X11/app-defaults

- /usr/X11R6/lib/X11/initrc (il resto di Xfree86 era da reinstallare)
- /var/named
- /var/openwin
- /var/texfonts

La mia macchina era relativamente semplice, dato che non c'erano file di spool di cui preoccuparsi. Non ho uno spool di news in quel sistema, e dato che ci sono solo due utenti è più semplice leggere tutta la posta prima di fare lo shutdown. Negli altri casi bisogna fare il backup delle directory /var/spool all'ultimo momento, come naturalmente le directory delle librerie e dei siti delle news!

8 Preparate i dischi di root e boot per la nuova installazione.

I dettagli di come farlo si possono trovare nella guida di installazione della distribuzione che usate.

9 Formattate i dischetti per il kernel temporaneo e per la compilazione finale.

Ve ne serviranno due, un dischetto per ciascuno.

Fatto tutto questo, siete pronti per l'Ora X. Il passo successivo rende il sistema non disponibile.

10 Inibite i login e fate il backup delle directory /root e /home.

E' l'ultima cosa da fare sui vecchi sistemi prima di distruggerli, in modo da poter avere i dati di root e degli utenti il più aggiornati possibile.

11 Avviate il sistema dai dischetti di boot e root.

12 Cancellate le partizioni Linux con fdisk e ricreatele.

La guida di installazione vi spiegherà come farlo; così distruggerete il sistema. Da ora in poi dipendete dalla qualità dei backup che avete fatto nei passi precedenti! Siete stati avvisati!

13 Installate la nuova versione di Linux.

Ci sono già diversi buoni documenti che descrivono come portare avanti questo passo, quindi non scenderò nel dettaglio. Continuate da qui quando il nuovo sistema può essere avviato dal'hard disk.

Nel frattempo, assicuratevi di creare un dischetto da cui possiate avviare il sistema, dato che il kernel installato di default da Linux deve essere sostituito, e nel farlo possono accadere degli incidenti. Assicuratevi di installare i pacchetti di sviluppo e i sorgenti del kernel.

14 Con il nuovo sistema linux avviato dall'hard disk, modificate /etc/fstab

ed aggiungete la partizione di swap. Poi date il comando "swapon -a". Non so perché, ma il programma di installazione della Slackware non si offre di farlo se la partizione di swap già esiste. Poi, quando avviate il sistema nuovo e lo script rc.S prova ad abilitare lo swap, non trova la partizione nel file fstab e lo swap non funziona. Questo procedimento risolve il problema.

15 Recuperate le configurazioni nella directory /etc e nelle sue sottodirectory.

Coem descritto sopra, non potete semplicemente copiare i vecchi file in /etc ed aspettarvi che funzioni tutto; con alcuni file si può fare, ad esempio /etc/XF86Config (sempre che stiate usando la stessa versione di Xfre86 – e lo stesso hardware video – dell'installazione precedente). Per la maggior parte, però, è meglio usare diff per paragonare i file vecchi ai nuovi prima di copiarli. Controllate specialmente dei cambiamenti significativi nei file in /etc/rc.d, dove potreste dover modificare tutto a mano, invece di copiare i vecchi script dal backup. Una volta finito, riavviate il sistema.

16 Configurate e ricompilate il kernel di Linux.

Anche se questo passo non è necessario per avere un kernel che supporta il vostro hardware, vale la pena farlo per evitare di avere nel kernel i driver di tutta quella roba che sulla macchina non c'è. Per dettagli su questo, leggete il Kernel HOWTO. Installate prima il kernel ricompilato su un dischetto: quando vedete che da lì l'avvio è a posto, installatelo sull'hard disk, avviate lilo se lo state usando, e riavviate il computer.

17 Recuperate quello che avevate salvato.

Potrebbe essere necessario reinstallare alcuni dei binari dai sorgenti: io ho dovuto farlo con lilo, ad esempio, dato che la mia versione era più recente di quella nell'installazione Slackware, e non avevo fatto il backup del file binario da /sbin. Dovreste controllare tutti i programmi che recuperate dai backup, e confermare l'esistenza e la correttezza dei file di configurazione, delle librerie e così via. In alcuni casi potreste dover recuperare i programmi in un ordine specifico: avete annotato delle cose durante il backup, vero? ;-)

18 Controllate la sicurezza.

Controllate i permessi dei file e delle directory in modo da essere sicuri che l'accesso non è né troppo ristretto né troppo facile. Io trovo che la Slackware tende ad essere un ambiente un po' aperto per i miei gusti, quindi vado in giro a cambiare i 755 in 711 per i binary nelle directory ../bin e cose del genere. O addirittura i 700 in quelli ../sbin. Ci vuole un'attenzione particolare se state facendo girare un server ftp, ma probabilmente ci avete già pensato ;-).

19 Abilitate i login.

E' tutto a posto. Nel prossimo periodo, ci saranno probabilmente dei dettagli da chiarire, ma il grosso del lavoro è fatto. Buon divertimento!

20 Mi dispiace, ma ancora una volta:

USATE QUESTE INFORMAZIONI A VOSTRO RISCHIO E PERICOLO!

(Vedere il disclaimer all'inizio di questo documento.)

21 Ringraziamenti

Grazie a Zoltán Hidvégi per aver contribuito al contenuto di questo mini-HOWTO.