Linux Partition HOWTO

Kristan Koehntopp, kris@koehntopp.de v2.4, 3 November 1997

Questo Mini-HOWTO spiega come pianificare e disporre lo spazio disco per il vostro sistema Linux. Parla di hardware, partizioni, dimensione dello spazio di swap e considerazioni di posizionamento, file system, tipi di file system e argomenti collegati. L'intenzione è quella di dare un po' di conoscenza generale, non delle procedure. Traduzione di Andrea Manzini <linux@netbusiness.it>

Indice

1 Introduzione

1.1 Di che cosa si tratta?

Questo è un testo Linux Mini-HOWTO. Un Mini-HOWTO è un piccolo testo che spiega qualcosa collegato all'installazione di Linux e alla sua manutenzione in forma esplicativa. È mini, perché o il testo o l'argomento che discute sono troppo brevi per un vero HOWTO o addirittura un libro. Un HOWTO non è una guida di riferimento: per questo esistono le pagine man.

1.2 Che cosa contiene? Altri documenti HOWTO collegati

Questo particolare Mini-HOWTO spiega come pianificare e suddividere lo spazio disco per il vostro sistema Linux. Parla di hardware, partizioni, dimensione dello spazio di swap e considerazioni di posizionamento, file system, tipi di file system e argomenti collegati. L'intenzione è quella di dare un po' di conoscenza generale, così in questo testo parleremo soprattutto di principi e non di strumenti.

In teoria, questo documento dovrebbe essere letto prima dell'installazione, ma questo risulta difficile per molte persone; i principianti hanno ben altri problemi che non l'ottimizzazione del disco. Così voi siete probabilmente tra quelli che hanno appena finito una installazione Linux e adesso starete pensando a come ottimizzare questa installazione o come evitare certe sviste nella prossima. Bene, preparatevi a cancellare e a rifare la vostra installazione quando avrete finito di leggere questo testo. :-)

Questo Mini-HOWTO si limita alla pianificazione e alla disposizione dello spazio su disco nella maggior parte dei casi. Non discute dell'uso di fdisk, LILO, mke2fs o programmi di backup. Ci sono altri HOWTO che trattano questi problemi. Fate riferimento all'indice degli HOWTO per informazioni sugli HOWTO disponibili. Nell'indice ci sono anche le informazioni su come e dove procurarsi i documenti HOWTO.

Per imparare a stimare le diverse dimensioni e velocità richieste dalle differenti parti del file system, vedere il mini-HOWTO "Linux Multiple Disks Layout", di Gjoen Stein <gjoen@nyx.net>.

Per istruzioni e considerazioni su dischi con più di 1024 cilindri, vedere il mini-HOWTO "Linux Large Disk", di Andries Brouwer <aeb@cwi.nl>.

Per istruzioni sulla limitazione dell'uso di spazio su disco per utente (quotas), vedere il mini-HOWTO "Linux Quota", di Albert M.C. Tam <bertie@scn.org>

Al momento, non ci sono documenti generici sul backup, ma ci sono vari documenti con riferimenti a soluzioni specifiche di backup. Vedere il mini-HOWTO "Linux ADSM Backup", di Thomas Koenig <Thomas.Koenig@ciw.uni-karlsruhe.de> per istruzioni su come integrare Linux in un ambiente di backup IBM ADSM. Vedere il mini-HOWTO "Linux Backup with MSDOS", di Christopher Neufeld <neufeld@physics.utoronto.ca> per informazioni circa il backup di dischi Linux via MS-DOS.

Per istruzioni sulla scrittura e l'invio di un documento HOWTO, vedere l'indice "The Linux HOWTO Index", di Tim Bynum <linux-howto@sunsite.unc.edu>.

Anche scorrere nella directory /usr/src/linux/Documentation può essere veramente istruttivo. Leggete i file ide.txt e scsi.txt per una informazione generale sulle caratteristiche dei driver e date un'occhiata alla sottodirectory filesystems/.

2 Che cos'è una partizione?

Poco dopo che dischi fissi furono inventati, la gente volle installare più di un sistema operativo, anche se disponevano di un solo disco. Nacque così la necessità di un meccanismo che dividesse un singolo disco fisico in tante unità logiche. Quindi ecco cos'è una partizione: un insieme di blocchi contigui sul vostro hard disk che viene trattato dal sistema operativo come se fosse un disco completamente autonomo e separato.

È abbastanza evidente che le partizioni non devono sovrapporsi: un sistema operativo non sarebbe felice se un altro sistema operativo installato sulla stessa macchina gli sovrascrivesse informazioni importanti a causa della sovrapposizione delle due partizioni. Non ci dovrebbe neanche essere dello spazio vuoto tra una partizione e un'altra, in quanto anche se non comporterebbe alcun danno, significherebbe sprecare prezioso spazio su disco lasciandolo inutilizzato.

Un disco non ha bisogno di essere completamente partizionato. Potete decidere di lasciare un po' di spazio alla fine senza assegnarlo ad alcun sistema operativo e, successivamente, quando sarà chiaro quale userete per la maggior parte del tempo, potrete partizionare questo spazio vuoto rimanente e metterci sopra un file system.

Le partizioni non possono essere spostate né ridimensionate senza distruggere il file system che contengono. Il ripartizionamento quindi comporta il backup e il ripristino di tutti i file system coinvolti nel ripartizionamento. In effetti, è molto facile fare dei pasticci durante il ripartizionamento, così dovreste in ogni caso fare una copia di backup di tutto prima ancora di usare cose come `fdisk`.

Beh, veramente alcuni tipi di partizione che contengono alcuni tipi di file system *possono* essere divisi in due senza alcuna perdita di dati (se siete fortunati). Per esempio, esiste un programma chiamato "fips" per dividere le partizioni MS-DOS in due per far spazio a una partizione Linux senza dover reinstallare MS-DOS. In ogni caso, non andreste a maneggiare queste cose senza aver prima fatto un accurato e completo backup, vero?

2.1 I backup sono importanti

I nastri sono molto utili per fare i backup. Sono veloci, affidabili e facili da usare, quindi potrete effettuare il backup spesso, preferibilmente in modo automatico e senza disagi.

Parlando di cassette: quello di cui sto parlando sono i veri nastri, non quelle porcherie pilotate dal controller dei dischetti. Tenete in considerazione SCSI: Linux lo supporta in modo nativo. Non occorre caricare driver ASPI, non sprecate preziosa HMA e una volta che l'adattatore SCSI è installato, potete attaccarci anche dischi, nastri, scanner e CD-ROM. Niente più indirizzi I/O, acrobazie con IRQ né master/slave e livelli PIO da abbinare.

In aggiunta: un buon adattatore SCSI fornisce alte prestazioni di I/O senza occupare troppo la CPU. Anche durante una pesante attività di disco, avrete tempi di risposta molto buoni. Se intendete usare un sistema Linux come un server USENET o se volete entrare nel mercato degli ISP, non pensate nemmeno di impostare un sistema senza SCSI.

2.2 Numeri e nomi di device

Il numero di partizioni di un sistema basato su Intel è stato limitato fin dall'inizio: la tabella delle partizioni era installata come parte del settore di boot e aveva spazio solo per 4 partizioni. Queste partizioni adesso sono chiamate partizioni primarie. Quando divenne chiaro che la gente voleva più partizioni nei loro sistemi, sono state inventate le partizioni logiche. Il loro numero non è limitato: ogni partizione logica contiene un puntatore alla prossima, così da avere una catena potenzialmente illimitata di partizioni.

Per ragioni di compatibilità, lo spazio occupato da tutte le partizioni logiche deve essere tenuto in conto. Se state usando partizioni logiche, una partizione primaria viene marcata come "partizione estesa" e il suo blocco di inizio e di fine segnano l'area occupata dalle partizioni logiche in essa contenute. Questo implica che lo spazio assegnato a tutte le partizioni logiche deve essere contiguo. Ci può essere solo una partizione estesa per disco: nessun programma `fdisk` può creare più di una partizione estesa.

Linux non può gestire più di un limitato numero di partizioni per disco. Così in Linux potrete avere 4 partizioni primarie (3 delle quali usabili, se state usando partizioni logiche) e al massimo 15 partizioni totali su un disco SCSI (63 in tutto su un disco IDE)

In Linux, le partizioni sono rappresentate come file di device. Un file di device è un file con tipo c (che sta per "carattere", cioè device che non usano un buffer di cache) oppure b (ossia device a "blocchi", che passano dal buffer di cache). In Linux, tutti i dischi sono rappresentati solo come device a blocchi. A differenza di altri unix, Linux non offre "pure" versioni a caratteri dei dischi e delle loro partizioni.

La sola cosa importante con un file di device sono il suo numero primario e secondario, visualizzato invece delle dimensioni del file:

```
$ ls -l /dev/hda
brw-rw----   1 root     disk       3,    0 Jul 18  1994 /dev/hda
                                   ^     ^
                                   |     numero secondario di device
                                   numero primario di device
```

Quando un file di device viene aperto, il numero primario indica quale driver viene usato per effettuare l'operazione di input/output. Alla chiamata viene passato il numero secondario come parametro e spetta al driver l'interpretazione del numero secondario. Di solito la documentazione del driver riporta come viene usato il numero secondario. Per i dischi IDE, questa documentazione è in `/usr/src/linux/Documentation/ide.txt`. Per i dischi SCSI, ci si aspetterebbe questa documentazione in `/usr/src/linux/Documentation/scsi.txt`, ma non c'è. Per accertarsene, occorrerebbe leggere il sorgente del driver, (`/usr/src/linux/driver/scsi/sd.c:184-196`). Fortunatamente, c'è la lista dei numeri e dei nomi di device curata da Peter Anvin in `/usr/src/linux/Documentation/devices.txt`; vedere le voci per i device a blocchi, primario 3, 22, 33, 34 per i dischi IDE e primario 8 per gli SCSI. Il numero primario e secondario sono un byte ognuno ed è per questo che il numero di partizioni per disco è limitato.

Per convenzione, i file di device hanno determinati nomi e molti programmi di sistema riconoscono questi nomi essendo stati compilati nel loro codice. Si aspettano che i vostri dischi IDE si chiamino `/dev/hd*` e i dischi SCSI `/dev/sd*`. I dischi sono numerati come a, b, c e così via, quindi `/dev/hda` è il primo disco IDE e `/dev/sda` il primo disco SCSI. Entrambi rappresentano interi dischi, iniziando dal primo blocco. Scrivere su questi device con il programma sbagliato distruggerà il master boot loader e la tabella delle partizioni dei dischi, rendendo così tutti i dati inutilizzabili o facendo sì che il sistema non possa più eseguire il boot. Siate sicuri di quello che fate e, ripeto, copiate su backup prima di farlo.

Le partizioni primarie su un disco sono numerate 1, 2, 3 e 4. Così `/dev/hda1` è la prima partizione primaria del primo disco IDE eccetera. Le partizioni logiche hanno numeri che vanno da 5 in su, così `/dev/sdb5` è la prima partizione logica del secondo disco SCSI.

Ogni voce nella tabella delle partizioni è contrassegnata da un blocco di inizio, uno di fine, e un tipo. Il tipo è un codice numerico (a un byte) che indica l'appartenenza di quella partizione a un determinato sistema operativo. Per la gioia dei consulenti informatici i codici dei tipi di partizione non sono del tutto univoci, cioè esiste la probabilità che due sistemi operativi usino lo stesso codice.

Linux riserva per le sue partizioni di swap il codice 0x82, e lo 0x83 per il filesystem "nativo" (che è ext2 per quasi tutti). L'oramai sorpassato e una volta famoso Linux/Minix filesystem usava il codice 0x81. OS/2 segna le sue partizioni con il tipo 0x07, lo stesso del NTFS di Windows NT. MS-DOS usa diversi codici per i vari tipi di filesytem FAT: 0x01, 0x04 e 0x06. DR-DOS usava il codice 0x81 per indicare partizioni FAT protette, creando una ambiguità con Linux/Minix, ma ormai né Linux/Minix, né DR-DOS vengono largamente usati. Per finire, le partizioni estese che fungono da contenitori per le partizioni logiche hanno codice 0x05.

Le partizioni vengono create e cancellate con il programma `fdisk`. Ogni sistema operativo che si rispetti viene fornito con una versione di `fdisk` e per tradizione si chiama proprio `fdisk` (oppure `FDISK.EXE`) in quasi tutti i sistemi operativi. Qualcuno di questi `fdisk`, come ad esempio quello del DOS, mostra qualche limitazione quando ha a che fare con partizioni di altri sistemi operativi. Queste limitazioni includono l'impossibilità di gestire altri tipi di codice, l'impossibilità di trattare numeri cilindro superiori a 1024, e l'impossibilità di creare o gestire partizioni che non terminano esattamente su un cilindro. Per esempio, fdisk di MS-DOS non può cancellare le partizioni NTFS, quello di OS/2 è conosciuto per "correggere" senza preavviso le partizioni create con fdisk di Linux che non terminano esattamente sul cilindro e entrambi, sia quello DOS sia quello OS/2, hanno problemi con i dischi che risultano avere più di 1024 cilindri (vedere il Mini-HOWTO "Large disk" per i dettagli)

3 Di quali partizioni ho bisogno?

3.1 Di quante partizioni ho bisogno?

Bene, allora quali partizioni vi servono? Tanto per cominciare alcuni sistemi operativi non riescono a fare il boot dalle partizioni logiche, per motivi che esulano dalla comprensione di un sano di mente. Così probabilmente vorrete riservare le partizioni primarie come partizioni di boot per MS-DOS, OS/2 e Linux o cos'altro state usando. Ricordate anche che una partizione estesa non è altro che una partizione primaria, che funziona da contenitore per il resto del disco suddiviso in partizioni logiche.

L'avvio (boot) di un sistema operativo è una operazione che si svolge in modalità reale, coinvolgendo così il BIOS e la limitazione dei 1024 cilindri. Probabilmente quindi piazzerete tutte le vostre partizioni bootabili nei primi 1024 cilindri del disco, giusto per evitare problemi. Ripeto, leggete il Mini-HOWTO "Large disk" per i dettagli salienti.

Per installare Linux, avrete bisogno di almeno una partizione. Se il kernel viene caricato da questa partizione (per esempio da LILO), essa deve essere leggibile dal BIOS. Se state usando altri sistemi per caricare il kernel (da un disco di boot o da MS-DOS con il programma LOADLIN.EXE) allora la partizione può essere ovunque. In ogni caso questa partizione avrà il tipo 0x83 "Linux nativa".

Il vostro sistema avrà bisogno di spazio di swap. A meno che vogliate paginare su file, avrete bisogno di una partizione di swap dedicata. Dal momento che solo il kernel di Linux accede a questa partizione, e il kernel non ha limitazioni legate al BIOS, la partizione di swap può essere posizionata ovunque. Io consiglio di usare una partizione logica (/dev/?d?5 e superiori). Le partizioni dedicate allo swap hanno codice tipo 0x82 "Linux Swap".

Questi sono i requisiti minimi. Può essere utile creare altre partizioni per Linux. Continuate a leggere.

3.2 Quanto deve essere grande lo spazio di swap?

Se avete deciso di usare una partizione di swap dedicata, il che è generalmente una Buona Idea [tm], seguite queste indicazioni per stimare la sua dimensione:

- In Linux la RAM e lo spazio di swap si sommano (questo anche in tutti gli altri Unix). Per esempio, se avete 8 MB di RAM e 12 MB di swap, in totale avete 20 MB di memoria virtuale.
- Quando dimensionate lo spazio di swap, tenete conto che dovreste avere almeno 16 MB di memoria virtuale totale. Così con 4 MB di RAM considerate un minimo di 12 MB di swap, per 8 MB almeno altri 8 di swap.
- In Linux, una partizione di swap non può essere più grande di 128 MB. Cioè, in effetti la partizione può anche essere più grande, ma lo spazio in più non viene utilizzato. Se volete più di 128 MB di swap, dovete creare più di una partizione di swap.
- Quando dimensionate lo spazio di swap, ricordate che troppo swap può essere del tutto inutile.

  Ogni processo ha uno "spazio di lavoro". È una serie di pagine in memoria a cui il processore accederà nell'immediato futuro. Linux tenta di anticipare questi accessi in memoria (assumendo che le pagine usate più di recente saranno usate ancora prossimamente) e cerca di tenere queste pagine in RAM se possibile. Se il programma ha una buona "località", questa assunzione risulterà vera e l'algoritmo di predizione farà il suo lavoro.

  Mantenere uno spazio di lavoro in memoria è possibile solo se c'è abbastanza memoria. Se avete troppi processi attivi, il kernel è obbligato a spostare su disco alcune pagine che verranno ricaricate dopo poco tempo (forzando lo swap di alcune pagine di altri processi). Di solito questo comporta un pesante incremento nell'attività di paginazione e una sostanziale perdita di prestazioni. Una macchina in questo stato va in "trashing", oppure si dice che si "siede", o "swappa" (per i lettori di lingua tedesca, "dreschen", "schlagen", "haemmern" e non "muellen").

Su una macchina "seduta" i processi girano sostanzialmente da disco e non in RAM. Le prestazioni si abbasseranno nella misura approssimativa del rapporto che c'è tra il tempo di accesso alla memoria e il tempo di accesso del disco.

Una vecchia regola di massima ai tempi del PDP e del Vax era che la grandezza dello spazio di lavoro di un programma è circa il 25% della sua dimensione. Pertanto probabilmente è inutile avere uno spazio di swap superiore al triplo della RAM.

Tenete a mente che questa è solo una regola di massima. È facile incontrare casi dove i programmi hanno bisogno di uno spazio di lavoro estremamente ampio o molto piccolo. Per esempio, un programma di simulazione che accede a un grosso numero di dati in maniera pseudocasuale, ha pochissima località nel suo spazio di dati, pertanto lo spazio di lavoro sarà molto ampio.

D'altra parte, un xv con tanti JPEG aperti contemporaneamente, tutti iconizzati tranne uno, avrebbe un segmento dati molto grande. Ma le trasformazioni vengono eseguite su una singola immagine, infatti la maggior parte della memoria occupata da xv non viene mai toccata. Lo stesso per un editor con tante finestre aperte, ma solo una di loro viene modificata alla volta. Questi programmi hanno, se sono progettati correttamente, una località molto alta e la maggior parte del loro spazio di lavoro può essere paginata su disco senza un grosso impatto sulle prestazioni.

Si potrebbe sospettare che quel 25%, derivato dall'era dei programmi a linea di comando non sia più valido per i programmi GUI moderni che agiscono su documenti multipli, ma non conosco nessun documento più recente che tenti di verificare questo indice.

Così per una configurazione con 16 MB di RAM, non serve swap per una configurazione minima e più di 48 MB di swap sono probabilmente inutili. La quantità esatta dipende dalle applicazioni che fate girare sulla macchina (cosa vi aspettavate?).

3.3 Dove dovrei mettere lo spazio di swap?

- La meccanica è lenta, l'elettronica veloce.

  I dischi fissi moderni hanno molte testine. Il passaggio tra testine della stessa traccia è rapido, poiché è elettronico. Passare da traccia a traccia è lento, perché significa far muovere fisicamente qualcosa.

  Così se avete un disco con tante testine e uno con meno e per entrambi gli altri parametri sono identici, il disco con più testine sarà più veloce.

  Separare lo swap in due e metterlo su entrambi i dischi sarebbe ancora meglio, penso.

- I vecchi dischi hanno lo stesso numero di settori su tutte le tracce. Con questi dischi sarebbe meglio mettere lo swap nel mezzo del disco, ottimizzando il caso che la testina si muova da una traccia casuale verso l'area di swap.

- I dischi più recenti usano lo ZBR (zone bit recording). Hanno più settori sulle tracce esterne. Con un numero costante di giri al minuto, questo comporta migliori prestazioni sulle tracce esterne che su quelle interne. Meglio mettere lo swap sulle tracce più veloci.

- Naturalmente la testina non si muove in maniera casuale. Se avete lo spazio di swap nel mezzo di un disco tra una partizione "home" molto occupata e una partizione di archivio usata molto poco, sarà meglio avere lo swap nel mezzo della partizione home per accorciare i movimenti della testina. Sarebbe ancora meglio spostare lo swap su un altro disco altrimenti inutilizzato.

In breve: Mettete lo swap sul disco più veloce con più testine che non sia occupato per dell'altro. Se avete più di un disco: dividete lo swap e spezzettatelo fra tutti i dischi e/o fra più controller.

Ancora meglio: Comprate altra RAM.

3.4 Alcune note sui file system e la frammentazione

Lo spazio su disco è gestito dal sistema operativo in blocchi e frammenti di blocchi. In ext2, i frammenti e i blocchi devono avere la stessa dimensione, quindi possiamo restringere la nostra discussione ai blocchi.

I file possono essere di qualsiasi dimensione, perciò possono non terminare sul margine di un blocco. Così per ogni file una parte dell'ultimo blocco viene sprecata. Assumendo che la dimensione dei file sia casuale, c'è approssimativamente mezzo blocco sprecato per ogni file sul vostro disco. Tanenbaum nel suo libro "Operating Systems" chiama questo evento "frammentazione interna".

Potete intuire il numero di file sul disco dal numero di i-node allocati. Sul mio disco

```
# df -i
Filesystem           Inodes   IUsed   IFree  %IUsed Mounted on
/dev/hda3             64256   12234   52022    19%  /
/dev/hda5             96000   43058   52942    45%  /var
```

Ci sono circa 12000 file su `/` e circa 44000 file su `/var`. Con dei blocchi grandi 1 KB, circa 6 + 22 = 28 MB di spazio su disco sono sprecati nei blocchi di coda dei file. Se avessi scelto una dimensione del blocco di 4 KB, avrei sprecato uno spazio 4 volte superiore.

Tuttavia il trasferimento dati è più veloce per blocchi di dati grossi e contigui. È per questo che ext2 prova a riservare spazio in unità di 8 blocchi contigui per i file in crescita. Lo spazio riservato e non utilizzato viene rilasciato quando il file viene chiuso, pertanto non c'è spreco di spazio.

Sistemare blocchi in modo non consecutivo è un male per le prestazioni, poiché l'accesso ai file avviene in maniera sequenziale. Il sistema operativo viene forzato a ripetere l'accesso al disco, e il disco deve spostare la testina. Questo viene chiamato "frammentazione esterna" o semplicemente "frammentazione" ed è un problema comune con i filesystem DOS.

Ext2 ha diverse strategie per evitare la frammentazione esterna. Normalmente la frammentazione non è un grosso problema per ext2, nemmeno sulle partizioni usate pesantemente come uno spool di news USENET. Anche se c'è un programma di deframmentazione per il filesystem ext2, nessuno lo usa e non è aggiornato con la versione corrente di ext2. Usatelo se volete, ma fatelo a vostro rischio e pericolo.

Il filesystem MS-DOS è ben conosciuto per la sua patologica gestione dello spazio disco. Insieme con l'irrisiorio buffer di cache usato da MS-DOS, gli effetti della frammentazione sulle prestazioni sono molto rilevanti. Gli utenti DOS sono abituati a deframmentare i dischi ogni poche settimane e qualcuno ha addirittura sviluppato alcune credenze rituali a riguardo della frammentazione. Nessuna di queste abitudini dovrebbe essere trasportata su Linux ed ext2. I file system nativi di Linux non hanno bisogno di deframmentazione durante il normale utilizzo, cioè ogni situazione con almeno il 5% di spazio libero su disco.

Il file system MS-DOS è famoso anche per sprecare grosse porzioni di disco a causa della frammentazione interna. Per partizioni più grandi di 256 MB, la dimensione dei blocchi diventa tale che non sono più efficaci (questo è stato corretto in qualche maniera con FAT32)

ext2 non vi costringe a scegliere grossi blocchi per grossi file system, eccetto per file sistem enormi dagli 0.5 TB (1 TB equivale a 1024 GB) in su, quando i blocchi piccoli diventano meno efficienti. Quindi a differenza del DOS non c'è bisogno di dividere grossi dischi in molte partizioni per mantenere piccola la dimensione del blocco. Usate la dimensione di default di 1 KB se possibile. Potete sperimentare con una dimensione di 2 KB per alcune partizioni, ma aspettatevi di incontrare qualche raro bug: la maggior parte della gente usa il default.

3.5 Il ciclo di vita dei file e le sequenze di backup come criterio di partizionamento

Con ext2, le decisioni di partizionamento dovrebbero essere governate da considerazioni di backup e dalla diversificazione dei tempi di vita dei differenti tipi di file per evitare la frammentazione esterna.

I file hanno dei tempi di vita. Dopo che un file è stato creato, rimarrà per un certo tempo nel sistema e poi verrà rimosso. I cicli di vita dei file variano molto nel filesystem e sono parzialmente dipendenti dal percorso del file. Per esempio, i file in `/bin`, `/sbin`, `/usr/sbin`, `/usr/bin` e directory simili hanno tipicamente una vita molto lunga: molti mesi e oltre. I file in `/home` hanno una vita media: circa qualche settimana. I file in `/var` hanno vita breve: quasi nessuno in `/var/spool/news` resterà per più di qualche giorno, mentre in `/var/spool/lpd` i file durano pochi minuti.

Per il backup è utile che la quantità di dati salvati nel backup giornaliero sia minore della capacità del singolo dispositivo di backup. Un backup giornaliero può essere completo o incrementale.

Potete decidere di tenere la dimensione delle partizioni abbastanza piccola da farle entrare completamente su un singolo backup (scegliete backup giornalieri completi). In ogni caso una partizione dovrebbe essere abbastanza breve che i cambiamenti da un giorno all'altro (tutti i file modificati) stiano su un solo backup (scegliete il backup incrementale e cambiate dispositivo di backup per il salvataggio totale settimanale o mensile - non è possibile una operazione automatica).

La vostra strategia di backup dipende da quella decisione.

Quando progettate di comprare spazio su disco, ricordate di mettere da parte dei soldi per il backup! I dati non salvati non valgono nulla! I costi per riprodurli sono molto maggiori di quelli per salvarli per chiunque!

Per ragioni di prestazioni, è utile tenere i file con diversi tempi di vita su differenti partizioni. In questo modo i file con vita breve sulla partizione delle news possono anche essere molto frammentati, ma questo non avrà impatto sulle prestazioni della partizione `/` o `/home`.

4 Un esempio

4.1 Un modello raccomandato per principianti ambiziosi

Un modello comune crea le partizioni `/`, `/home` e `/var` come discusso in precedenza. Così è semplice installare, mantenere e la distinzione è abbastanza buona da evitare effetti indesiderati a causa dei differenti tempi di vita. Si adatta bene a un modello di backup: quasi nessuno si preoccupa di salvare gli spool USENET, e solo qualche file in `/var` vale la pena di essere salvato (mi viene in mente /var/spool/mail). D'altra parte, `/` cambia raramente e può venire salvata su richiesta (dopo cambiamenti di configurazione) ed è abbastanza piccola da stare nei moderni sistemi di backup come un backup completo (tipicamente dai 250 ai 500 MB, dipende dal software installato). `/home` contiene importanti dati degli utenti e dovrebbe essere salvata in modo giornaliero. Qualche installazione ha delle partizioni `/home` molto grandi e si devono usare i backup incrementali.

Alcuni sistemi mettono `/tmp` su una partizione separata, altri creano un link simbolico a `/var/tmp` per ottenere lo stesso effetto (notare che questo può influire sul modo utente singolo, dove `/var` non sarà disponibile e nel sistema non ci sarà `/tmp` finché non ne creerete una o monterete `/var` manualmente) oppure la mettono su un disco RAM (Solaris, ad esempio, lo fa). Questo lascia `/tmp` separato da `/`, che è una buona idea.

Questo modello è conveniente per aggiornamenti o reinstallazioni: salvate i vostri file di configurazione (o l'intero ramo `/etc`) in qualche directory `/home`, cancellate `/`, reinstallate e ripristinate le vecchie configurazioni dalla directory di salvataggio in `/home`.

5 Come ho fatto sulla mia macchina

Avevo un vecchio 386/40 che giaceva sugli scaffali, l'avevo abbandonato due anni fa perché non potevo più utilizzarlo. Pianificavo di trasformarlo un un piccolo server (senza X-Window) per la mia rete casalinga.

Ecco come ho fatto: ho preso quel 386 e ci ho messo 16 MB di RAM, ho aggiunto un disco EIDE economico, il più piccolo che ho trovato (800 MB) e una scheda di rete Ethernet. Ho messo anche una vecchia Hercules, dato che avevo ancora un monitor. Ci ho installato Linux e i server NFS, SMB, HTTP, LDP/LPR, NNTP oltre che il mail router e il server POP3. Con l'aggiunta di una scheda ISDN la macchina divenne il mio router TCP/IP con firewall.

La maggior parte dello spazio su disco andava nel ramo `/var`, `/var/spool/mail`, `/var/spool/news` e `/var/httpd/html`. Ho messo `/var` su una grossa partizione separata. Ci saranno pochi utenti su questa macchina, perciò non ho creato partizioni home e monto `/home` da qualche altra workstation via NFS.

Linux senza X più qualche utility installata localmente sta comodo in 250 MB di partizione `/`. La macchina ha 16 MB di RAM, ma fa girare molti server. Servono minimo altri 16 MB di swap, meglio 32 MB. Non abbiamo problemi di spazio, perciò ne ho messi 32. Per motivi sentimentali ho lasciato 20 MB di partizione MS-DOS. Avendo deciso di importare `/home` da un'altra macchina, i rimanenti 500 e oltre MB andranno in `/var`. Sono più che sufficienti per un utilizzo domestico di scarico dei newgroup USENET.

Abbiamo

```
Device     Montato su                     Dimensioni
/dev/hda1  /dos_c                          25 MB
/dev/hda2  - (Spazio swap)                 32 MB
/dev/hda3  /                              250 MB
/dev/hda4  - (Partizione Estesa)          500 MB
/dev/hda5  /var                           500 MB

homeserver:/home /home                    1.6 GB
```

I salvataggi vengono effettuati via rete usando il nastro su `homeserver`. Poiché è stato tutto installato da CD-ROM, devo salvare solo qualche file di configurazione in `/etc`, i miei *.tgz personalizzati installati localmente da `/root/Source/Installed` e `/var/spool/mail`, assieme a `/var/httpd/html`. Copio questi file in una directory dedicata `/home/backmeup` su `homeserver` ogni notte, per essere prelevati dalla procedura di backup ordinario di homeserver.