

1 Il problema da risolvere

Far coesistere diversi sistemi operativi “live” o installer di sistemi operativi sullo stesso disco USB (o anche interno, volendo, anche se meno utile in pratica) senza alcun conflitto tra essi.

2 Com'è stato risolto fino ad oggi?

2.1 SARDU - <http://www.sarducd.it/> (o altri software simili)

Svantaggi che vedo in questa soluzione:

- Molto spesso i diversi sistemi operativi hanno file con lo stesso nome, e dovendoli alloggiare sulla stessa partizione vengono rinominati con conseguente cambio dei file di configurazione (quando va bene, sistemi open) o hex-edit di file binari (nei casi peggiori, Windows)
- Richiede personalizzazione per ogni singola distribuzione
- Spesso gli aggiornamenti provocano problemi che devono essere risolti manualmente
- Alcune “coesistenze”, specialmente tra differenti Windows PE e diverse installazioni di Windows sono (quasi) impossibili.

Vantaggi:

- Funziona.

2.2 Emulazione lettore CD tramite sostituzione dell' interrupt handler 13 (accesso al disco)

Mi riferisco ad emulazioni come quella di grub4dos o di memdisk.

Vantaggi:

- La pen-drive risulta molto ordinata e non crea problemi di manutenzione

Svantaggi:

- Non funziona: Qualsiasi sistema operativo “moderno” accede all'hardware tramite i driver del controller del disco e non tramite l'INT 13h quindi non è possibile far funzionare l'emulazione per le fasi successive del boot.

Fonte: manuale di grub4dos (<http://diddy.boot-land.net/grub4dos/files/map.htm>) e wiki di memdisk (http://syslinux.zytor.com/wiki/index.php/MEMDISK#INT_13h_access:_Not_all_images_will_boot_completely.21)
L'utilizzo di Firadisk, WinVBlock, memdiskfind o simili richiede modifiche al singolo sistema operativo, con tutti i problemi che ne conseguono.

3 Cosa propongo

Ormai la quasi totalità dei sistemi “live” o dei setup può partire da disco USB oltre che da cdrom.

Nel caso più semplice (un solo sistema) esso sta nella prima partizione primaria del disco in questione. Cerchiamo di “ricostruire” questa struttura.

Windows (XP, Vista, 7) richiede che sia formattato in NTFS, e se il dispositivo USB indica di essere rimovibile, considera accessibile soltanto la prima partizione primaria.

Linux consente di avere un filesystem ISO9660 in una partizione di un disco, motivo grazie al quale sono recentemente nate le “iso ibride”.

Quasi tutti gli script contenuti negli initramfs odierni (caricati tramite int13h) cercano in tutti i block device (partizioni comprese) a loro disposizione un filesystem contenente un file “noto”.

Quindi creando più partizioni eventuali distribuzioni tra loro “strettamente parenti” verrebbero confuse.

Si può quindi salvare come file CONTIGUI in una partizione le immagini ISO (o le immagini della chiavetta mono-OS acquisite con dd) e far modificare al boot loader l'entry della prima partizione primaria nell'MBR per far puntare la zona (CONTIGUA) del disco dove c'è l'immagine selezionata dall'utente in fase di boot.

Attualmente molti boot loader supportano la modifica dell'MBR per attivare / disattivare il flag avviabile delle partizioni o modificarne il tipo per nasconderle / mostrarle, ma non i cambiamenti di “posizione” della partizione.

Quindi ogni sistema operativo avviato si troverebbe, una volta caricati i driver, la prima partizione primaria del device di boot con il contenuto a lui più gradito senza forzate coesistenze né modifiche al sistema operativo stesso.

4 Come penso di farlo funzionare?

Fase di boot:

1. La partizione attiva è la seconda. Viene caricato Grub2
2. L'utente sceglie un entry dal menù
3. Il primo comando da eseguire modifica tipo offset e lunghezza della prima partizione del disco secondo i parametri passati come argomenti.
 - (a) La lettura diretta dei metadati del filesystem, come verificabile con il comando `blocklist`, richiede tempo proporzionale alla lunghezza del file, ed è facilmente evitabile in questo caso.
4. Immagine ISO con ISOLINUX o immagine di partizione?
 - (a) Immagine di partizione: chainloading del boot sector della partizione e prosecuzione normale del boot.
 - (b) Immagine ISO con ISOLINUX:
 - i. Avvio di Grub4Dos che caricherà `/menu.lst`
 - ii. Il cui contenuto è fisso e riassumibile con "Emula un lettore CD il cui contenuto è la prima partizione primaria, e avvia da quel CD virtuale"

Per la scrittura del file di configurazione di Grub2, contenente offset e lunghezza delle varie "finte partizioni" corrispondenti ai file, basta utilizzare l'ioctl `FIBMAP` con cui si può verificare contiguità e posizione iniziale del file. Un semplice script (+ helper in C) risolve il problema.

Per la scrittura di file contigui su Linux esiste la systemcall "fallocate" che viene usata per comunicare al filesystem la dimensione che avrà il file in modo che possa essere scelto lo spazio libero contiguo ottimale.

5 Svantaggi noti?

Nel device in questione si troverà una tabella delle partizioni con partizioni "sovrapposte" fra loro.

A meno di usare tool di partizionamento particolarmente "automatizzati", nell'uso comune non si dovrebbero incontrare grossi problemi.

Una possibile soluzione è salvare offset e lunghezza della partizione 3 in un menu entry di grub chiamato "Ripristina partizione di deposito immagini" ed eliminarla dalla tabella delle partizioni prima di avviare un altro sistema operativo. Appare tanto "spazio inutilizzato" sul disco, ma finché nessuno ci scrive su, non è un problema e la tabella delle partizioni non sembra scorretta.

Ulteriore possibilità, per non avere nemmeno le partizioni non in ordine: creare la partizione 2 alla fine del disco, e usare sempre il 1° record della tabella delle partizioni anche per la partizione di storage.

6 Bello, ma in pratica?

6.1 Ricompilare grub1.99

NB: Per i più pigri e fiduciosi: inserire il `setpart.mod` fornito nella cartella `/boot/grub/` nella partizione dove installerete grub-1.99 e potete evitare di ricompilare grub.

1. Applicare la patch `setpart.patch`, il cui "contenuto principale" è il file `setpart.c`
2. Ricompilare Grub1.99

```
./autogen.sh ;
./configure
make -j $(cat /proc/cpuinfo | grep ^processor | tail -n1 | awk '{print 2 + $3}')
```

```
make install
```

6.2 Creare una tabella delle partizioni sul device in questione:

N°Partizione	Dimensione	Tipo	Filesystem	Flag avviabile	Utilizzo
1	Min 1 settore	83	non importa	NO	Entry dell'MBR che sarà sovrascritto.
2	Min 20MB	83	ext2	SI	Conterrà il bootloader
3	Max disp.	83	XFS	NO	Storage delle immagini

Il filesystem XFS conviene crearlo con l'opzione `-d agcount=1` per minimizzare la possibilità che i file grossi diventino frammentati.

Un file spezzato in due frammenti, sebbene perfettamente accettabile per usi normali, in questo caso diventa assolutamente inservibile.

6.3 Preparare la partizione 2

Installando Grub1.99 nell'MBR usando come boot directory il mount point dove avrete montato la partizione 2.
Copiare l'eseguibile di grub4dos e menu.lst nella root della partizione 2

6.4 Compilare l'utility fibmap

NB: Per i più pigri e fiduciosi, con architettura x86_64: usate il binario fornito.

```
gcc -o fibmap fibmap.c
```

6.5 Mettere le immagini nella partizione 3

Usando lo script copia_contiguo che utilizzando falloccate tenta di ottenere un file contiguo. Qualora il file non risulti effettivamente contiguo scrive a video e ritorna 1.

Sintassi:

```
./copia_contiguo /directory/sorgente/file.iso /directory/destinazione/file.iso
```

6.5.1 Generare la configurazione di Grub

```
./writeconfig /mount/point/della/partizione3 >/mount/point/della/partizione2/boot/grub/grub.cfg
```

La configurazione generata è valida per i file .iso .

Qualora siano state salvate immagini di intere chiavette USB o di singole partizioni (Setup di Windows???) è necessario modificare l'entry nel file grub.cfg generato, cambiando:

- Il tipo, 7 per NTFS, c per FAT32 della partizione considerata
- Qualora il file immagine contenga anch'esso un MBR aggiungere al parametro startlba l'inizio della prima partizione all'interno del file (tipicamente 63 settori)
- Utilizzare l'effettiva lunghezza della partizione in settori.

Correzione del bootsector delle partizioni NTFS (se presenti) Il boot sector delle partizioni NTFS contiene l'offset dell'inizio della partizione.

Bisogna correggerlo utilizzando <https://sourceforge.net/projects/ntfsfixboot/> indicando manualmente il settore di avvio, copiando il parametro -startlba dal file di configurazione di grub2.

Possibile evoluzione futura: Mettere questi script, una volta completamente automatizzati, in un initramfs e creare nel boot loader una voce "Aggiorna lista ISO". Ciò ha particolarmente senso qualora la partizione contenente le immagini abbia un filesystem NTFS o FAT32 (per i quali esiste l'utility "contig") in modo che il tutto sia facilmente mantenibile da sistemi Windows.

6.6 Reboot e test.

Sono graditi commenti.

Massimo Maggi

massimo@mmmm.it