

Masterizzare col pinguino

Andrea Chiffi a.k.a. much0 <sixware@inwind.it>

03/09/2002 - 27/04/2003

Guida pratica alla masterizzazione in GNU/Linux da linea di comando, pensata e realizzata come contributo al sito del [SaLUG!](#) (Salento Linux User Group) e per tutta la comunità Open Source.

Contents

1	Pre-messa	2
2	Copyright, license and terms of usage (in english)	2
3	Note iniziali	2
4	Un po' di teoria	3
5	Configurazione del sistema (emulazione SCSI)	4
6	Strumenti indispensabili	5
7	La famosa "tripletta" di BIL (scsi Bus, Id, Lun)	5
8	Siete pronti? Puntate il laser... FUOCO!!! (operazioni di base)	6
8.1	CD Dati	6
8.1.1	Copiare un cd	6
8.1.2	Creare un cd	7
8.1.3	Cancellare un cd riscrivibile	8
8.1.4	Verificare un cd	8
8.2	CD Audio	9
8.2.1	Copiare un cd audio	9
9	Per veri piromani... (operazioni avanzate)	9
9.1	Montare un file immagine	9
9.2	Vado al massimo...	10
9.2.1	Compressione dei dati di un cd	10
9.2.2	Overburning ovvero "scrivi finchè c'è plastica"	10
9.3	Copiare al volo	11
9.4	Ottimizzare il sistema per masterizzare	12

10 Thanks 2...	12
11 ToDo	12
12 Note finali (fiuuuu...)	13

1 Pre-messa

Questa guida non si sostituisce ai numerosi *HOWTO* di riferimento per la masterizzazione in GNU/Linux, ma si propone di dare un approccio di tipo più pratico, senza tuttavia tralasciare una minima nozione teorica pur necessaria e indispensabile. Questo documento nasce quindi con l'intento di aiutare gli utenti più esperti a masterizzare con configurazioni o in formati particolari e avvicinare i newbies alla *conoscenza* :) ed alla potenza fornita dalla shell.

2 Copyright, license and terms of usage (in english)

Masterizzare col pinguino, Copyright (c) 2002-2003 Andrea Chiffi <sixware@inwind.it>

This is free document; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This document is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this document or GNU CC; if not, write to the: Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

3 Note iniziali

Ecco una lista delle convenzioni adottate in questo testo:

- il simbolo del dollaro **\$** prima di ogni comando indica che il comando può essere eseguito da qualsiasi utente;
- il simbolo cancelletto **#** invece indica che questo deve (!) essere eseguito dall'utente **root**. (Per acquisire i privilegi di root basta digitare **\$ su** ed immettere la relativa password);
- il simbolo **?** inserito in alcune espressioni è un carattere *jolly*, cioè può assumere diversi valori. In questo testo il punto interrogativo sostituisce una qualsiasi lettera dell'alfabeto ([a-z] ovvero a, b, c, d, ecc.). Ad esempio *?ndrew* sta per *andrew* o *bndrew* o *cdrew*, ecc. fino a *zndrew*;
- il simbolo **N** è un altro carattere jolly che assume solo valori numerici (0, 1, 2, 3, ecc.).

4 Un po' di teoria

I masterizzatori e i lettori cd possono essere di due tipi: **IDE/ATAPI** o **SCSI** (chi ha il masterizzatore SCSI può saltare le seguenti righe e andare a leggere subito la sezione 6 (Strumenti indispensabili)).

La maggior parte di quelli “casalinghi” sono IDE/ATAPI, ma Linux gestisce solo masterizzatori SCSI: nessun problema! Per questi dispositivi il nostro amato pinguino ci permette di utilizzare la cosiddetta **emulazione SCSI**.

I lettori cd e i masterizzatori IDE sono visti dal sistema come `/dev/hd?` dove il punto interrogativo sta per:

- **a** se il dispositivo è collegato come **master** sul controller primario;
- **b** se il dispositivo è collegato come **slave** sul controller primario;
- **c** se il dispositivo è collegato come master sul controller secondario;
- **d** se il dispositivo è collegato come slave sul controller secondario.

Se non sapete come è collegato il vostro masterizzatore allora, in alternativa, potete:

- aprire il computer e guardarci dentro :o (occhio alla garanzia!)
- aprire un terminale e digitare il comando `$ dmesg | grep hd`
- guardare i file relativi ai dispositivi IDE `$ cat /proc/ide/hd?/model`

Tenete presente comunque che una configurazione tipica è:

- un hard-disk collegato come master sul primo **controller**;
- il masterizzatore come master sul secondo controller;
- il lettore CD o DVD come slave sul secondo controller;

oppure:

- un hard-disk collegato come master sul primo controller;
- il lettore CD o DVD come slave sul primo controller;
- il masterizzatore come master sul secondo controller.

Quest'ultima configurazione permette, tra l'altro, di masterizzare cd “al volo” o, per dirla in altri termini, *on the fly*; ovvero copiare i cd direttamente dal lettore al masterizzatore, senza utilizzare un'**immagine** temporanea sull'hard-disk. Ma di questo ne parleremo più avanti.

5 Configurazione del sistema (emulazione SCSI)

Ora è il momento dell'emulazione SCSI vera e propria che si effettua semplicemente passando il parametro `hd?=ide-scsial kernel` al momento del `boot`.

Si deve quindi, in alternativa:

- ad ogni avvio di Linux digitare `hd?=ide-scsi` al prompt di **L**ILO (LInux LOader) o di **G**RUB (GRand Unified Bootloader); ad esempio con LILO `"boot: linux hdc=ide-scsi"`
- ricompilare il **kernel** (panic! :P). Facendo riferimento ai kernel della serie 2.4.x dovranno essere abilitate le seguenti caratteristiche: nella sezione ATA/IDE/MFM/RLL support e IDE, ATA and ATAPI Block devices scegliere di includere (come modulo) le voci Include IDE/ATAPI CDROM support e SCSI emulation support, mentre nella sezione SCSI support le voci SCSI support, SCSI CD-ROM support e SCSI generic support. Inoltre nella sezione File systems abilitare ISO 9660 CDROM file system support, Microsoft Joliet CDROM extensions, Transparent decompression extension e la voce Loopback device support presente nella sezione Block devices.
- se utilizzate GRUB dovete editare il file `/boot/grub/menu.lst`, aggiungendo `hd?=ide-scsi` alla riga contenente la parola *kernel* (ricordate poi di re-installare il vostro boot-loader in modo che carichi la nuova configurazione `# grub-install /dev/hda` dove `hda` è il vostro hard-disk)
- se invece utilizzate LILO dovete editare `/etc/lilo.conf` aggiungendo `append="hd?=ide-scsi"` nella sezione relativa al kernel che state utilizzando (ricordate poi di re-installare il vostro boot-loader in modo che carichi la nuova configurazione `# lilo -v`)

Al prossimo riavvio del sistema, il masterizzatore ed il lettore cd non saranno più disponibili come `/dev/hd?`, ma come `/dev/scdN` o `/dev/srN` se utilizzati in lettura e `/dev/sgN` se utilizzati in scrittura, dove 'N' è un numero. Per verificarlo date un'occhiata al file `/proc/scsi/scsi` o date il comando `$ dmesg`

```
SCSI subsystem driver Revision: 1.00
scsi0 : SCSI host adapter emulation for IDE ATAPI devices
  Vendor: PLEXTOR   Model: CD-R   PX-W8432T   Rev: 1.09
  Type:   CD-ROM                               ANSI SCSI revision: 02
  Vendor: GoldStar Model: CD-ROM GCD-R580B   Rev: 1.05
  Type:   CD-ROM                               ANSI SCSI revision: 02
Attached scsi CD-ROM sr0 at scsi0, channel 0, id 1, lun 0
Attached scsi CD-ROM sr1 at scsi0, channel 0, id 0, lun 0
sr0: scsi3-mmc drive: 32x/32x writer cd/rw xa/form2 cdda tray
sr1: scsi3-mmc drive: 1x/7x xa/form2 cdda tray
```

A questo punto vi conviene creare un collegamento simbolico (*link*) a questi nuovi dispositivi assegnandogli un nome più umano ed aggiornare il file `/etc/fstab` :

- `# ln -sf /dev/scd1 /dev/cdrom` per il lettore cd
- `# ln -sf /dev/scd0 /dev/cdburner` per il masterizzatore

6 Strumenti indispensabili

Per poter “bruciare” i vostri cd dovete installare i seguenti programmi:

cdrecord

è il motore primario della masterizzazione e supporta cd dati, audio, mixed mode, multisessione e CD+;

cdrdao

analogo a **cdrecord** ma più utilizzato per masterizzare cd audio poichè utilizza la modalità Disc-At-Once (**DAO**);

mkisofs

serve per creare immagini **ISO9660**, che è il *filesystem* standard dei cd. Ad esso possono essere aggiunte delle *estensioni* come le **Rock Ridge** tipiche dei sistemi UNIX (che gestiscono anche i permessi dei file e i link simbolici) o le **Joliet** utilizzate, invece, dai sistemi operativi Micro\$oft;

mkhybrid

come **mkisofs** ma con la possibilità di creare immagini con HFS, il filesystem tipico dei Macintosh;

mkzftree

tool per creare (insieme a **mkisofs**) immagini compresse

dd

sta per *disk dump* e serve, tra l'altro, per creare le immagini dei cd che si vogliono copiare;

cdda2wav

serve per estrarre le tracce audio da un cd e memorizzarle in file **wave**;

cdparanoia

come **cdda2wav** ma più preciso, quasi paranoico, nel senso che nel catturare le tracce audio effettua anche ulteriori correzioni come quelle per il *jitter* o gli *scratch*.

7 La famosa “tripletta” di BIL (scsi Bus, Id, Lun)

Prima di qualsiasi altra parola eseguite il comando `$cdrecord -scanbus` in un qualsiasi terminale. Se tutto è andato bene dovrete vedere una lista dei vostri dispositivi a cui sono associati 3 numeri (esempio: **0,1,0**).

Dovrebbe venirvi fuori qualcosa del genere:

```
Cdrecord 1.11 Copyright (C) 1995-2001 Jörg Schilling
Linux sg driver version: 3.1.22
Using libscg version 'schily-0.5'
scsibus0:
0,0,0 0) 'GoldStar' 'CD-ROM GCD-R580B' '1.05' Removable CD-ROM
0,1,0 1) 'PLEXTOR ' 'CD-R PX-W8432T' '1.09' Removable CD-ROM
0,2,0 2) *
```

```
0,3,0 3) *
0,4,0 4) *
0,5,0 5) *
0,6,0 6) *
0,7,0 7) *
```

(in questo esempio il GoldStar **0,0,0** è il mio lettore cd ed il PLEXTOR **0,1,0** è il masterizzatore).

A cosa serve questa tripletta? Semplice: **cdrecord** la utilizza per riferirsi ad un particolare dispositivo.

Se volete visualizzare la lista delle caratteristiche del vostro amato “forno a laser” basterà digitare il comando
`$ cdrecord dev=N,N,N -prcap` dove l’opzione `dev=N,N,N` indica al programma quale dispositivo utilizzare.

8 Siete pronti? Puntate il laser... FUOCO!!! (operazioni di base)

Ecco una serie di operazioni di base con il relativo comando (ricordate che per masterizzare avete bisogno dei privilegi di **root** e, se non li avete, chiedeteli al vostro amministratore di sistema, oppure prendetevi da soli ;-)

8.1 CD Dati

Partiremo col trattare i cd di dati.

Se volete fare solo prove vi consiglio di aggiungere l’opzione `-dummy` nella riga di comando di **cdrecord** poichè questo vi permette di simulare la masterizzazione dato che il laser, in realtà, rimane spento.

8.1.1 Copiare un cd

Per copiare un cd è necessario che il contenuto di questo cd venga letto e memorizzato sotto forma di file. Fino ad oggi, i due tagli comunemente utilizzati sono di 650MB o 700MB, quindi avrete bisogno di questo spazio sul vostro hard-disk per creare il file **immagine**. L’ultimo passo è quello della masterizzazione vera e propria (intesa come la *rogo* :) di un cd) e verrà effettuata utilizzando il programma **cdrecord**, precedentemente descritto.

Ecco, in ordine, i comandi da usare:

```
$ dd if=/dev/cdrom of=image.iso bs=2048 conv=notrunc
```

(copia il cd creando il file immagine `image.iso` sull’hard-disk)

- **if** sta per Input File e specifica l’origine dei dati
- **of** sta per Output File (che fantasia! :) ed indica invece la destinazione dei dati

Per creare il file immagine (senza usare **dd**) si può, in alternativa usare il programma **readcd** presente nelle ultime versioni di `cdrecord` (`$ readcd dev=0,0,0 f=image.iso`).

```
# cdrecord dev=0,1,0 speed=8 -v -eject -data image.iso
```

(masterizza il cd a partire dal file immagine `image.iso`)

- **dev=N,N,N** indica il dispositivo di scrittura (masterizzatore)
- **speed=8** indica la velocità di masterizzazione (es. 8x)
- **-eject** serve per far “sputare” il cd una volta finita la masterizzazione (a volte non supportato in modalità simulazione)
- **-data** indica il tipo di traccia da scrivere (traccia dati)

Se avete un masterizzatore abbastanza moderno, questo supporterà sicuramente la tecnologia *Burn-Proof* (Plextor, Sanyo) o *Just-Link* (Ricoh, Philips) o *Lossless-Link* (Yamaha) e quindi vi conviene aggiungere l'opzione **driveropts=burnfree** nella riga di comando di **cdrecord** per evitare di bruciare i cd quando si è in condizioni di *buffer underrun* ovvero quando al masterizzatore non arrivano più dati da scrivere, poichè la memoria buffer si è svuotata per problemi imprevisti, ed il laser non sa più se scrivere 1 o 0 :).

8.1.2 Creare un cd

```
$ mkisofs -f -J -l -r -o image.iso -P much0 -p much0 -V NEW -v ~/iso/
```

Il comando crea un file immagine a partire dai file e directory presenti in `~/iso`; così facendo, una volta masterizzato il cd, i file presenti in `~/iso` sull'hard-disk, si verranno a trovare nella directory principale (radice) del cd.

- **-f** comunica a **mkisofs** di seguire i collegamenti o *link* che quindi verranno memorizzati sul cd come file normali (questa opzione può anche essere omessa nel caso in cui, ad esempio, si vuole fare un backup della *home* directory);
- **-J** e **-r** abilitano rispettivamente le Joliet e le rationalized Rock Ridge Extension (nel caso in cui si sta facendo un backup, ad esempio della *home* directory, è consigliabile sostituire questa coppia di opzioni con **-R** che, a differenza di **-r**, mantiene i permessi originali dei file);
- **-l** permette di utilizzare nomi di file di 31 caratteri (ciò è consentito nello *standard* ISO9660), invece che del formato 8.3 caratteri (omettendo questa opzione si potrà ottenere un cd compatibile con M\$-DOS);
- **-o** sta per Output ed indica il file che verrà creato;
- **-P** imposta il nome dell'editore;
- **-p** imposta il nome del preparatore dei dati;
- **-V** imposta l'etichetta o *label* del cd (nel nostro caso NEW)
- **-v** sta per *verbose* ossia visualizza molto output

Attenzione! Nel caso si renda necessario utilizzare il cd in ambiente DOS (che utilizza il formato 8.3 per i nomi dei file) e si vuole comunque memorizzare i nomi dei file utilizzando i 31 caratteri, conviene inserire nella riga di comando l'opzione **-T** che genera, in ogni directory del cd, il file di testo `TRANS.TBL` contenente le corrispondenze dei nomi dei file in 8.3 e 31 caratteri, e l'opzione **-hide-joliet-trans-tbl** che nasconde questi file nella struttura Joliet con la conseguenza che non saranno visibili sotto Windows.

Ora, finalmente, è arrivato il momento di masterizzare il file immagine con il comando già visto:

```
# cdrecord dev=0,1,0 speed=8 -v -eject -data image.iso
```

A volte, però, questo metodo risulta dispendioso in termini di spazio disponibile sull'hard-disk in quanto i dati, prima di essere scritti su cd, devono essere organizzati nel file immagine; quindi se ho 700B di “robba” da masterizzare dovrò avere 700MB aggiuntivi di spazio sull'hard-disk per creare il file immagine. Tutto questo spreco si può evitare semplicemente concatenando i due comandi appena visti con una **pipe**:

```
# nice --18 mkisofs -f -J -r -quiet -V NEW ~/iso/ | cdrecord dev=0,1,0 speed=8 fs=16m
tsize='mkisofs -f -J -r -quiet -print-size ~/iso/'s -v -eject -
```

Come avranno sicuramente notato i più scaltri sono state aggiunte delle ulteriori opzioni e comandi. In dettaglio:

- **nice -18** serve ad assegnare a **mkisofs** una **priorità** più alta per evitare che altri processi “mangino” troppe risorse provocando un errore di *buffer underrun* e rendendo quindi inservibile il cd ormai bruciato;
- **-quiet** serve per far “star zitto” **mkisofs** che quindi non produrrà nessun output che verrebbe a sovrapporsi a quello di **cdrecord**;
- **fs=16m** specifica la dimensione del **buffer** da utilizzare (in questo caso è stato scelto il valore di 16MB in quanto la concatenazione dei due comandi rende più rischiosa l'operazione di masterizzazione). Per chi non lo sapesse la memoria buffer è un particolare tipo di memoria che immagazzina temporaneamente i dati per compensare differenze di *transfer rate* esistenti fra due periferiche. Durante la masterizzazione il masterizzatore, prima dell'avvento della tecnologia *Burn-Proof* o *JustLink*, ha bisogno di un flusso continuo di dati. Qualora il sistema non riesca a inviare la quantità di dati richiesta allora provvederà la memoria buffer a garantire per un breve tempo, nell'ordine dei secondi, la continuità necessaria. Quando questa si svuotasse, però, si avrà il *buffer underrun* error, con conseguente fallimento della masterizzazione. Si capisce da ciò quanto sia importante avere una buona disponibilità di memoria buffer in configurazioni prive di sistemi ulteriori di controllo;
- **tsize=Ns** dove 'N' è il numero di **settori** ed 's' è proprio una esse ;) comunica a **cdrecord** la dimensione (in settori da 2048 bytes) dei dati da scrivere su cd. I veterani avranno sicuramente intuito che il numero N è in realtà calcolato da **mkisofs** e poi passato a **cdrecord** attraverso l'utilizzo degli apici inversi (se non ci hai capito niente non ti preoccupare... semplicemente copia e incolla il comando! ;)

8.1.3 Cancellare un cd riscrivibile

Si può scegliere tra la cancellazione veloce che cancella solo la TOC (table of contents):

```
# cdrecord dev=0,1,0 blank=fast
```

oppure quella lenta che cancella tutti i settori del cd:

```
# cdrecord dev=0,1,0 blank=disk
```

8.1.4 Verificare un cd

Esistono molti modi per verificare che un cd appena sfornato non contenga errori. Ve ne suggerisco alcuni:

- `$ readcd dev=0,1,0 f=/dev/null`
- `$ dd if=/dev/cdrom of=/dev/null`
- `$ cat /dev/cdrom > /dev/null`

8.2 CD Audio

Permettetemi una piccola premessa prima di iniziare:

da accurati studi di riviste specializzate di audio professionale sembra che i CD audio masterizzati ad alte velocità siano di **qualità** inferiore. Vero è che per giungere a queste conclusioni si è dovuti ricorrere a strumenti molto più sensibili dell'orecchio umano. Si può dire dunque che è meglio masterizzare non alla massima velocità, pur essendo le differenze a volte nemmeno udibili, tutto dipende anche dalla qualità del masterizzatore utilizzato.

8.2.1 Copiare un cd audio

Bisogna prima estrarre le tracce audio dal cd sorgente (in gergo *grabbare*).

Possiamo usare indistintamente i due programmi più noti: `cdda2wav` o `cdparanoia`.

```
$ cdda2wav -D 0,1,0 -B -H -P 0 ~/DAE/autore
```

- **-D** indica il lettore in cui è il cd da copiare
- **-B** copia tutte le tracce
- **-H** evita la scrittura di file `inf` e `cddb`
- **-P N** dove N sono i settori di overlap per la correzione jitter (0 = nessuna correzione)
- `~/DAE/autore` sono directory e nome dei file wave a cui viene aggiunto un numero (es. `~/DAE/autore_01.wav`)

```
$ cdparanoia -B -Z -d /dev/cdrom
```

- **-B** copia tutte le tracce
- **-Z** smette di essere paranoico!
- **-d** specifica il lettore in cui è il cd da copiare

Non so se è possibile dare un nome per tutte le tracce wav, come in `cdda2wav`.

9 Per veri piromani... (operazioni avanzate)

9.1 Montare un file immagine

Questo comando è utile nei casi in cui si vuole controllare un'immagine appena creata, prima di essere scritta definitivamente su cd. Fornisce, quindi, una sorta di anteprima della struttura con cui appariranno i file una volta masterizzati.

```
mount -t iso9660 -o loop,ro image.iso /mnt/iso/
```

Ovviamente:

- il punto di montaggio, ossia la directory `/mnt/iso` deve esistere;
- `image.iso` va sostituito con il nome del tuo file immagine;
- il tuo kernel dovrebbe avere il supporto per il *loopback filesystem* (in genere in `/lib/modules/kernel-version/kernel/drivers/block/` dovrebbe esserci un modulo chiamato `loop.o`);
- il montaggio avviene in sola lettura (proprio come avviene con i cd);
- in `/mnt/iso` troverai tutti i file contenuti nell'immagine iso.

9.2 Vado al massimo...

...vado a gonfie vele!

Riprendendo una vecchia canzone di *Vasco*, in questa sezione ci occuperemo di “sfondare” il limite di 650 o 700 MB dei comuni supporti in commercio. In particolare vedremo come le tecniche di compressione dei dati (9.2.1 ()) e di overburning (9.2.2 ()) consentono di sfruttare al massimo tutto lo spazio disponibile sul cd da masterizzare.

9.2.1 Compressione dei dati di un cd

Con i kernel Linux 2.4.14 o successivi si può far uso di una interessante caratteristica: la **decompressione trasparente** dei file. Questa nuova funzionalità consente di poter masterizzare su cd dei dati che sono stati precedentemente compressi (e che quindi, nella maggior parte dei casi, occuperanno meno spazio degli originali). I dati presenti in questo cd saranno poi visibili all'utente come se non fossero stati compressi, grazie alla decompressione al volo effettuata dal kernel.

Un vantaggio innegabile nell'uso di questa caratteristica è quello di . L'unico svantaggio è dato dalla non-compatibilità di questo cd, che sarà quindi letto solo dal sistema operativo Linux dotato di kernel recente.

Ma entriamo subito in azione! Per creare un'immagine compressa, prima di usare **mkisofs**, lanciamo il comando:

```
mkzftree ~/iso/ ~/ziso/
```

Questo tool copia i dati presenti in `~/iso/` comprimendo tutti file e memorizzandoli con gli stessi percorsi in `~/ziso/`.

Ora diamo il solito comando per creare il file immagine:

```
$mkisofs -f -J -r -z -o zimage.iso -V COMPRESSED_CD -v ~/ziso/
```

facendo attenzione ad aggiungere l'opzione **-z** che serve a **mkisofs** per capire che sta lavorando su una directory (nel nostro esempio `~/ziso/`) contenente file compressi. Ora si può masterizzare su cd utilizzando il solito comando 8.1.1 (`cdrecord`).

9.2.2 Overburning ovvero “scrivi finchè c'è plastica”

L'*overburning* è una particolare modalità di scrittura che permette ad un masterizzatore di scrivere oltre alla capacità nominale del supporto CD. Per esempio, se un supporto normalmente può contenere 652 MB di

dati, in generale è possibile spingersi fino a 7-8 MB oltre quel limite. Da notare però che sia il masterizzatore che il software devono essere predisposti per questo tipo di scrittura. Sebbene **cdrecord** e **cdrdao** prevedono questa modalità, lo stesso non si può dire dei masterizzatori, poichè diversi marchi non lo supportano.

Per farvi capire meglio ci di cui sto parlando dico solo che, se masterizzare è diventato il vostro sport preferito, fare overburning è un pò come praticare uno sport estremo: non è per tutti e bisogna avere molto fegato (in questo caso molti cd vergini :) perchè comporta dei rischi in più.

Per praticare overburning si deve utilizzare la modalità di scrittura *Disk-At-Once* (DAO) e quindi bisogna specificare le opzioni **-dao -overburn** per **cdrecord** e **--overburn** per **cdrdao** (il nome stesso indica che questo programma utilizza la DAO come modalità di scrittura predefinita).

9.3 Copiare al volo

Se il cd contiene un'unica sessione dati senza altre diavolerie come la TOC illegale o alcuni settori danneggiati inseriti di proposito, possiamo utilizzare il nostro solito **cdrecord**:

```
# cdrecord dev=0,1,0 speed=8 fs=16m -isozsize -v -eject /dev/cdrom
```

(Nota che non c'è bisogno di montare il cd presente nel lettore in quanto **cdrecord** fa riferimento al suo device `/dev/cdrom`).

Se invece avete tra le mani un cd più originale e particolare (non so se mi sono spiegato! :P) bisogna utilizzare **cdrdao** che fornisce una copia fedele del cd che avete acquistato e di cui state facendo una (solo una!) copia per uso personale:

```
# cdrdao copy --on-the-fly --device /dev/cdburner --driver generic-mmc-raw
--source-device /dev/cdrom --source-driver generic-mmc-raw --speed 4 --buffers 480
--paranoia-mode 0 --read-subchan rw_raw
```

- **-device** è il masterizzatore;
- **-driver** specifica il tipo di driver da utilizzare per il masterizzatore (in questo caso *generic-mmc-raw*);
- **-source-device** è la sorgente dei dati ovvero il lettore cd `/dev/cdrom`;
- **-source-driver** specifica il driver riferito al lettore cd;
- **-speed N** è la velocità di scrittura;
- **-buffers N** è un'unità di buffer che corrisponde ad un secondo di audio scrivendo ad una velocità di 1x; dunque per sapere il tempo a disposizione prima di un "buffer underrun" si deve dividere il valore dei buffer per la velocità di scrittura (ad esempio 480 buffers / 4x = 120 sec. = 2 min.);
- **-paranoia-mode N** è il livello di correzione errori (in questo caso viene disabilitata per non rallentare l'operazione di lettura)
- **-read-subchan rw_raw** forza una lettura grezza (senza correzione di errori) dei sottocanali.

9.4 Ottimizzare il sistema per masterizzare

Prima di masterizzare assicuratevi che non vi siano processi “mangia-risorse”: in particolare è consigliabile sospendere anacron e cron (in quanto potrebbero lanciare processi come **locate** che accedono prepotentemente ai dischi) e disattivare temporaneamente il risparmio energetico (provate ad immaginare cosa accadrebbe se l’hard-disk si spegnesse nel bel mezzo di una masterizzazione!).

E come al solito, ecco i comandi necessari:

```
# /etc/init.d/anacron stop
# /etc/init.d/crond stop
# /etc/init.d/apm stop
# /etc/init.d/acpi stop
```

Una volta terminata la scrittura dei cd potete ricaricarli con:

```
# /etc/init.d/anacron start
# /etc/init.d/crond start
# /etc/init.d/apm start
# /etc/init.d/acpi start
```

Inoltre vi conviene disattivare lo *screensaver*, specie se particolarmente *pesante*, ed il risparmio energetico del monitor (DPMS).

10 Thanks 2...

Vorrei ringraziare le persone che hanno contribuito in qualche modo alla stesura di questo documento o ne hanno ispirato miglioramenti.

In ordine cronologico:

- tutto il **SaLUG!** (Salento Linux User Group) senza il quale tutto questo coinvolgimento emotivo non ci sarebbe stato;
- Benet Delia a.k.a. Beneth che ha ispirato la stesura di questa guida inizialmente nata per risolvere i suoi errori di masterizzazione :P;
- Francesco Palermo a.k.a. NoHouse che mi ha aiutato a scrivere la sezione sui cd audio.

11 ToDo

- Aggiungere “creazione di cd con file system criptato” nella sezione 9 (Per veri piromani);
- aggiungere “creazione di cd avviabili” nella sezione 9 (Per veri piromani);
- aggiungere “creazione di cd protetti” nella sezione 9 (Per veri piromani);
- aggiungere “creazione di cd multisessione” nella sezione 8 (Operazioni di base);

- aggiungere tutta la webografia e i vari URL dove reperire il software descritto nella sezione 6 (Strumenti indispensabili);
- completare e integrare la sezione 8.2 (CD Audio);
- altro che non mi viene in mente.

Sono ben accetti suggerimenti o critiche (meglio i primi ;).

Se sei interessato a collaborare alla stesura di questo documento scrivimi su sixware@inwind.it .

12 Note finali (fiuuuu...)

In GNU/Linux ci sono molti modi di fare le cose e non è detto che i metodi appena visti siano i più semplici: oGNUno è libero di utilizzarli o modificarli a suo piacimento.

Per i principianti: non vi preoccupate, i cd bruciati potete sempre utilizzarli come sotto-bicchieri! :)

Pace e BeneT... ...buona cottura!