

Convertire file audio/musicali

File audio a linea di comando

Volete nutrire il vostro player audio con in file musicale che non viene riconosciuto? LXP vi spiega come trasformarlo

L'autore

Emiliano Grilli

Ormai capita a tutti di avere a che fare con una moltitudine di file audio di diverso tipo. In questo articolo esamineremo i principali formati dell'audio digitale e alcuni strumenti e procedure di base per convertire le informazioni audio da e verso questi formati. Prima di cominciare, però, è necessaria una spiegazione di come funziona l'audio digitale e in cosa si differenzia dall'audio analogico: la trovate nel box presente nella pagina a fianco. Letto? Bene, allora si va a incominciare...

Estrazione di audio da CD

Una prima esigenza che sorge agli utenti di Compact Disc musicali è quella di voler trasformare le tracce del CD audio in file sul proprio hard disk (per esempio, per poi trasferirli in formato compresso nell'iPod). Nel mondo Linux esistono una gran quantità di software adatti a questo scopo che vanno sotto il nome di "CD rippers" (estrattori di CD). I più famosi sono **cdda2wav** e **cdparanoia**. Vi sono front-end grafici per questi comandi (ad esempio **grip**) che consentono anche di automatizzare il processo di estrazione e codifica in un formato compresso (come **MP3** o **Ogg**, di cui parleremo in seguito). Senza dilungarci nel dettaglio dell'uso di questi programmi (di cui vi invitiamo a studiare la pagina man), è bene sapere che essi effettuano vari controlli per assicurare una copia il più fedele possibile all'originale, tenendo conto degli algoritmi di correzione d'errore e anche leggendo più volte la stessa traccia. Per questo, se si imposta la modalità più precisa possibile, il processo di estrazione può essere anche decisamente lento. Fate alcune prove con i vostri CD. Una volta estratte le tracce dal CD vi ritroverete dei file .wav con le stesse caratteristiche dello standard CD. Verificatelo con il comando "file":

```
$ file Track1.wav
Track1.wav : RIFF (little-endian) data, WAVE audio,
Microsoft PCM, 16 bit, stereo 44100 Hz
```

Questo è quello che si chiama un file audio non compresso. Per questo tipo di dato il formato **WAV** è uno standard *de facto* ma ci sono anche molti altri

formati per l'audio non compresso (ad esempio **AIFF**, comune nel mondo Macintosh). Allo stream audio aggiunge una intestazione di pochi byte che permette di identificarne frequenza di campionamento, bit resolution, numero di canali, numero di campioni (e quindi durata). Per accedere a queste informazioni più dettagliate si può usare il comando **sndfile-info** della libreria **libsndfile** (pacchetto Debian: **sndfile-programs**). Il problema principale dell'audio non compresso a qualità CD è la sua dimensione ragguardevole: facendo qualche semplice calcolo scoprirete che $44.100 \text{ campioni} * 16 \text{ bit} * 2 \text{ canali} = 1.411.200 \text{ bit per secondo}$, ovvero $1.411.200/8 = 176.400 \text{ byte per secondo}$. Moltiplicando per i 60 secondi di un minuto, otteniamo 10.584.000 byte che equivalgono a circa 10,09 megabyte per un minuto di audio a qualità CD. Per una normale canzone di quattro minuti occorrono quindi ben 40 MB di spazio.

Formati audio lossy

Per ovviare a questo problema sono stati creati degli algoritmi di compressione audio, che si basano sulla percezione psicoacustica del suono, e che consentono di rimuovere parti del suono che l'orecchio umano reputa meno significative. La più famosa di queste codifiche è certamente **MPEG layer 3**, comunemente nota come **MP3** e sviluppata dalla società tedesca Fraunhofer verso gli inizi degli anni 90. Si tratta di un algoritmo di compressione cosiddetto *lossy*, ovvero con perdita di informazione. Ciò significa che convertendo un file Wav in MP3 e riconvertendolo nuovamente in Wav alcune informazioni sonore vengono perse nel passaggio, come avviene per le immagini con la compressione JPEG. La qualità della conversione può essere impostata limitando il bit-rate del file finale: a maggiore bit-rate corrisponde meno perdita d'informazione e quindi maggiore fedeltà. Un bit-rate molto diffuso e considerato un buon compromesso è 128 kbit/s, che permette una compressione della dimensione del file fino a dodici volte (nell'esempio della canzone da 4 minuti



INTERMEDIO

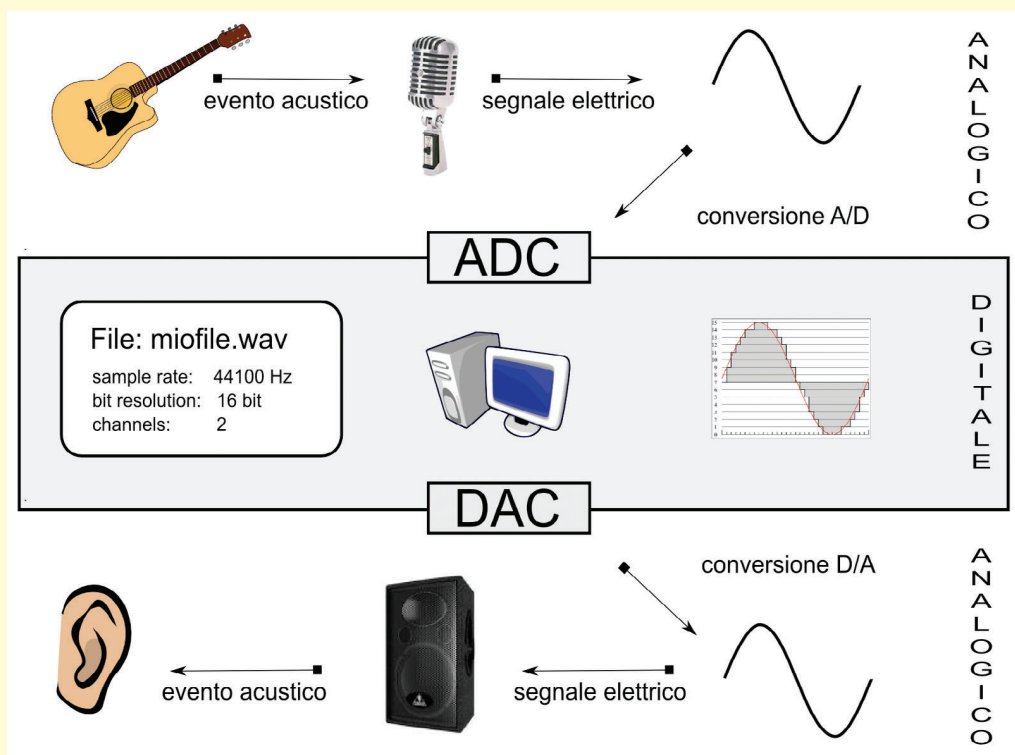
Un po' di teoria

Il segnale audio analogico (per esempio quello che dal vostro iPod arriva alle cuffie tramite un cavo) è un segnale elettrico di bassissima entità, nell'ordine di grandezza dei millivolt, che rappresenta un dato acustico - il suono - in modo elettrico. Per convertire l'evento acustico in uno elettrico e viceversa, si utilizzano dei *trasduttori*: microfoni e altoparlanti. Queste apparecchiature sono in grado di convertire analogicamente l'informazione acustica in informazione elettrica (microfoni) e l'informazione elettrica in informazione acustica (altoparlanti). L'informazione elettrica può essere poi immagazzinata e fissata su supporti di vario tipo, tra cui il più popolare e pratico è stato per anni il nastro magnetico. Ancora oggi vi sono sostenitori della superiorità sonora dell'analogico rispetto al digitale, e in molti studi professionali di registrazione si utilizzano ancora macchine a nastro, anche se la praticità e il continuo miglioramento delle tecnologie digitali stanno soppiantando quasi completamente queste tecnologie e stanno rendendo molto più abbordabili processi di registrazione che in passato erano molto costosi e ad esclusivo appannaggio dell'utenza professionale. La tecnologia dell'audio digitale è stata introdotta sul mercato di massa con l'avvento negli anni 80 del Compact Disc, che andò pian piano a sostituirsi ai dischi in vinile (analogici), che è a tutt'oggi il formato standard di distribuzione di opere musicali e audio in genere. I due componenti elettronici chiave per la registrazione e la riproduzione di segnali audio in modo digitale sono i convertitori: ADC (*Analog to Digital Converter*) e DAC (*Digital to Analog Converter*). Essi sono presenti nella vostra scheda audio, e il loro compito consiste nel convertire un segnale elettrico analogico in una sua rappresentazione digitale e viceversa (si veda il diagramma in questo box). La conversione in digitale comporta la trasformazione del segnale analogico continuo in un flusso

(*stream*) di valori discreti a una velocità prefissata che è detta *frequenza di campionamento*. Questo processo di conversione A/D viene definito appunto *campionamento*. Nel caso del Compact Disc ad esempio, la frequenza di campionamento è pari a 44.100 Hertz. In altre parole, volendo registrare audio a qualità CD, ogni secondo verrebbero valutati 44.100 valori dell'ampiezza istantanea del segnale analogico. Ognuno di questi "campioni" (*samples*) assume un valore compreso in un range di numeri grande come il massimo numero rappresentabile dalla dimensione in bit del singolo campione. Ad esempio, parlando sempre dello standard CD audio, la *bit resolution*, ovvero la dimensione in bit del singolo campione, è di 16 bit (65.536 valori possibili). Altro parametro importante è il numero di canali di cui è composto il segnale audio. Nel caso del CD audio il numero di canali è 2 (stereo), ma potete tranquillamente registrare un segnale mono (1 canale) o con qualsiasi altro numero di canali. Sono questi tre parametri (frequenza di campionamento, bit resolution e numero di canali) le principali caratteristiche che definiscono un flusso audio digitale non compresso. Vi sono anche altre possibili variabili, come il tipo di ordinamento dei bit nel campione (*big-endian*, *little-endian*), che i valori siano con o senza segno (*signed*, *unsigned*), eccetera, ma per semplicità ci atteniamo allo standard CD audio (S16_LE ovvero: signed 16 bit little-endian). Il flusso audio viene letto e salvato nel Compact Disc audio direttamente (secondo uno standard chiamato **Red Book**) e come ben saprete, un CD audio non possiede filesystem, diversamente dal CD-ROM (standard **Yellow Book**). I lettori di CD audio durante la lettura effettuano un complesso meccanismo di correzione degli errori e ridondanza dei dati volto ad assicurare una riproduzione precisa anche in presenza di graffi o sporco sulla superficie del Compact Disc.

Rumore bianco

Il rumore bianco è un particolare tipo di rumore (che nella pratica non esiste) caratterizzato dall'assenza di periodicità e da ampiezza costante su tutto lo spettro di frequenze. Quindi il rumore bianco presenta uno spettro "piatto" su tutto l'intervallo di lunghezze d'onda considerato.



Il percorso di un ipotetico segnale, prima nel dominio analogico, poi convertito in digitale e quindi nuovamente in analogico per il riascolto





1 La homepage del progetto LAME

i vostri 40 MB sarebbero ridotti a meno di 4). Naturalmente la percezione della qualità audio è soggettiva, quindi molti utilizzano anche bit-rate superiori come 192, 256 e 320 kbit/s, che portano approssimativamente a tassi di compressione di 8:1, 6:1 e 4:1 rispetto alla dimensione del file originale.

FLAC: compressione lossless

A questo punto però potrebbe essere più interessante ricorrere ad algoritmi *lossless*, ovvero senza perdita: il più usato tra questi (e soprattutto libero da brevetti) è **FLAC**, acronimo di *Free Lossless Audio Codec*. Esegue una compressione simile a quella effettuata da compressori "generici" come gzip o rar, con la differenza che essendo studiato appositamente per l'audio permette di ottenere fattori di compressione migliori rispetto a gzip o rar. Il tasso di compressione di FLAC non è basato sul bit-rate ma è proporzionale alla quantità di informazione codificata: ad esempio, codificando silenzio si otterrà un file piccolissimo, codificando

rumore bianco a pieno volume un file quasi grande come l'originale. Quindi si può ottenere mediamente un dimezzamento della dimensione del file senza alcuna perdita qualitativa. Un enorme vantaggio per chi vuole archiviare grosse quantità di audio a massima qualità. I file .flac possono essere eseguiti direttamente con la maggior parte dei player, oppure con il tool dedicato **flac123**. Il comando per codificare e decodificare in FLAC si chiama (guarda caso) **flac**. Lo trovate nelle maggiori distribuzioni o tramite la sua homepage: <http://flac.sourceforge.net>.

MP3: problemi legali

Tornando a MP3, bisogna citare il fatto che questo algoritmo è brevettato e quindi in teoria chiunque lo utilizzi in un apparecchio hardware o in un software dovrebbe pagare una royalty al detentore del brevetto, il Fraunhofer Institute. Per questo molte distribuzioni Linux, a cominciare da Red Hat, hanno smesso di distribuire i programmi (o le loro librerie) che usavano tale codifica. I software sono comunque reperibili su Internet, ad esempio quello che è considerato il migliore encoder per i bit-rate medio/alti: **LAME** (<http://lame.sourceforge.net>, vedi Fig. 1). Se la vostra distribuzione non include LAME, sarete costretti a compilarlo dai sorgenti o a trovare repository non ufficiali (come www.debian-multimedia.org per Debian).

Con LAME potete facilmente convertire i file Wav in MP3, con un controllo molto fine sulle proprietà del file risultante. Ad esempio, oltre alla grandezza del bit-rate è possibile decidere se questo valore debba essere costante, medio o variabile. O ancora, in che modo debba essere processata la stereofonia, e molte altre impostazioni di codifica che influenzano la dimensione e la resa sonora del vostro MP3. Ad esempio, per codificare un file a 192 kbit/s in modo "stereo" potete usare il seguente comando:

```
$ lame -ms -b 192 miofile.wav miofile.mp3
```

È inoltre possibile, tramite LAME, operare la conversione anche in senso opposto, da MP3 a Wav, con l'opzione **-decode**:

```
$ lame -decode miofile.mp3 miofile.wav
```

La risposta Open Source: Ogg/Vorbis

Un altro algoritmo di compressione che si è diffuso rapidamente nel mondo del Software Libero è **vorbis**, un codec lossy analogo a MP3, che a differenza di MP3 è libero da brevetti e secondo alcune analisi comparative offre una migliore resa sonora a parità di dimensione del file. Questi file sono chiamati **.ogg**, dal nome del formato contenitore Ogg che può incapsulare audio, video

Script di test formati e bit-rate

```
Con questo script potete rendervi conto delle differenze sonore tra vari formati e bit-rate. Lo script estrae le prime due tracce di un CD audio e le converte. Il processo dura qualche minuto, se volete fermarlo premete Ctrl+C. Strumenti necessari: cdparanoia, lame, oggenc, flac.  
#!/bin/bash  
  
mkdir -p ~/test/original  
cd ~/test/  
mkdir mp3_64 mp3_128 mp3_192  
ogg_3 ogg_6 flac  
  
# estrazione delle prime due tracce del CD  
cd ~/test/original
```

```
cdparanoia -Z -B -- -2  
  
# conversione in vari formati e bit-rate  
ls *.wav | while read l  
do  
  lame -b 64 "$l"  
  ~/test/mp3_64/"${l}.wav.mp3"  
  lame -b 128 "$l"  
  ~/test/mp3_128/"${l}.wav.mp3"  
  lame -b 192 "$l"  
  ~/test/mp3_192/"${l}.wav.mp3"  
  oggenc "$l" -q3 -o  
  ~/test/ogg_3/"${l}.wav.ogg"  
  oggenc "$l" -q6 -o  
  ~/test/ogg_6/"${l}.wav.ogg"  
  flac "$l" -o ~/test/flac/"${l}.wav.flac"  
done
```

(con il codec *theora*) e metadati. Gli strumenti per codificare e decodificare gli Ogg/Vorbis dovrebbero essere presenti in tutte le distribuzioni: **oggenc** per l'encoding da Wav a Ogg e **oggdec** per il processo inverso. Il codec Vorbis è supportato da tutti i maggiori player software per Linux e non. Purtroppo, invece, molti player hardware sono limitati al solo formato MP3. Se avete un player che non supporta gli Ogg siete perciò costretti a usare il formato MP3, se invece il lettore dovete ancora acquistarlo, potrebbe essere una forma di sostegno verso il produttore e verso la diffusione di questo formato comprarne uno che supporta gli Ogg. Per vedere informazioni dettagliate sui file .ogg, utilizzate il comando **ogginfo**.

ffmpeg e MPlayer

Altri formati compressi proprietari come Microsoft ASF (file .wma) possono essere gestiti con il convertitore video **ffmpeg** (<http://ffmpeg.mplayerhq.hu>) che supporta un gran numero di codec (compresi FLAC, MP3 e Ogg/Vorbis). La sintassi è molto semplice:

```
# da wav a wma
ffmpeg -i miofile.wav miofile.wma
```

```
# da wma a ogg
# NB: la conversione tra due formati lossy
è sconsigliata
```

```
ffmpeg -i miofile.wma miofile.ogg
```

Se vi addenterate nella lettura della lunga pagina di manuale di ffmpeg scoprirete uno strumento molto complesso con una quantità enorme di opzioni possibili e di formati supportati (provate ad esempio "ffmpeg -formats"). Un player video strettamente imparentato con ffmpeg che può tornare spesso utile, ad esempio per estrarre l'audio da un filmato, è il celebre **MPlayer** (<http://mplayerhq.hu>). MPlayer supporta praticamente qualsiasi codec conosciuto dall'uomo, e tramite l'opzione "-ao" (audio out) potete redirigere il flusso audio su un file, in questo modo:

```
$ mplayer -ao pcm:file=miofile.wav -vo null xyz.flv
```

Questo comando crea un file audio "miofile.wav" che è una copia (e ha le stesse caratteristiche in termini di sample rate, numero di bit, ecc...) dell'audio del filmato

flash xyz.flv, che potrebbe, per esempio, essere un video musicale scaricato da YouTube. Tutti i codec video e audio che MPlayer è in grado di riprodurre possono essere usati in questo modo, scrivendo l'audio su file invece che sull'uscita della scheda audio.

Usare semplici cicli Bash

Una volta presa dimestichezza con i comandi di conversione è possibile con poche righe di scripting Bash applicare lo stesso processo a tutti i file di una directory. Ad esempio, lo script riportato nel riquadro **Script di test formati e bit-rate** crea una directory "test" nella vostra home, estrae le prime due tracce da un Compact Disc audio ed esegue una serie di conversioni in diversi formati e diversi bit-rate in modo che possiate apprezzare e toccare con mano (o meglio, sentire con le vostre orecchie) le differenze soniche e valutare come queste si rapportano alle differenze di dimensione tra i vari formati e bit-rate. Da notare che con tassi di compressione più bassi (o bit-rate più alti) le differenze tra il file originale o il FLAC e la copia lossy sono appena apprezzabili, e soltanto con un sistema audio dotato di una buona (o addirittura ottima) fedeltà di riproduzione. Ad esempio provate con un amico a effettuare un test in "doppio cieco" e vedete se riuscite a distinguere il Wav originale dalla sua versione MP3 a 192 kbps.

Registrare e riprodurre file audio

Per registrare un file audio non compresso, prendendo come sorgente l'ingresso line oppure mic della scheda audio, si possono usare le utility di base di ALSA: **arecord** e **aplay**, rispettivamente per registrare e per riprodurre file in formato Wav (ma anche .voc e .au). Per registrare è necessario prima impostare la sorgente di cattura e il livello di registrazione. Per farlo utilizzate il mixer standard **alsamixer** (vedi **Fig.2**), tenendo presente che il tasto Tab consente di accedere ai controlli di registrazione. È necessario selezionare la sorgente di cattura (con la barra spaziatrice), togliere eventuali mute (con il tasto "M"), sia dalla sorgente che dal canale "capture" sul mixer, e impostare i livelli (con i tasti freccia su/freccia giù) in modo che il segnale

Prova in doppio cieco

In ambito scientifico, si dice prova in "doppio cieco" quando entrambi gli attori della prova (dottore/paziente in campo medico, ad esempio) non conoscono l'operazione eseguita ma ne vedono solo gli effetti. Questo tipo di prova è molto usata anche in ambito musicale.

I mille volti dei formati MPEG-4

Il formato **MPEG-4 Part 14** è un formato contenitore analogo a Ogg, e deriva dal formato **Quicktime/MOV** di casa Apple. Esso può contenere un numero arbitrario di flussi digitali audio e video, oltre ad altri dati come sottotitoli e immagini statiche. Questi flussi possono usare una serie di codec audio e video diversi, tutti appartenenti al complesso standard MPEG-4. Tra quelli audio i più noti sono AAC e MP3. L'estensione ufficiale di questi file è **.mp4**, ma è ormai di uso comune (specialmente dopo il successo di *iTunes* e *iPod*, che adottano questa convenzione) etichettare i file contenenti unicamente audio con l'estensione **.m4a**. Altre estensioni utilizzate sono **.m4b** per audiolibri e podcast (con la possibilità di inserire segnalibri, immagini

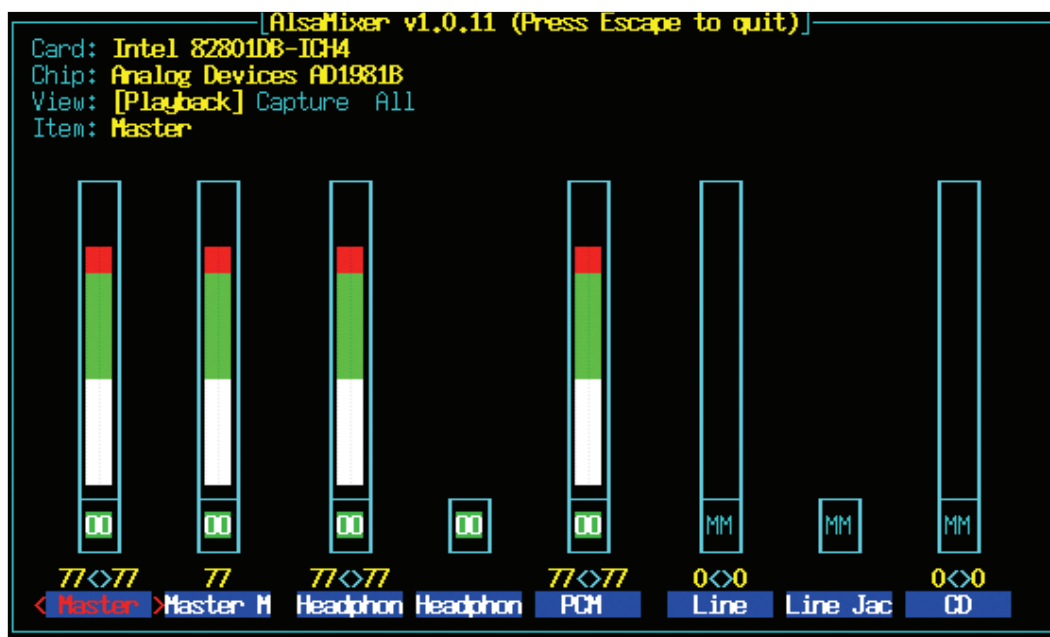
e collegamenti ipertestuali), e spesso **.m4v** per i file video. Oppure ancora, **.m4p** a denotare i file di iTunes con contenuti protetti da DRM. Anche i file con estensione **.3gp** sono una variante di questo formato contenitore, normalmente sono file prodotti dai telefoni cellulari o destinati ad essere riprodotti su cellulari, e possono contenere anche codec non MPEG-4 (ad esempio AMR per l'audio e H.263 per il video). In Linux potete gestire i dati contenuti nei file MPEG-4 con il potente convertitore audio/video **ffmpeg** e con l'altrettanto potente **MPlayer** per la riproduzione. Per esempio è possibile convertire un file audio .m4a in .wav molto semplicemente con ffmpeg:

```
ffmpeg -i miofile.m4a miofile.wav
```



TUTORIAL Conversione audio

2 Ecco alsamixer, il mixer di default di ALSA. Vi permette di gestire le proprietà della vostra scheda audio, come la selezione della sorgente di cattura



entrante non causi distorsioni. Fatto questo con arecord potete impostare le proprietà del nostro Wav con lo switch **-f** (formato), che offre un comodo shortcut "cd" per registrare a qualità CD:

```
$ arecord -f cd poesia_1.wav
```

Quindi si ferma la registrazione premendo la combinazione di tasti **Ctrl+C**. Potete quindi riascoltare la vostra registrazione con aplay:

```
$ aplay poesia_1.wav
```

Con arecord, invece di premere **Ctrl+C** è possibile impostare la durata della registrazione con lo switch **-d** specificando la durata voluta in secondi.

Combinando con il comando **date** potete creare un alias per registrare clip audio da 20 secondi, per esempio per piccole annotazioni vocali:

```
$ alias audioclip='arecord -d 20 -f cd  
$(date +%d%b%Y-%H:%M:%S).wav'
```

Da file a CD audio

Altra operazione che quasi tutti hanno bisogno di compiere almeno una volta nella vita è quella di mettere su CD audio i propri MP3 o Ogg preferiti. Abbiamo visto come con LAME o ffmpeg sia semplice effettuare la conversione da MP3 a Wav (stessa cosa dicasi per Ogg e FLAC), ma prima di scrivere un CD audio dovete assicurarvi che tutti i file che intendete scrivere sul supporto rispecchino le impostazioni di formato richieste dallo standard Red Book, ossia devono essere tutti file a 44.100 Hz di frequenza di campionamento, 16 bit di risoluzione e avere 2 canali. Non sempre gli MP3, gli Ogg e gli altri file che ci capitano sottomano sono già in questo formato (per esempio i file registrati con device come telefonini o palmari), e a volte anche i Wav risultanti dalla decodifica hanno un formato non adatto. Per verificarlo usate sempre il comando file:

```
$ file *.wav
```

Tutto ciò che non risulta "16 bit, stereo 44100 Hz" va convertito in questo formato prima di scrivere le tracce su un Compact Disc vergine. Lo splendido *coltellino svizzero dell'audio* **sox** (contrazione di "sound exchange") vi consente di cambiare le proprietà dei vostri file audio .wav mediante un'operazione di *ricampionamento* per cambiare la frequenza di campionamento e la risoluzione in bit, e mediante una operazione di copia dei canali per passare da mono a stereo, se necessario. Il comando da usare è

```
sox input.wav -c2 -r44100 -s output.wav
```

Anche sox è generalmente pacchettizzato per le maggiori distribuzioni, oppure potete trovarne i sorgenti sul suo sito: <http://sox.sourceforge.net/>. Inoltre sox è in grado di effettuare tutta una serie di altre operazioni interessanti sui file, che potrete andare a scoprire nella sua ricca pagina di manuale. Una volta portate tutte le tracce allo standard CD, potete scriverle sul supporto con la vostra applicazione di masterizzazione preferita (**cdrecord** e i suoi front-end, **K3b**, **gcdmaster**, eccetera). Prima però potreste notare che i volumi delle diverse tracce presentano notevoli differenze tra loro. In questo caso potete usare l'utility **normalize** (<http://normalize.nongnu.org/>) che aggiusta i livelli dei file a un livello fisso specificato dall'utente, massimizzando il volume di ogni traccia. Questa carrellata dovrebbe avervi dato l'idea di cosa è possibile fare con l'audio usando gli strumenti a linea di comando che Linux ci mette a disposizione, soprattutto per quanto riguarda il "batch processing" e le conversioni tra vari formati. Ricordatevi comunque che ogni passaggio a un formato lossy degrada il segnale, e pertanto è sconsigliato convertire da un formato lossy a un altro.

Vi raccomandiamo pertanto di effettuare anche dei controlli "a orecchio" sul materiale con cui lavorate prima di far partire la "madre di tutte le conversioni batch". **LXP**