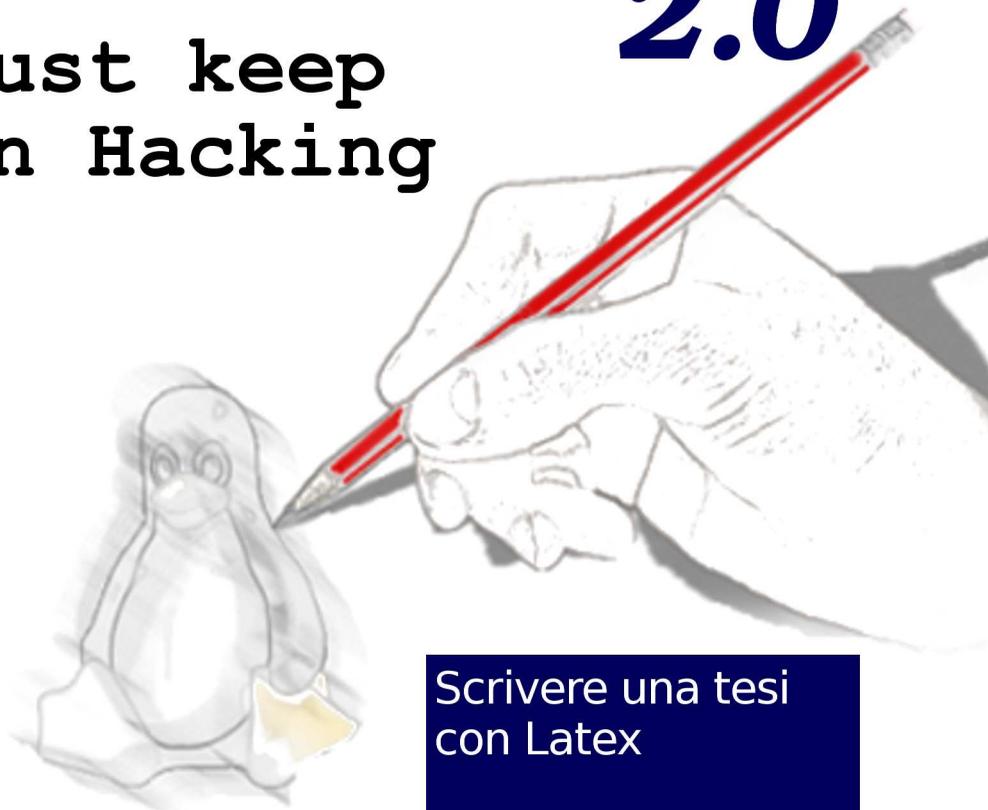


GiornaLinux

2.0

Just keep
on Hacking



Linux



Scrivere una tesi
con Latex

Zend Framework

Preparare una
base musicale con
GNU/Linux

DOSBox: Ritorno al
passato

POuL: un piccolo bilancio

Sante Rotondi <satene.r@gmail.com>

ECCOCI A VOI, cari lettori del Giornale Linux, con il primo numero dopo la meritatissima pausa estiva.

Come non tracciare un bilancio delle mille attività che il POuL ha svolto o ha contribuito a svolgere nell'ultimo anno: quattro GiorniLinux, tre conferenze (Drupal-Day, presentazione Ubuntu 9.04, presentazione Polinux), il workshop sulla sicurezza informatica, la partecipazione al Linux Day di Cinisello Balsamo, le quattro giornate di corsi di GNU/Linux gratuiti, la partecipazione al Synesis Forum con il nostro stand, l'incontro a Palazzo Marino con le redazioni degli altri giornali studenteschi di Milano, la collaborazione con Ingegneria Senza Frontiere, le tantissime assistenze WiFi, la partecipazione alla gara di sicurezza informatica iCTF'08... Tutte queste attività non sarebbero state pensabili senza l'impegno e la passione dei membri del POuL.

Caro studente, aspettiamo solo te! Non serve essere un geek né avere competenze esoteriche, il POuL è aperto a tutti gli studenti che condividano il nostro interesse verso il software libero, la libertà di espressione, e in generale a coloro che ritengano l'università un luogo in cui crescere come persona *facendo* e non limitandosi a guar-

dare un quaderno ed una lavagna.

Nell'arco dell'ultimo anno, personalmente, ho avuto modo di partecipare a molte delle iniziative e ho conosciuto molte persone interessanti, la cui esistenza avevo soltanto supposto annoiandomi in mezzo ai tanti studenti del Politecnico che sembrano frequentare i corsi con lo stesso interesse col quale si seguirebbe una replica di Porta a Porta... Ebbene, persone attive e volenterose esistono, per fortuna si cercano e si trovano tra loro e si divertono un mondo! Abbiamo appena ricominciato ad organizzare conferenze, scrivere e diffondere articoli, e... tu cosa vorresti fare? Passa a trovarci in sede, iscriviti alla nostra mailing list sul sito www.poul.org, leggi i blog dei membri dell'associazione: scoprirai di certo che anche tu hai delle idee e con il POuL potrai metterle in pratica.



Indice

| | |
|---|----|
| Scrivere una tesi con \LaTeX | 3 |
| Zend Framework | 6 |
| Preparare una base musicale con GNU/Linux | 9 |
| DOSBox: Ritorno al Passato | 12 |

Quest'opera è rilasciata sotto la licenza Creative Commons BY-NC-SA 2.5. Questo significa che sei libero di riprodurre, distribuire, comunicare al pubblico, esporre in pubblico, rappresentare, eseguire o recitare l'opera e creare opere derivate alle seguenti condizioni:

-  **Attribuzione.** Devi riconoscere il contributo dell'autore originario.
-  **Non commerciale.** Non puoi usare quest'opera per scopi commerciali.
-  **Condividi allo stesso modo.** Se alteri, trasformi o sviluppi quest'opera, puoi distribuire l'opera risultante solo per mezzo di una licenza identica a questa.

In occasione di ogni atto di riutilizzazione o distribuzione, devi chiarire agli altri i termini della licenza di quest'opera. Se ottieni il permesso dal titolare del diritto d'autore, è possibile rinunciare ad ognuna di queste condizioni.

Le tue utilizzazioni libere e gli altri diritti non sono in nessun modo limitati da quanto sopra.

Questo è un riassunto in linguaggio accessibile a tutti del Codice Legale:

<http://creativecommons.org/licenses/by-nc-sa/2.5/it/legalcode>



Scrivere una tesi con L^AT_EX

Nicola Vitucci <nick.vitucci@yahoo.it>

RIECCOCI QUI, dopo qualche altro numero del Giornalinux, a parlare di LaTeX e delle sue infinite potenzialità.

Se vi ricordate, nel primo articolo abbiamo visto come scrivere un semplice ma professionale articolo scientifico e come aggiungerci un po' di altre cose; stavolta vedremo come si può scrivere qualcosa di un po' più "grosso", come può essere una tesi di laurea oppure un libro. Naturalmente, in rete (anche tra i siti del Politecnico) è possibile trovare molti layout già pronti per essere usati, ma noi ne costruiremo passo passo uno più semplice per capire dove si può andare a "mettere mano" per ottenere risultati migliori.

Un ottimo sito che potete usare come riferimento è <http://www.guit.sssup.it>, del Gruppo Utilizzatori Italiani di TeX.

Impostare il lavoro

Una tesi tipicamente è organizzata in più parti: ha un frontespizio, un sommario, un indice, magari è suddivisa in sezioni ognuna contenente uno o più capitoli, e ha una bibliografia. Ci possono essere anche altri elementi, ma questi sono quelli principali.

Sarebbe quindi scomodo scrivere tutto in un unico file: come si fa con una tesi di un centinaio di pagine?!

Incominciamo quindi impostando il file principale, che chiameremo *tesi.tex*. Come prima riga scriviamo:

```
\documentclass[11pt,a4paper,
twoside,italian,openany]{report}
```

in cui stabiliamo rispettivamente che il font è grande 11 punti, il formato dei fogli è A4, si utilizzano entrambe le facciate di una pagina, il documento è in lingua italiana (opzione utile ad esempio per lasciare indicare a LaTeX i capitoli con "Capitolo" piuttosto che con "Chapter") e i capitoli possono iniziare indifferentemente sulla pagina destra o sinistra. Usiamo la classe `report` perché la tesi è intesa come una relazione, ma potremmo usare anche `book` (per somiglianze e differenze, vi rimando a un manuale di LaTeX). Per quanto riguarda i package da includere, dipende più che altro da cosa si vuole inserire nella tesi: a parte `graphicx` e `subfigure`, utili in ogni caso per la gestione delle figure, per tesi ingegneristiche consiglio i pacchetti `tabularx` per le tabelle, `amsmath` e `amssymb` per simboli matematici, oltre naturalmente a quelli che ho spiegato la scorsa volta e `geometry`, che merita un discorso a parte.

In sostanza, il package `geometry` serve, qualora fosse necessario, per modificare in maniera coerente tutti i parametri di una pagina. Un esempio di utilizzo è

```
\usepackage[a4paper,top=3cm,
  bottom=3cm,left=2cm,right=2cm,
  bindingoffset=2mm]{geometry}
```

Includere altre parti

Fatto tutto ciò, includere altre parti (che per ora supponiamo aver già scritto) è molto semplice: usiamo `\include{nomeparte}` per includere nella tesi tutto ciò che compare nel file `nomeparte.tex`. Possiamo quindi già scrivere subito dopo `\begin{document}`:

```
\include{frontespizio}
\include{sommario}
\include{capitolo1}
```

Più tardi vedremo perché queste righe non sono sufficienti. Intanto apriamo un nuovo file e salviamolo come `frontespizio.tex`. In questo file scriveremo il titolo della tesi, il nome, il numero di matricola ecc. Attenzione ad includere il logo dell'università: ci vuole l'autorizzazione!

Per centrare il contenuto si può usare

```
\begin{center}
  Nome tesi ecc.
\end{center}
```

e si possono poi usare `\large` e `\LARGE` per scrivere in caratteri più grandi ad esempio il nome dell'università e il titolo della tesi, ricordandosi poi di inserire `\normalsize` per riportare il carattere alla dimensione standard; `\vspace{Ncm}` può

servire per spaziare verticalmente gli oggetti di N cm, dove N è un qualsiasi numero decimale; si possono infine allineare gli oggetti sulla destra usando

```
\begin{flushright}
  Bla bla bla
\end{flushright}
```

oppure sulla sinistra usando `flushleft`.

Passiamo ora al sommario: come al solito, apriamo un nuovo file e salviamolo come `sommario.tex`. In questo file scriviamo un breve riassunto della tesi, poi salviamo. Facciamo la stessa cosa per `capitolo1.tex`, in cui scriviamo un ipotetico primo capitolo, magari usando il generatore di testi casuali della scorsa volta. Siamo pronti per compilare la tesi con `pdflatex tesi.tex`; finita la compilazione apriamo `tesi.pdf` e... diciamolo pure: fa schifo!

Innanzitutto notiamo che il frontespizio ha un numero di pagina, ma sicuramente noi non siamo d'accordo: per risolvere questo problema, apriamo di nuovo il file `frontespizio.tex` e come prima riga scriviamo `\thispagestyle{empty}`. Possiamo anche aggiungere un `\clearpage` come ultima riga per assicurarci che la pagina del frontespizio finisca qui. Così facendo, però, il sommario comparirà sul retro della pagina del frontespizio con un numero di pagina, e noi non vogliamo: nel file `tesi.tex` aggiungiamo, subito dopo `\include{frontespizio}`, le righe

```
\thispagestyle{empty}
\cleardoublepage
```

per lasciare una facciata vuota e iniziare sulla pagina destra. Lo stesso dobbiamo fare per il sommario, e volendo possiamo

inserire `\thispagestyle{empty}` nel file `capitolo1.tex` per eliminare il numero di pagina dalla prima pagina del capitolo.

Ora mancano i titoli e l'indice. Nel file `sommario.tex` inseriamo come prima riga `\chapter*{Sommario}` prima del testo: l'asterisco serve a non far comparire il nome di questo capitolo nell'indice. Nel file `capitolo1.tex` invece scriveremo per esempio `\chapter{Introduzione}`. Per creare l'indice, nel file `tesi.tex` dovremo aggiungere la riga `\tableofcontents` nel punto in cui vogliamo inserirlo.

Ora ricompiliamo (per due volte, avendo creato un indice) e rivediamo il risultato: il risultato è decisamente migliore!

Adesso potete sbizzarrirvi a provare queste poche cose per creare la pagina della dedica, la sezione dei ringraziamenti, le appendici e tutto quello che volete. Quando avete finito, passate alla prossima sezione, quella della bibliografia.

Creare la bibliografia

Per creare una bibliografia come si deve, abbiamo un ottimo strumento: BibTeX. Creiamo un nuovo file e chiamiamolo `biblio.bib`. In questo file scriviamo:

```
@INPROCEEDINGS {doc1,
  author = {Tizio Caio},
  title = {Titolo},
  booktitle = {Raccolta},
  volume = {1},
  pages = {1--10},
  year = {2009},
}
```

e salviamo. Nel file `tesi.tex` inseriamo le seguenti righe nel posto in cui vogliamo che

compaia la bibliografia, tipicamente dopo tutti i capitoli e prima delle appendici:

```
\addcontentsline{toc}{chapter}
{Bibliografia}
\bibliographystyle{plain}
\bibliography{biblio}
```

e poi ricompiliamo. Se esaminiamo il documento risultante, non notiamo cambiamenti: dobbiamo infatti prima dare il comando `bibtex tesi.aux`, senza preoccuparci del messaggio di errore, e poi ricompilare ancora due volte. Adesso vediamo che c'è una sezione *Bibliografia* vuota e la voce corrispondente nell'indice; la sezione è vuota perché la fonte bibliografica non è mai stata citata nel documento, e questa è anche la causa dell'errore evidenziato da BibTeX. Per far funzionare tutto correttamente, apriamo il file `capitolo1.tex` e in un punto del testo a piacere inseriamo la riga `\cite{doc1}`. Salviamo, rieseguiamo BibTeX e ricompiliamo due volte: adesso funziona tutto!

Noterete che però le parole chiave del riferimento bibliografico sono in inglese: dobbiamo perciò usare un package in cui le parole sono in italiano, per esempio `babelbib`, inserendo in `tesi.tex` la riga `\usepackage[italian]{babelbib}` e sostituendo `babplain` a `plain` in `\bibliographystyle`; bisogna poi aggiungere la riga `language = {italian}` tra i campi del riferimento bibliografico.

Al solito, rieseguiamo BibTeX e ricompiliamo due volte: ora abbiamo i riferimenti in italiano, magari con alcune parole tradotte in maniera un po' approssimativa. Per i vari tipi di riferimento, e per modificare il file di stile e inserire le parole giuste, vi rimando alla documentazione di BibTeX.

Zend Framework

Massimo Maino <maintux@gmail.com>

QUESTO ARTICOLO vuole essere una guida per tutti coloro che hanno voglia di capire come funzionano tutti quei siti Internet come Facebook, Drupal, Joomla, Wordpress, Wikipedia ecc.

Effettivamente l'espressione "sito Internet" in questa circostanza è decisamente riduttiva: queste sono vere e proprie applicazioni Web-based che offrono all'utente una quantità impressionante di funzionalità. Tale realtà è stata introdotta e si è radicata nella rete grazie al cosiddetto Web 2.0. Ma veniamo al dunque.

Cos'è un framework?

Nella produzione del software, un framework è una struttura di supporto su cui un software può essere organizzato e progettato. Alla base di un framework c'è sempre una serie di librerie di codice utilizzabili con uno o più linguaggi di programmazione, spesso corredate da una serie di strumenti di supporto allo sviluppo del software, come ad esempio un IDE, un debugger, o altri strumenti ideati per aumentare la velocità di sviluppo del prodotto finito.

Lo scopo di un framework è di risparmiar-

re allo sviluppatore la riscrittura di codice già steso in precedenza per compiti simili. Questa circostanza si è presentata sempre più spesso man mano che le interfacce utente sono diventate sempre più complesse, o più in generale man mano che è aumentata la quantità di software con funzionalità secondarie simili.

Linguaggi per il web

Per realizzare applicazioni Web dinamiche ci sono diversi linguaggi, ciascuno con pregi e difetti, i più famosi dei quali sono sicuramente PHP, ASP e JSP. Certamente, tra questi, quello che ha avuto più successo è PHP. PHP (acronimo ricorsivo di PHP Hypertext Preprocessor, preprocessore di ipertesti) è un linguaggio di scripting interpretato, con licenza Open Source e parzialmente libera (ma incompatibile con la GPL), originariamente concepito per la realizzazione di pagine Web dinamiche. Attualmente è utilizzato principalmente per sviluppare applicazioni Web lato server, ma può essere usato anche per generare script a linea di comando o applicazioni standalone con interfaccia grafica. Il motore di PHP è chiamato Zend (dai due sviluppatori Zeev Suraski e Andi Gutmans).

Zend framework

PHP ha riscosso un grande successo prima di tutto per la sua semplicità d'utilizzo nei siti dinamici. Al contrario, per applicazioni di grosse dimensioni, PHP non si è dimostrato così performante, tanto da risultare in queste circostanze macchinoso.

Per questo motivo gli sviluppatori di Zend hanno deciso di creare delle librerie per facilitare la scrittura di grosse applicazioni (tipicamente "Enterprise Applications") ed agevolare la produttività. L'insieme di queste librerie rappresenta il Framework Zend.



Pattern MVC

Il framework, come già detto, mette a disposizione una serie di librerie di supporto alla programmazione. Gli sviluppatori hanno pensato che la soluzione che meglio può aiutare i programmatori è quella di implementare il framework con il pattern MVC (*Model-View-Controller*).

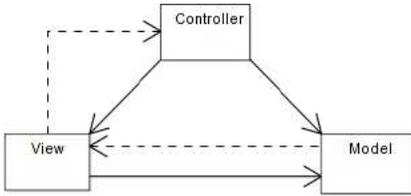
Nell'ingegneria del software, un *design pattern* (struttura di progettazione) può essere definito "una soluzione progettuale generale a un problema ricorrente": non è una libreria o un componente di software riusabile, quanto una descrizione o un modello da applicare per risolvere un problema che può presentarsi in diverse situazioni

durante la progettazione e lo sviluppo del software. La differenza tra un algoritmo e un design pattern è che il primo risolve problemi computazionali, mentre il secondo è legato agli aspetti progettuali del software. Model-View-Controller è un pattern architetturale molto diffuso nello sviluppo di interfacce grafiche di sistemi software object-oriented. Questo pattern è basato sulla separazione dei compiti fra i componenti software che interpretano tre ruoli principali:

- il *model* fornisce i metodi per accedere ai dati utili all'applicazione;
- il *view* visualizza i dati contenuti nel model e si occupa dell'interazione con utenti e agenti;
- il *controller* riceve i comandi dell'utente (in genere attraverso il view) e li attua modificando lo stato degli altri due componenti.

Questo schema, fra l'altro, implica anche la tradizionale separazione fra la logica applicativa (in questo contesto spesso chiamata "logica di business"), a carico del controller e del model, e l'interfaccia utente a carico del view.

I dettagli delle interazioni fra questi tre oggetti software dipendono molto dalle tecnologie usate (linguaggio di programmazione, eventuali librerie, middleware e via dicendo) e dal tipo di applicazione (per esempio se si tratta di un'applicazione Web o di un'applicazione desktop). Quasi sempre la relazione fra view e model è descrivibile anche come istanza del pattern Observer. A volte, quando è necessario cambiare



il comportamento standard dell'applicazione a seconda delle circostanze, il controller implementa anche il pattern Strategy. Viene da sé capire l'importanza di questo approccio allo sviluppo, che permette, in ambito Web, la realizzazione di applicazioni abbastanza complesse; basti pensare alla chat di Facebook o a quella di Google (Gtalk).

Altri framework

Ovviamente, per avere applicazioni complesse che permettano una forte interazione con l'utente si necessita di altre tecnologie oltre a quella server-side fornita da PHP. Nello specifico Javascript, essendo client-side, offre possibilità di natura diversa, come ad esempio eventi in real-time generati sul browser. Anche in questo caso utilizzare Javascript può risultare molto complesso; per questo motivo sono stati sviluppati anche framework per Javascript stesso, tra cui i più famosi sono Dojo, jQuery, Prototype, mooTools.

Personalmente apprezzo molto jQuery, che offre appunto una serie di librerie che facilitano l'implementazione di feature complesse nell'applicazione client-side grazie anche ad AJAX; dal canto suo, AJAX permette una comunicazione asincrona tra il client ed il server, utile - ovviamente combinato

a PHP - per realizzare ad esempio una chat come quella di Facebook.

Risorse

Un ottimo IDE che facilita la scrittura di applicazioni PHP con il framework Zend è Zend Studio basato su Eclipse, che però è a pagamento. In alternativa si posso usare IDE altrettanto validi come ad esempio NetBeans e lo stesso Eclipse, che ovviamente non hanno nativamente il supporto alla creazione di progetti Zend: ciò significa che bisogna creare l'albero delle directory a mano.

Per quanto riguarda la documentazione, esistono una serie di documenti come *Zend in Action* che spiegano come usare il framework; in alternativa, sul sito del framework si trova facilmente tutta la documentazione delle librerie.

Preparare una base musicale con GNU/Linux

Alessandro Sivieri

<alessandro.sivieri@gmail.com>

SE SIETE MUSICISTI o aspiranti tali, vi sarà sicuramente capitato di avere un brano di cui avreste voluto la base per poter provare per conto vostro: anche se la qualità finale di questa non dovesse essere proprio identica ad una suonata dal vivo, sarebbe comunque utile per raggiungere lo scopo. In questo articolo vi presenterò una serie di passi per poter ottenere una base musicale tramite software libero, su sistema operativo GNU/Linux, utilizzando programmi presenti praticamente in tutte le distribuzioni esistenti.

Pro e contro

Partiamo dai pro e contro del procedimento che andiamo a rappresentare: a nostro favore abbiamo la facilità di reperimento dei brani, diciamo da quelli mediamente conosciuti in su, e la possibilità di suonare pezzi anche se non abbiamo una band, potendo scegliere quali parti suonare noi e quali includere nella base; i problemi principali sono dati dalla difficoltà di ottenere

basi di tutte le canzoni esistenti (anche se comunque di molte di esse), e dal fatto che gli strumenti non suonati da noi vengono suonati dal computer, con un andamento che inevitabilmente non potrà che essere più o meno meccanico, a seconda della bravura di chi ha trascritto le note. Ma partiamo con ordine.

Reperire il brano: quale formato?

Il nostro primo passo sarà trovare il brano da noi scelto in formato MIDI, il quale ci permette di manipolare le tracce dei singoli strumenti nonché le note stesse (se abbiamo le conoscenze musicali necessarie); il principale difetto è che non è detto che su Internet, nostra fonte primaria, si trovi effettivamente qualche volenteroso che abbia trascritto al posto nostro tale canzone; naturalmente, se abbiamo un buon orecchio possiamo anche trascriverla noi stessi. Le due fonti principali che vi consiglio sono www.musicrobot.com, motore di ricerca per file MIDI, e www.911tabs.com, motore di ricerca per tablature; in questo

secondo caso, il formato che a noi interessa è quello del noto programma Guitar-Pro, poiché spesso chi scrive basi con questo formato include non solo le tracce delle chitarre (come accade ad esempio con PowerTab), ma anche eventuali tastiere, percussioni e quant'altro. Se possibile, io vi consiglio questo secondo formato piuttosto che il MIDI puro e semplice: la manipolazione delle note risulta infatti semplificata utilizzando i programmi che sto per presentarvi.

Il fatto che vi consigli il formato di Guitar-Pro, infatti, non implica che voi dobbiate acquistare il suddetto programma: esiste un'alternativa libera, chiamata TuxGuitar (www.tuxguitar.com.ar), multipiattaforma, che vi permette di aprire il suddetto formato e di manipolarlo in tutti i modi necessari; qualcuno magari troverà l'interfaccia non alla pari con la versione commerciale, ma per i nostri scopi va più che bene. Una volta aperto il file (potete anche importare i MIDI, anche se è una funzionalità che non brilla in quanto a precisione), l'obiettivo è di individuare ciò che volete sia suonato da voi ed eliminarlo dal file, per rimanere solamente con quanto vorrete che sia il computer a suonare.

Da MIDI a mp3, oggi ecc.

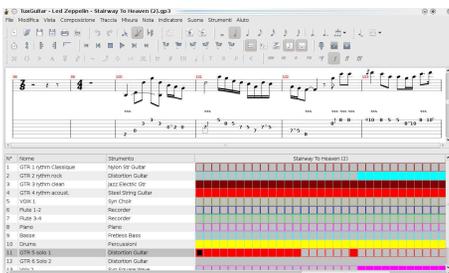
Completato questo passo, potete esportare la base così ottenuta in MIDI. A questo punto, se necessitate di ulteriore manipolazione e volete un programma più ricco per l'editing di questo formato di uscita, potete installare Rosegarden, applicazione appartenente alla suite di KDE, che vi permette di editare la velocità del brano, i volumi delle singole tracce, gli strumenti associati a queste e molti altri dettagli. L'unica cosa cui dovrete a volte stare attenti è: verificate che la traccia di batteria sia associata con il canale numero 10 (contrassegnato dalla lettera 'D'), così che il vostro riproduttore audio riproduca correttamente quello strumento.

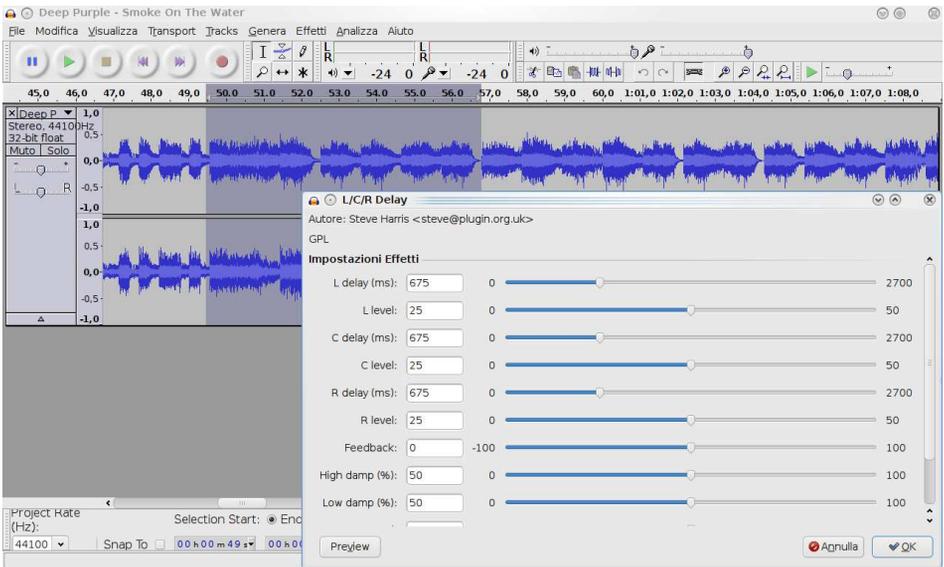
Quando avrete editato il brano in versione definitiva, è ora di sporcarsi un po' le mani: esportate il brano in MIDI (il formato intermedio di Rosegarden ha estensione .rg, che non serve ai nostri scopi), ricordatevi la cartella in cui lo avete salvato, lanciate un terminale e, da quella cartella, lanciate il comando

```
timidity -Ow nomefile.mid
```

Timidity, che avrete sicuramente installato come dipendenza di Rosegarden, vi permette non solo di riprodurre i MIDI stessi, ma anche di trasformarli in file wave. Dopo qualche secondo/minuto (in base a quante tracce ha il brano e alla sua durata) avrete un file dal nome *nomefile.wav*.

L'operazione successiva è aprire Audacity, editor di file wave, ed importare il .wav appena ottenuto: una volta caricato, selezionate l'intero pezzo ed applicate l'effetto Normalize, tenendo buoni i settaggi





di default: avete in pratica appena massimizzato il volume della traccia. Volendo, potete anche applicare un fade out se lo ritenete opportuno, questa volta selezionando per trascinamento gli ultimi secondi ed applicando il relativo effetto.

Al termine delle modifiche, esportate il brano in formato Wave se avete intenzione di masterizzarlo su CD, altrimenti avete a disposizione *mp3*, *ogg*, *flac* ed altri differenti formati, magari per caricare i brani così ottenuti su un lettore musicale. Avviate il brano, aumentate il volume a volontà, impugnate il vostro strumento preferito (dalla chitarra al triangolo) e buon divertimento!

DOSBox: Ritorno al Passato

Alfonso Cancellara

<perchemichiamokinuzzo@yahoo.it>

CONSERVI IN UNA TECA blindata la tua copia originale di Dune II come se fosse il diamante Cullinan? Ritieni che Doom non sia un videogioco, ma un'opera d'arte di fattezza pari all'“Atena Lemnia” di Fidia? E come se non bastasse, passi notti sveglio a pensare a come farli funzionare sulla tua amata Ubuntu? La tua cura all'insonnia si chiama DOSBox!

DOSBox è un perfetto emulatore, rilasciato sotto licenza GNU-GPL, del sistema operativo MS-DOS. Usando questo emulatore è possibile far girare senza (grossi) problemi la maggior parte dei giochi rilasciati per questo vetusto SO sulle moderne distribuzioni GNU-Linux (o, per chi volesse, anche su Windows).

Il seguente tutorial descriverà passo passo come installare e configurare DOSBox. Le istruzioni che seguiranno sono state scritte per chi usa Ubuntu 9.04, tuttavia anche per chi usa versioni differenti non dovrebbero esserci significative differenze.

Una precisazione: nel seguito, i comandi in corsivo si riferiranno ai comandi che vanno inseriti dal terminale di GNU/Linux, quelli in grassetto ai comandi da inserire nella console di DOSBox.

Installazione

Per prima cosa installiamo DOSBox lanciando il solito *sudo apt-get install dosbox*; troveremo DOSBox nel menù “Applicazioni->Giochi”. Ora facciamo partire DOSBox e digitiamo:

keyb it

CONFIG -writeconf dosbox.conf

Verrà creato il file *dosbox.conf* nella cartella */home/<nome_utente>*. Chiudiamo DOSBox, spostiamoci in */home/<nome_utente>* e creiamo una cartella, ad esempio di nome *DRIVEC*, che costituirà per DOSBox la “partizione C:”, cioè l'hard disk del nostro DOS emulato. Ora apriamo il file *dosbox.conf* con un editor di testi, raggiungiamo la riga *usescancodes=true* e modifichiamola in *usescancodes=false*.

In seguito, alla fine del file, dopo la riga *[autoexec]* aggiungiamo le seguenti righe:

keyb it

**mount c /home/<nome_utente>/DRIVEC
C:**

e salviamo il file. DOSBox è configurato: ora vediamo se funziona!

Per conoscere la sintassi completa dei comandi per la shell di DOSBox basta comunque il comando **help /all**. Molti comandi sono simili a quelli per la shell di GNU/Linux, ad esempio **cd ..** per tornare alla cartella precedente oppure il tasto TAB per il completamento automatico dei nomi dei file.



A caccia di demoni

Il gioco che ho scelto per questa prova è naturalmente l'immortale Doom (la versione shareware 1.9 può essere trovata all'indirizzo <http://www.doomworld.com/pageofdoom/shareware.html>). Scaricato il file *doom19s.zip*, lo decomprimiamo in */home/<nome_utente>/DRIVEC*, avviamo DOSBox e digitiamo **DEICE.EXE**. Alla richiesta "Which drive to install to?" digitiamo semplicemente **C**, poi premiamo Invio e poi **Y**. Ora sempre nella console di DOSBox digitiamo **DOOMS_19.EXE** ed attendiamo pazientemente che l'installazione sia finita.

A questo punto digitiamo **SETUP.EXE**: apparirà un menù di configurazione che appunto ci servirà a configurare l'audio di Doom. In questo menù ci muoveremo usando i tasti freccia e selezionando le opzioni con il tasto Invio.

Selezioniamo "Keyboard Only", poi "Sound Blaster" e dopo "240". Ora selezioniamo "Choose Sound FX Card" e poi, in sequenza, "Sound Blaster", "240", "5", "3", "3", poi Premiamo ESC e Invio. Chiudiamo DOSBox, riapriamolo e spostiamoci, utilizzando sempre la console di DOSBox, nella cartella di installazione

del gioco digitando **cd DOOMS**. Ora digitiamo **DOOM.EXE**. A questo punto non resta che esultare con moderazione! Ai più giovani ricordo che per far apparire il menù di gioco basta premere un tasto qualsiasi, ai più anziani dico che se vogliono gustarsi Doom a schermo intero (giusto per riassaporare "i bei vecchi tempi®") basta premere ALT+Invio o modificare nel file *dosbox.conf* la riga *fullscreen=false* nella riga *fullscreen=true*. Per uscire dalla modalità fullscreen serve solo ripremere ALT+Invio.

CD e Floppy disk

In generale, per giocare con DOSbox, basta copiare la cartella con i file del nostro videogioco nella cartella *DRIVEC*, entrare nella cartella del gioco tramite il terminale di DOSBox con il comando **cd** e digitare il nome del file di avvio del gioco (di solito è **NOME_DEL_GIOCO.EXE**, ma non sempre è così). Tuttavia può capitare che il nostro gioco risieda su un CD-ROM oppure su un caro, vecchio floppy disk (quanti ricordi!). Per caricare il nostro gioco da CD-ROM, prima di tutto dobbiamo montare il CD-ROM

su GNU/Linux (di solito Ubuntu lo fa automaticamente).

Consideriamo l'esempio in cui esso sia stato montato in `/media/cdrom`. In DOSBox digitiamo:

```
mount d /media/cdrom -t cdrom
-usecd 0 -ioctl
```

Se tutto è andato bene, saremo in grado di "entrare" nel CD-ROM tramite DOSBox semplicemente digitando **D**:

A questo punto cerchiamo il file `.exe` che fa partire l'installazione del gioco (di solito è **SETUP.EXE**) e, se durante questa installazione imponiamo "C:" come partizione di destinazione, potremo accedere alla cartella del gioco digitando

```
C:
cd cartella_del_gioco
```

Discorso analogo vale per il floppy (consideriamo l'esempio in cui esso sia stato montato da Ubuntu su `/mnt/floppy`):

```
mount a /mnt/floppy -t floppy
```

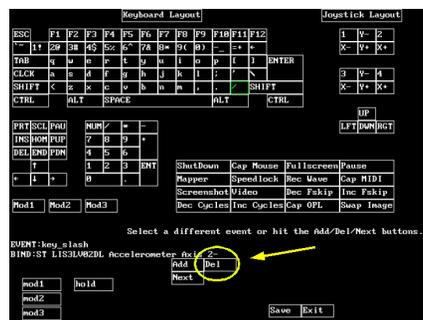
Approfondimenti

Per gli "anglofobi" è disponibile all'indirizzo <http://www.dosbox.com/download.php?main=1> nella sezione "Translations" un archivio contenente il file README di DOSBox in italiano più il file e le istruzioni per tradurre DOSBox stesso in italiano. Per gli "shellofobi" sono invece disponibili numerose interfacce grafiche per DOSBox; per queste rimando alla pagina <http://www.dosbox.com/download>.

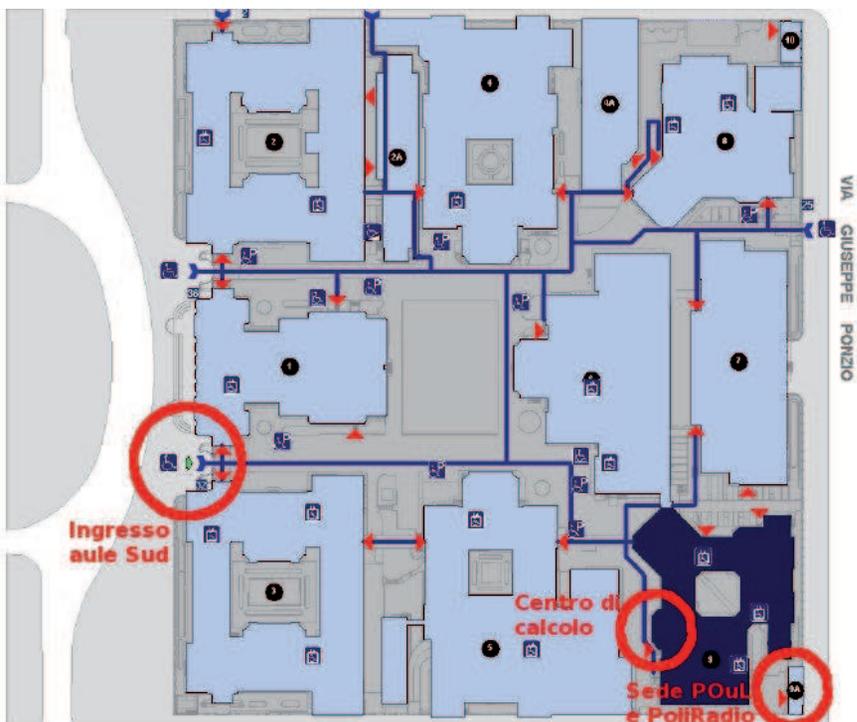
`php?main=1` sezione "Frontends".

Per gli "avventuregrafiche-ofili", invece, consiglio di lasciar perdere DOSBox e provare SCUMMVM (che è presente nei repository Universe di Ubuntu).

Un piccolo avvertimento: possono crearsi dei problemi nel digitare il carattere "-". Per risolvere ciò, basta premere CTRL+F1, selezionare con il mouse la casella "/" sull'immagine della tastiera che apparirà sullo schermo, poi cliccare sulla casella "DEL" in fondo alla schermata (nella figura qui sotto viene messa in evidenza la posizione di "DEL"), cliccare su "ADD" di fianco a "DEL", premere il tasto "-" sulla tastiera del proprio PC, cliccare sulla casella "SAVE" e poi su "EXIT". Quando digiteremo nel terminale di DOSBox il tasto "-" è probabile che sullo schermo vi sia ancora un problema, ovvero che appaiano più caratteri al posto del solo "-". Per eliminare i caratteri in eccesso, basta spostarsi con il tasto FRECCIA SINISTRA della propria tastiera sul carattere "-" e premere il tasto BACKSPACE (il tasto sopra l'INVIO).



Ora che sapete come "resuscitare" i vostri vecchi ma gloriosi videogames... chiudete Giornalinux e tornate a studiare! :-P



Vi è venuta voglia di conoscere il mondo di Linux? Volete partecipare più da vicino alle nostre attività? Volete scrivere un articolo su questa rivista? Iscrivetevi alla nostra mailing list oppure venite a trovarci presso la nostra sede!

sito Internet: www.poul.org
informazioni: info@poul.org
numero sede: +39 02 2399 2477



La stampa della rivista è interamente finanziata dal Politecnico di Milano, che non si assume alcuna responsabilità sul contenuto.
Stampa a cura di *Acheias* di S. Siragusa, Milano 2009.