Howtos-with-LinuxDoc-mini-HOWTO

David S. Lawyer

v0.01, Marzo 2001

Questo testo tratta di come scrivere gli HOWTO usando il marcatore (markup) semplice di LinuxDoc. È rivolto principalmente agli autori di LDP (Linux Documentation project) (e autori futuri che vorranno iniziare rapidamente). Se si vuole usare DocBook, il marcatore più avanzato, (che include XML) vedere la Guida Autorevole di LDP. Traduzione ed adattamenti in italiano a cura di Hugh Hartmann `hhartmann@libero.it`, revisione a cura di Giulio Daprelà `giulio@pluto.it`

Contents

Se si può rispondere affermativamente a queste domande, allora si è incoraggiati a scrivere qualcosa per LDP. Ma bisogna essere consapevoli che tutto questo ((lavoro)) potrebbe prendere molto più tempo di quello che ci si aspetta.

1.2 Copyright (saltare questo se si ha fretta)

Copyright (c) 2001 by David S. Lawyer. Please freely copy and distribute (sell or give away) this document in any format. Send any corrections and comments to the document maintainer. You may create a derivative work and distribute it provided that you:

1. If it's not a translation: Email a copy of your derivative work to the LDP (Linux Documentation Project) for free distribution on the Internet in a format LDP accepts. Also email such a copy to the author(s) and maintainer (could be the same person).
2. License the derivative work in the spirit of this license or use GPL. Include a copyright notice and at least a pointer to the license used.
3. Give due credit to previous authors and major contributors.

1.3 Perché ho scritto questo documento

Perchh́o scritto questo documento quando c'è già una "Guida Autorevole di LDP"? La "LDP Authoring Guide" è un lavoro lungo e dettagliato. Se si vuole iniziare rapidamente, è necessario qualcosa di più semplice e breve. Inoltre, la guida di LDP evita anche di menzionare la semplicità di LinuxDoc. È necessario dire altro?

Un particolare ringraziamento va a Matt Welsh per il suo file example.sgml, che ho usato come sorgente principale di informazioni per le sezioni di esempio.

2 Il formato degli HOWTO

2.1 Introduzione

Gli HOWTO sono rilasciati in vari formati: Testo semplice (txt), HTML, PostScript, e PDF. Invece di dover scrivere lo stesso HOWTO in tutti questi formati si scrive solamente un HOWTO in un formato "sorgente", che viene convertito dal computer in tutti gli altri.

Per avere un'idea di come appaia questo formato basta guardare il file sorgente di una pagina web (se non lo si è già fatto). Si vedranno tante parole brevi racchiuse tra due <partentesi angolari> queste parole sono chiamate "tag". Queste pagine web (tag e tutto il resto) sono in html: *Hypertext Markup Language*. L'LDP usa qualcosa di simile per i propri documenti.

I linguaggi che usa l'LDP soddisfano i requisiti dello *Standard Generalized Markup Language* o, più sinteticamente *SGML*. Attualmente LDP usa le due seguenti varianti di sgml: *LinuxDoc* e *DocBook*. In origine si usava solamente il LinuxDoc (che qualche volta era chiamato inappropriatamente sgml). È interessante notare che html finisce per essere una variante di sgml. Ora alcune persone sono passate dalla variante DocBook di sgml alla variante DocBook di XML standard.

Questo mini-HOWTO è completamente orientato all'uso di LinuxDoc, una variante semplice di sgml. Questa variante si può chiamare "Marcatore LinuxDoc". Esso può essere convertito automaticamente in html, testo semplice, postscript, pdf, e DocBook. Esso è un po' più facile rispetto all'html o al DocBook e non è necessario disporre di un editor speciale per il LinuxDoc in quanto è facile scrivere tra i tag (o usare le macro per loro).

3 Comparazione tra LinuxDoc e DocBook

Un modo per comparare il LinuxDoc con il DocBook è quello di andare al sito di LDP , fare clic sugli HOWTO e confrontare i sorgenti dello stesso HOWTO nei due formati: LinuxDoc e DocBook.

I tag del DocBook sono spesso più lunghi rispetto ai tag equivalenti del LinuxDoc e, qualche volta, sono necessari molti di questi tag per svolgere lo stesso compito rispetto al LinuxDoc. Il DocBook usa i tag <para> e </para> per racchiudere ogni paragrafo mentre il LinuxDoc usa solo un linea vuota per separare un paragrafro dell'altro (non è necessario alcun tag).

Per enfatizzare la parola "release" il confronto è:

```
<emphasis>release</emphasis> release

<em>release</em>
```

Sì, in DocBook bisogna scrivere release due volte. Per iniziare una sezione:

```
<sect>
<title>Introduzione<title>

<sect>Introduzione
```

Così, usando il DocBook c'è molto più da scrivere se si stanno scrivendo i tag manualmente. Ma il DocBook ha tutta una varietà di tag che non esistono nel LinuxDoc e che lo rendono così più avanzato. Usare semplicemente un sottoinsieme di DocBook non aiuta, come si può notare dagli esempi precedenti. C'è ancora una gran confusione di tag. Con un numero maggiore di tag e anche più lunghi il documento diventa difficile da leggere a meno che non si usi un editor che nasconda i tag. Ma nasconderli ha i suoi svantaggi, poiché è piacevole vedere quali tag sono stati usati.

Tuttavia, il numero delle persone che usano il DocBook eccede di molto il numero di quelle che usano il LinuxDoc. Ma, se si decide di migrare al DocBook, esiste un programma di Reuben Thomas (ld2db) che può aiutare a fare la conversione. Tale programma non è perfetto al 100% e, probabilmente, si dovranno apportare alcune correzioni sul documento manualmente con un editor. Anche l'LDP converte automaticamente un HOWTO da LinuxDoc a DocBook dopo che è stato sottoposto a LDP.

4 Imparare il LinuxDoc

4.1 Introduzione

LinuxDoc è un pò più facile da imparare rispetto al DocBook. Ma molto di ciò che si impara riguardo il LinuxDoc sarà utile anche per il DocBook. Così, se eventualmente si decidesse di usare il DocBook, la maggior parte dello sforzo speso per l'apprendimento di LinuxDoc, non sarà sprecato.

Un modo per imparare il LinuxDoc è attraverso gli esempi. Ho scritto tre file d'esempio che spaziano dal facile all'intermedio. Si inizia con l'esempio 1. Questi file sono inclusi qui come sezioni. Per trasformarli in file individuali si può tagliarli dal testo e scriverli ciascuno in un file. Poi si può tentare di trasformarne uno in un testo semplice usando ad esempio ”sgml2txt -f example2.sgml” per vedere come appare. Assicurarsi che i nomi dei file finiscano in .sgml.

Se si desidera vedere alcuni esempi reali si può andare su di un sito mirror di LDP (Linux Documentation Project), trovare gli HOWTO e selezionare LinuxDoc SGML. Oppure si può andare direttamente al sito principale:

Ora il primo semplice esempio.

4.2 Esempio 1

```
<!doctype linuxdoc system>
<article>
<title>Primo Esempio (esempio1)
<author>David S.Lawyer

<sect> Introduzione
<p>Questo è un esempio molto facile di "sorgente" per il sistema
di formattazione testi LinuxDoc. Il paragrafo inizia con tag del
paragrafo (una "p" racchiusa tra due parentesi angolari). Notare che
ci sono altri tag, anch'essi racchiusi tra parentesi angolari.
Se non si vede nessun tag, allora si sta leggendo un file
convertito, perciò si cerchi il file sorgente: example1.sgml
(che contiene i tag).

Questo è il paragrafo successivo. Notare che esso è
separato dal paragrafo precedente solo da una linea vuota.
Pertanto non è necessario mettere un tag "p" all'inizio di
esso. Il tag "p" è necessario solamente per il primo paragrafo
di una sezione (appena dopo il tag di sezione <sect>). Il suffisso
sgml significa Standard Generalized Markup Language. Ora si sta
leggendo la variante LinuxDoc di sgml come specificato nella
primissima linea di questo file.

<sect> I Tag
<p>I tag sono qualsiasi cosa all'interno delle parentesi angolari.
Il tag "sect" sopra marca l'inizio di una nuova sezione di questo
```

```
documento di esempio. "Introduzione" era la prima sezione e ora si sta
leggendo la seconda sezione intitolata "I Tag". Se questo fosse un
documento lungo (come ad esempio un libro), una sezione dovrebbe
corrispondere a un capitolo.

Notare che, all'inizio di questo articolo, ci sono i tag "article",
"title" (titolo) e "author"(autore). Alla fine dei questo articolo
c'è un tag "/article", che segna la fine di questo articolo.
Così c'è una coppia di tag "article", il primo stabilisce
il tag iniziale e il secondo stabilisce il tag finale. Quindi l'intero
articolo è racchiuso nei tag "article". Negli esempi successivi, si
vedrà che ci sono altri tag che vanno in coppia come questi. Essi
hanno effetto su tutto ciò che si trova tra la coppia (tag iniziale
e tag finale). Ogni nome di tag che ha il simbolo "/" appena prima di
esso è un tag di chiusura.

Quando questo codice sorgente è convertito in un altro formato
(come un testo semplice usando il programma sgml2txt) i tag sono rimossi.
I tag aiutano solamente il programma sgml2txt a fare la conversione.
Ci sono molti altri tag da imparare. Così, quando si ha compreso
questo esempio1, si può passare all'esempio successivo: esempio2.
In questo momento non è necessario memorizzare i tag, poiché
saranno ripetuti (ma con poca spiegazione oppure senza spiegazione)
negli esempi successivi.
</article>
```

4.3 Esempio 2

```
<!-- Questo ?un commento. E' ignorato quando questo file sorgente viene
convertito in altri formati. -->
<!-- Il tag successivo richiesto implica che questo file ?nel
formato LinuxDoc -->
<!doctype linuxdoc system>

<article>

<title>Secondo Esempio (esempio2)
<author>David S. Lawyer
<date>v1.0, Luglio 2000

<abstract>
Questo è il sommario. Questo documento è il secondo esempio
sull'uso della variante Linuxdoc-SGML di sgml. Esso è più
complesso rispetto al primo esempio (esempio1.sgml), ma più
semplice del terzo esempio (esempio3.sgml).
Dopo che si ha assimilato questo si sarà in grado di
scrivere un semplice HOWTO usando il LinuxDoc. Fine del sommario.
</abstract>
```

```
<!-- Il "toc" = Table of Contents (tavola dei contenuti). Esso
sarà creato qui. -->
<toc>

<!-- Qui inizia la parte principale dell'articolo (o documento). La
parte precedente ?una specie di lunga intestazione. -->

<sect>Questo è il Secondo Esempio (esempio2.sgml)

<p>A meno che non si abbia preso confidenza con i linguaggi a
marcatori, si dovrebbe leggere prima l'esempio1. Si potrebbe volere
eseguire questi file di esempio attraverso un traduttore come sgml2txt
per convertirli in testo e notare come il risultato appaia differente
ripetto a questo documento "sorgente" con tutti i suoi tag.

<sect>Impaginazione dell'articolo
<sect1>Intestazione del Documento
<p>Un modo per creare una parte di intestazione è di
copiarla da un altro file sgml. Poi basta sostituire ogni cosa con le
informazioni corrette per il proprio documento eccetto i tag.
Questo è come usare un modello.

<sect1> Corpo del Documento
<p>Dopo l'intestazione si arriva al corpo del documento, che consiste
di sezioni annidate marcate dai tag di sezione ("sect"). Le
sottosezioni sono marcate con i tag "sect1". Poiché questa è
la prima sottosezione all'interno della seconda sezione principale, essa
è la sezione 2.1. All'interno di una sottosezione ci
può essere una sotto-sotto-sezione marcata con un tag "sect2", ecc.
Per una sotto-sotto-sotto-sezione usare il tag "sect3". Ci sono anche tag
come "sect4" e "sect5", ma è improbabile che sia necessario usarli.

<sect2> Questa è una sotto-sotto-sezione
<p>
È la sezione 2.2.1. Notare che il tag "p" può essere su
di una linea da solo. Questo non cambia nulla nel risultato del documento.

<sect>Ulteriori caratteristiche nell'esempio3
<p>Con i tag di questo esempio2, si può scrivere un semplice breve
documento lungo qualche pagina. Ma per documenti più lunghi o per
altre importanti caratteristiche come l'inserimento di link all'interno
del documento, è necessario studiare l'esempio successivo: esempio3.
In questo esempio si mostrerà come creare elenchi e font.
</article>
```

4.4 Terzo Esempio

```
<!doctype linuxdoc system>
<!-- Notare il mailto: dopo il mio nome. Questo permette al lettore
di cliccare su di esso per inviarmi una email -->

<article>
<title>Terzo Esempio (esempio3)
<author>David S. Lawyer <url url="mailto:dave@lafn.org">
<date>v1.0, Luglio 2000
<abstract>
Questo documento è il terzo esempio sull'uso della variante
LinuxDoc di sgml. Esso è più complesso rispetto al
secondo esempio.
</abstract>
<!-- Commento: toc = Tavola dei Contenuti  -->
<toc>

<sect>I Font
<p>
Mentre nella conversione in un testo semplice i font non appariranno,
essi funzioneranno per altre conversioni.

<bf>boldface font</bf>     <em>emphasis font</em>
<sf>sans serif</sf>        <sl>slanted font</sl>
<tt>typewriter font</tt>   <it>italics font</it>

C'è un altro modo per ottenere alcuni di questi font
racchiudendo il testo con il simbolo "/" (slash) come questi:

<bf/boldface font/    <em/emphasis font/      <sf/sans serif/
<sl/slanted font/     <tt/typewriter font/    <it/italics font/

<sect>I Link <label id="links_">
<p>Si possono creare link (qualcosa su cui si può cliccare con un
navigatore html per andare in qualche altra parte).
I link possono mandare in un'altra parte di questo documento
(riferimenti-incrociati), o potrebbero mandare a un sito web in Internet.

<sect1>Riferimenti Incrociati
<p> Se si clicca su <ref id="links_" name="Links"> ci si
sposterà all'inizio della sezione precedente "I Link" (che è
ettichettata con links_). L'etichetta dell'id può essere
qualsiasi parola che si desidera, ma è una buona idea evitare
parole comuni, così che si possano cercare etichette uniche
usando il proprio editor. Questa è la ragione per cui uso links_
(con il simbolo di sottolineatura ). Il nome di questo link sarà
mostrato (nel formato html) come il nome su cui fare clic. Questo nome
```

```
(Links) sarà anche presente nella trasformazione del testo.

<sect1>Link URL
<p> Se si clicca su <url url="http://www.tldp.org"> si otterrà
il sito web del Linux Documentation Project. Il successivo link aggiunge
un nome su cui gli utenti potranno cliccare:
<url url="http://www.tldp.org" name="Linux Documentation Project">.
Usando questo secondo metodo, non è necessario spiegare dove
porta il link, in quanto è ovvio dal nome.

<sect>Caratteri Proibiti
<p>Ogni cosa che si scriverà tra le parentesi angolari
sarà interpretata come un tag. Ma se si vuole mostrare un tag in
un documento? Per questo si deve usare un carattere codificato tra
le parentesi angolari.

Si può usare &lt per il simbolo < e &gt per il simbolo >.
lt = Less Than (Minore), gt = Greater Than (Maggiore). Per esempio,
questo è un tag p: &lt;p&gt;. Certamente esso
non inizia alcun paragrafo, ma apparirà nei documenti convertiti
come un <p>. Tutti questi codici iniziano con un carrattere &. Il
carattere ; posto dopo di esso serve per separarlo. Non è necessario
se c'è uno spazio dopo di esso. Per esempio: 3 &lt 4.
Effettivamente, se si sapesse che va bene usare dei > disaccoppiati si
potrebbe scrivere <p> come &lt;p>. Questo non sarebbe riconosciuto
erroneamente come tag, poiché non c'è l'apertura <.

Ci sono altri caratteri che non si possono mettere direttamente nel testo
del documento. Per il carattere & in un comando AT del modem usare:
AT&amp;. Se altri caratteri provocano delle difficoltà
vedere l'esempio 3 o la "guida" che viene fornita nel pacchetto
linuxdoc-tools o sgml-tools.

<sect>Verbatim, Code & Newline

<sect1>Verbatim
<p> Se si vuole essere sicuri che le linee non siano congiunte insieme
durante la conversione ad altri formati, usare il verbatim (tag verb).
Alcune cose vengono riconosciute anche se si trovano tra tag verbatim.
Questo include le macro che iniziano con un carattere & e i tag finali
con il carattere /.

<tscreen><verb>
% sgml2txt -f example.sgml
</verb></tscreen>
Il tag "tscreen" imposta il font a typewriter e lo indenta in modo carino.

<sect1>Code
```

```
<p>Questo racchiude codice del computer tra due linee.
<tscreen><code>
Mettere qui il codice sorgente del computer
</code></tscreen>

<sect1>Newline (A capo)
<p>Per forzare una nuova linea usare il tag <newline>
Questa frase inizierà sempre al margine sinistro.

<sect>Elenchi
<p>Questo mette degli elementi dentro un elenco con un marcatore
all’inizio di ogni elemento.
Le liste iniziano con un tag "itemize".
<itemize>
<item> Questo è il primo elemento in un elenco.
<item> Questo è il secondo elemento
       <itemize>
       <item> Sono supportati livelli multipli (nidificazione).
       <item> Il secondo elemento in questo sottoelenco
       </itemize>
       <enum>
       <item> Si possono avere anche elenchi numerati usando <tt/enum/.
       <item> Questo è l’elemento numero 2
       </enum>
<item> Questo  l’elemento finale nell’elenco principale
</itemize>
</article>
```

4.5 Guida di consultazione rapida di Linuxdoc

4.5.1 Intestazione

```
<!doctype linuxdoc system>
<article>
<title>Guida di consultazione rapida
<author>David S. Lawyer
<date>v1.0, July 2000
<abstract> Qui c’è il sommario </abstract>
<toc> <!-- Commento: toc = Indice  -->
```

4.5.2 Impaginazione del Corpo

```
<sect> Capitolo 1           Nota: Mettere un <p> sulla prima linea di
<sect1> Sottosezione 1.1    ogni sezione (o sottosezione, etc.)
<sect1> Sottosezione 1.2
<sect> Capitolo 2           Scegliere i nomi dei titoli da sostituire a
<sect1> Sottosezione 2.1      "Capitolo" "Sottosezione", ecc.
```

```
<sect2> Sotto-sottosezione 2.1.1
<sect2> Sotto-sottosezione 2.1.2
<sect1> Sottosezione 2.2
</article>
```

4.5.3 Font

Per ottenere i font ci sono due modi:

```
<bf>boldface font</bf>        <em>emphasis font</em>
<sf>sans serif font</sf>      <sl>slanted font</sl>
<tt>typewriter font</tt>      <it>italics font</it>
<bf/boldface/      <em/emphasis/      <sf/sans serif/
<sl/slanted/       <tt/typewriter/    <it/italics/
```

4.5.4 Elenchi (va bene l'annidamento)

```
Liste ordinarie non numerate:           Liste numerate:
<itemize>                               <enum>
<item> Primo elemento                   <item> Primo elemento
<item> Secondo elemento                 <item> Secondo elemento
<item> ecc.                             <item> ecc.
</itemize>                              </enum>
```

4.5.5 Link

```
(Riferimenti incrociati):               Un Link di Email:
<ref id="links_" name="Link">      <url url="mailto:bob@linuxdoc.org">
```

4.5.6 Newline, Verbatim, URL

```
Per forzare una nuova linea <newline>
<tscreen><verb>
<url url="http://www.linuxdoc.org">
<url url="http://www.linuxdoc.org" name="Linux Documentation Project">.
</verb></tscreen>
```

4.5.7 Codici di Caratteri (macro).

- Usare `&` per la e commerciale (&),
- Usare `<` per un simbolo di minore (<),
- Usare `>` per un simbolo di maggiore (>),
- Usare `&etago;` per un simbolo di minore con una barra (</)

- Usare `$` per il simbolo del dollaro ($),
- Usare `#` per un cancelletto (#),
- Usare `%` per il sibolo percentuale (%),
- Usare `˜` per il simbolo tilde (˜),
- Usare ``` `` ``` and `''` per le virgolette singole, o usare `&dquot;` per `"`.
- Usare `­` per un trattino (ovvero, un'indicazione che questo è un buon punto per spezzare una parola per una giustificazione orizzontale).

5 Ottenere/Usare il Software LinuxDoc

Si potrebbe scrivere un documento in LinuxDoc senza avere alcun software LinuxDoc. Tuttavia è assai probabile che esso possa contenere alcuni errori nei tag (o nel loro uso), così che verrebbe restituito per le correzioni. Anche se non ci fossero errori, il risultato probabilmente non apparirebbe completamente giusto. Così è: meglio avere sul proprio computer il software per convertire il proprio file sorgente (sgml).

La distribuzione di Linux Debian ha un pacchetto linuxdoc-tools. Esiste anche un pacchetto rpm per distribuzioni diverse dalla Debian. Questo pacchetto precedentemente si chiamava sgml-tools e la versione 1.0.9 (quella finale) potrebbe essere ancora utile se non si riesce a trovare linuxdoc-tools. Non si deve usare il pacchetto sgmltools-2 perché è destinato principalmente per il DocBook.

Per usarlo, si devono eseguire i programmi di conversione sui file *.sgml. Per esempio per ottenere un file di testo digitare: "sgml2txt -f mio-HOWTO.sgml". Per ottenere html digitare: "sgml2html mio-HOWTO.sgml". Se appaiono degli errori questi mostreranno il numero della linea e della colonna dove si trova l'errore nel file sorgente. Digitando "man -k sgml" dovrebbe apparire un certo numero di altri programmi con una descrizione di una linea per ognuno.

6 Scrivere un HOWTO

6.1 Prima di iniziare a scrivere

Prima inviare una e-mail al coordinatore degli HOWTO al o . Se si desidera rilevare un HOWTO non più mantenuto contattare il vecchio autore. Questo potrebbe essere richiesto dalla licenza di copyright, ma si dovrebbe comunque fare per cortesia, anche quando non è richiesto.

6.2 Linee Guida

Queste sono le linee guida, dovute per la maggior parte a Tim Bynum (un vecchio coordinatore di HOWTO).

- Assicurarsi di usare un formato accettato (come il LinuxDoc :-).

- Tentare di usare una struttura e un'organizzazione significativa, e scrivere chiaramente. Ricordare che molte delle persone che leggono gli HOWTO non parlano l'inglese come loro prima lingua.
- Assicurarsi che tutte le informazioni siano corrette. Non insisterò mai abbastanza su questo. Nel dubbio, fare delle congetture, ma si faccia chiarezza sul fatto che si sta solamente ipotizzando. Io uso ?? se non sono sicuro.
- Assicurarsi di trattare la più recente versione del software disponibile.
- Considerare di includere una sezione di "FAQ". Potrebbero essere chiamate anche "Problemi Comuni" o "Trouble Shooting".
- Assicurarsi di mettere nel copyright il proprio nome e includere una licenza che soddisfi i requisiti dichiarati nel manifesto di LDP.
- Usare l'intestazione standard con il titolo, l'autore, la data (includendo un numero di versione). Vedere 4.4 (Terzo Esempio)
- Per finire prepararsi a ricevere e-mail di domande e commenti dai lettori. Sta alla propria decisione quanto aiutare la gente ma si dovrebbero usare i buoni suggerimenti e le segnalazioni di errori. Si potrebbero anche ricevere delle email di "grazie", così come mail da persone che chiedono aiuto e che non hanno mai nemmeno guardato al proprio HOWTO.

6.3 Sottoporre l'HOWTO, etc.

Dopo aver scritto l'HOWTO, inviare via email il sorgente SGML all'indirizzo: submit@linuxdoc.org. Poi tutto ciò che è necessario fare è tenere aggiornato l'HOWTO sottoponendo periodicamente gli aggiornamenti allo stesso indirizzo di e-mail usato per la prima edizione del proprio HOWTO.

7 Ulteriori informazioni

È in arrivo un prossimo HOWTO riguardo il LinuxDoc che lo tratterà in maniera molto più dettagliata rispetto a questo mini-HOWTO. Verrà rilasciato più o meno in aprile 2001 ??