HOWTO HOWTO

Mark F. Komarinski

v0.13, 19 settembre 1999

Aiutare un nuovo autore LDP a cominciare con strumenti, idee e convenzioni usate da LDP. Traduzione di Mariani Dario <*darkpand@uni.net*>.

Indice

1 Introduzione

1.1 Cronologia

Questo documento è stato iniziato il 26 Agosto 1999 da Mark F. Komarinski *markk@cgipc.com* dopo due giorni di frustrazione nel cercare di far funzionare gli strumenti necessari. Se almeno un autore LDP viene aiutato da questo documento, ho fatto il mio lavoro.

1.2 Nuove versioni

La versione più recente di questo documento può essere trovata sulla mia homepage *http://www.cgipc.com/~markk* `<http://www.cgipc.com/~markk>` nel suo sorgente SGML. Altre versioni possono essere trovate in altri formati all'homepage di LDP *http://www.linuxdoc.org/*.

1.2.1 Cronologia delle versioni

v0.25, 20 settembre 1999

- Corretti alcuni collegamenti errati
- Aggiunta la sezione per i nuovi autori
- Vari cambiamenti rispetto alla prima versione pubblica
- Dichiarazione di copyright ora inclusa al posto di una URL

v0.12, 2 settembre 1999

- Completate la maggior parte delle sezioni
- Integrati dei cambiamenti dalla lista ldp-discuss

v0.10, 27 agosto 1999

- Scritto fino alla sezione 3.4
- Aggiunto qualcosa allo schema
- Cambiata la locazione della mailing list di LDP a lists.debian.org da thepuffingroup.com.

v0.01, 27 agosto 1999

- Prima versione, creata la pagina web, scritto un semplice sommario.
- Qualcosa di quello che ho scritto è da prendere con le pinze. Delle cose devono essere verificate.

1.3 Copyright e marchi registrati (in inglese)

1.4 Riconoscimenti e ringraziamenti

Grazie a tutti quelli che hanno mandato commenti mentre stavo scrivendo questo documento. Questo include Deb Richardson, Daniel Barlow e altri membri della lista ldp-discuss.

Alcune sezioni sono state prese dall'HOWTO Index (disponibile in molte locazioni di LDP) e la documentazione degli sgmltools. Ci sono riferimenti agli sgmltools e a LDP altrove in questo documento.

2 Basi su LDP e SGML

2.1 LDP

Il Linux Documentation Project (LDP) fu creato per dare ai nuovi utenti un modo per avere informazioni velocemente su di un particolare argomento. Esso non solo contiene una serie di libri su amministrazione, reti e programmazione, ma anche un gran numero di lavori minori su argomenti individuali, scritti da chi vi ha lavorato. Per trovare qualcosa sulle stampanti, basta prendere il Printing HOWTO. Per trovare qualcosa sulle reti, basta prendere l'Ethernet HOWTO, e così via.

All'inizio, molti di questi lavori erano scritti in testo o HTML. Col passare del tempo, serviva un modo migliore per gestire questi documenti. Un modo che permettesse di leggerlo da una pagina web, da un file di testo su un CD-ROM o anche da un PDA. La risposta, appena esso uscì, fu l'SGML.

2.2 SGML

Lo Standard Generalized Markup Language (SGML) è un linguaggio basato sul testo marcato. In questo modo, è simile al Tex o al groff, o a HTML. La potenza di SGML è che, diversamente dal WYSIWYG (What You See Is What You Get, quello che vedi è quello che avrai), non vengono definite cose come colori, grandezza dei caratteri o un tipo di formattazione. Invece, vengono definiti degli elementi (paragrafi, sezioni, elenchi numerati) e si lascia al processore SGML alla fine di preoccuparsi di posizione, colori, caratteri e così via. L'HTML fa la stessa cosa, ed è attualmente un subset dell'SGML

L'SGML, in verità, è composto da due parti. La prima è la Struttura, chiamata comunemente DTD o Document Type Definition. Il DTD definisce le relazioni tra ognuno degli elementi. Il LinuxDoc DTD, usato per creare questo documento, ne è un esempio. Il DTD dà un aspetto comune ad ogni documento creato utilizzandolo. La seconda è il Contenuto, ovvero ciò che viene prodotto dal processore SGML e alla fine visto dall'utente. Questo paragrafo fa parte del Contenuto, ma ne farebbe parte anche un'immagine, una tabella, un elenco numerato e così via. Il Contenuto è circondato da tag per separare ogni differente elemento.

Col passare del tempo, il LinuxDoc DTD sta per essere sostituito dal DocBook DTD, usato da altri, dando così all'LDP un aspetto coerente col resto della documentazione SGML. Quando questo accadrà, sarete tenuti informati tramite questo HOWTO o sulle mailing list. La differenza più grande tra LinuxDoc e DocBook, è che DocBook assegna i tag a differenti tipi di contenuto (come comandi, nomi di file, directory e così via) mentre LinuxDoc assegna i tag basandosi sul modo in cui il testo deve apparire (si può enfatizzare il testo o farlo assomigliare ad una macchina da scrivere, per esempio).

2.2.1 Perché SGML invece di HTML o di altri formati?

L'SGML fornisce più della semplice formattazione. Si possono automaticamente costruire indici, sommari e collegamenti all'interno del documento o altrove. Il pacchetto sgmltools permette di esportare (lo chiamerò formattare da ora in poi) l'SGML in formato LaTeX, info, testo, HTML e RTF. Da questi formati di base, possono poi essere creati altri formati (DOC, PostScript e così via). L'SGML non soffre degli appesantimenti visti ultimamente nell'HTML. Non penso si vedrà molto presto un tag <blink> nell'SGML. Questo rende il codice non solo semplice da formattare, ma anche semplice da scrivere. Programmi come LyX (al momento il mio editor WYSIWYM scelto) permette di scrivere in formato TeX, esportarlo come SGML e poi formattare l'SGML ad un qualsiasi formato scelto.

Per finire, l'SGML si concentra più sul modo in cui gli elementi funzionano invece che sul modo in cui appaiono. Una grossa distinzione, che permette di scrivere più velocemente, in quanto non bisogna preoccuparsi della posizione dei paragrafi, grandezza dei caratteri, tipi di carattere e così via.

2.3 Gli strumenti

In questa sezione tratterò alcuni degli strumenti di cui avrete bisogno o che vorrete usare per creare la vostra documentazione LDP. Li descriverò qui e li definirò meglio più avanti, insieme alle istruzioni per l'installazione. Se utilizzate altri strumenti per aiutarvi nello scrivere materiale LDP, fatemelo sapere e aggiungerò una nota per esso.

2.3.1 sgmltools

Richiesti

Il pacchetto sgmltools contiene gli strumenti SGML necessari per formattare l'SGML in uno dei formati di file suddetti. Contiene anche il LinuxDoc DTD, necessario per creare la documentazione LDP. Per creare solo documentazione SGML, è tutto quello che serve. Se si vuole formattare in formati quali TeX, bisogna prendere almeno questo pacchetto. Il pacchetto sgmltools è disponibile nelle distribuzioni, oppure all'indirizzo *http://www.sgmltools.org/*

2.3.2 TeX

Opzionale

TeX (che fa rima con blech!) è il linguaggio di markup scelto da molti, incluse le persone del mondo matematico. Ancora ricordo molti esami di Calcolo scritti in TeX. È anche uno dei primi linguaggi di markup che sia ancora in giro (un altro è il formato *roff utilizzato nelle pagine man). Il TeX attualmente segue alcuni dei concetti dell'SGML. Comunque, esso compila i suoi file in DVI (DeVice Independent) che possono essere convertiti in un altro formato. Sfortunatamente, il DVI non può essere facilmente convertito in altro che alcuni linguaggi di stampanti (PostScript, PCL), rendendo difficile utilizzarlo per generare codice HTML. TeX è disponibile su praticamente tutte le distribuzioni come LaTeX o TeTeX. Dovrebbero andar bene entrambi.

2.3.3 LyX

Opzionale

Il programma LyX è un WYSIWYM (What You See Is What You Mean, quello che vedi è quello che intendi) grafico e fornisce un collegamento molto utile tra una applicazione grafica e un formattatore facile da utilizzare e le a volte complesse regole dell'SGML. LyX, effettivamente, serve per scrivere documenti TeX e molte delle regole del TeX si applicano in LyX. Per esempio, mentre le sezioni sono numerate automaticamente, non si possono inserire facilmente spazi bianchi (spazi e tabulazioni). È contro lo scopo per cui il TeX è stato creato. Nello stesso modo, l'SGML spesso ignora gli stessi spazi bianchi. Il programma LyX può leggere il LinuxDoc DTD e fornisce un documento modello per scrivere (o modificare) la propria documentazione LDP in modo familiare, senza dover utilizzare vi e ricordare quali sono i tag per fare un elenco puntato. LyX è disponibile all'indirizzo *http://www.lyx.org/*.

Per chi utilizza KDE, è disponibile un port di LyX che utilizza le librerie Qt. Ulteriori informazioni possono essere trovate all'indirizzo *http://www.devel.lyx.org/˜ettrich/klyx.html*. Se utilizzate KLyX per scrivere SGML, per favore mandate una e-mail al *mio indirizzo* per farmi partecipe delle vostre esperienze con esso.

3 Cominciare

Questa sezione mostra come venire coinvolto nello scrivere la propria documentazione LDP. Come ottenere e configurare gli strumenti, contattare LDP e distribuire le proprie conoscenze a tutti gli utenti Linux là fuori.

3.1 Per i nuovi autori

Se siete nuovi di LDP e volete prendere un HOWTO non mantenuto o scrivere un nuovo documento HOWTO o mini-HOWTO, contattate il coordinatore degli HOWTO all'indirizzo *linux-howto@metalab.unc.edu*. Questo per essere sicuri che il coordinatore degli HOWTO sappia chi sta lavorando su quale documento.

Notate anche che tutti gli HOWTO proposti devono essere in formato SGML (al momento utilizzando il LinuxDoc DTD). I mini-HOWTO proposti possono essere in formato SGML o HTML, ma solo quelli formattati in SGML potranno essere inclusi nelle versioni stampate degli HOWTO.

3.2 Le Mailing List

Ci sono delle mailing list alle quali iscriversi per sapere come funziona LDP. La prima è *ldp-discuss@lists.linuxdoc.org*, che è il maggiore gruppo di discussione di LDP. Per iscrivervi, mandate un messaggio con oggetto "subscribe" all'indirizzo *ldp-discuss-request@lists.linuxdoc.org*. Per deiscrivervi, mandate una e-mail con oggetto "unsubscribe" a *ldp-discuss-request@lists.linuxdoc.org*.

3.3 Scaricare ed installare gli strumenti

3.3.1 sgmltools

Scaricate il pacchetto sgmltools da *http://www.sgmltools.org/* o direttamente dalla vostra distribuzione. I file da sgmltools.org sono in codice sorgente, per cui dovete compilarli per la vostra macchina. Utilizzare un pacchetto precompilato per la propria distribuzione è più semplice, in quanto non bisogna compilarlo e potenzialmente ricevere errori di compilazione (è così, se non siete programmatori).

Con la RedHat, gli sgmltools sono inclusi nella distribuzione. Altrimenti, potete scaricarli da ftp.redhat.com o un altro dei suoi mirror come parte della distribuzione principale.

Se utilizzate la Debian, anche essa ha gli sgmltools nella distribuzione standard. Se non avete il pacchetto installato, potete utilizzare il comando apt-get per scaricare ed installare il pacchetto al posto vostro:

```
# apt-get install sgml-tools
```

Per maggiori informazioni sui pacchetti Debian, potete andare all'indirizzo *http://www.debian.org/Packages/stable/text/sgml-tools.html*.

Se state compilando il codice sorgente, tutto quello che dovete fare è:

```
# tar -zxvf sgmltools-x.x.x.tar.gz
# cd sgmltools-x.x.x
# ./configure
# make
# make install
```

Sostituite `sgmltools-x.x.x` con la versione attuale del pacchetto sgmltools che state utilizzando. Mentre sto scrivendo, la versione che supporta LinuxDoc è la 1.0.9. Quella che supporta il DocBook è la 2.0.2. Entrambe sono disponibili al sito web riportato sopra.

Una volta che gli strumenti sono installati, sono disponibili una serie di comandi.

`sgmlcheck file.sgml` - Controlla la sintassi di un dato documento.

`sgml2html file.sgml` - Converte un file SGML in HTML. Crea un file `file.html` che contiene l'Indice, poi crea i file `file-x.html`, dove `x` è il numero della sezione.

`sgml2rtf file.sgml` - Converte un file SGML in formato Rich Text Format (RTF). Crea due file, il primo chiamato file.rtf che contiene la TOC (Indice), ed uno chiamato `file-0.rtf`, che contiene tutte le sezioni.

`sgml2txt file.sgml` - Converte un file SGML in testo ASCII. La TOC e tutte le sezioni sono messe tutte nel file `file.txt`.

`sgml2info file.sgml` - blabla SGML blabla INFO, utilizzato dal comando info. Tutto l'output viene messo nel file `file.info`.

`sgml2latex file.sgml` - blabla SGML blabla TeX.

`sgml2lyx file.sgml` - SGML yadda LyX graphical editor. Ottimo se si hanno SGML pre-generati e si vogliono convertire per l'uso con LyX.

3.4 Scrivere l'SGML a mano

Come con HTML, è possibile scrivere SGML a mano, quando si conoscono i codici di marcatura (tag) che si vogliono utilizzare. Questa sezione illustrerà più tag possibile, con esempi pratici di ognuno. Un buon posto da dove iniziare può essere il sorgente SGML di questo documento, che è disponibile all'indirizzo riportato nell'1 (Introduzione). Anche se SGML può essere processato in modi diversi a seconda del formato del file in cui viene convertito, proverò ad elencare delle cose da sapere durante la scrittura.

3.4.1 Iniziare

Per iniziare un nuovo documento, create un nuovo file nel vostro editor ASCII preferito e iniziate con:

```
<!doctype linuxdoc system>
```

Questo definisce il tipo di documento (LinuxDoc nel caso nostro) che il processore SGML utilizzerà in fase di formattazione del file nel formato di uscita. Questo tag non produce nulla in uscita.

Poi bisogna chiudere il resto del lavoro nei tag `<article>` e `</article>`. Questi indicano l'inizio del contenuto (o articolo, eh?). Se si è familiari con l'HTML, è simile all'includere tutto il contenuto nei tag `<html>` e `</html>`.

3.4.2 Informazioni dell'intestazione

La prima parte del contenuto dovrebbe contenere informazioni generali sul resto del contenuto. Vorrebbe essere simile alle prime pagine di un libro, dove troviamo titolo, autore, data di pubblicazione, indice, eccetera.

Il titolo del contenuto è chiuso nei tag `<title>` e `</title>`. L'autore è specificato nei tag `<author>` e `</author>`. La data usa i tag `<date>` e `</date>`.

Le due sezioni rimanenti sono i tag `<abstract>` e `</abstract>`, che forniscono un sommario sul contenuto e il tag `<toc>`, che specifica la posizione della TOC. La TOC viene generata automaticamente dal processore SGML. Vedremo le sezioni più tardi.

Ora, come appare tutto insieme? Preso questo bel pezzo di codice SGML (utilizzato per creare questo documento), si può vedere:

```
<!doctype linuxdoc system>
<!-- LinuxDoc file was created by LyX 1.0 (C) 1995-1999 by <markk>
 Fri Aug 27 09:42:28 1999 -->
<article>
<title>HOWTO HOWTO</title>
<author>Mark F. Komarinski</author>
```

```
<date>27 Agosto 1999 </date>
<abstract>Aiutare un nuovo autore LDP a cominciare con strumenti, idee, e
 convenzioni usate dall'LDP</abstract>
<toc>
```

Questo pezzo del contenuto ha creato la pagina principale che è possibile vedere guardando questo documento in formato RTF o HTML, mettendo tutte le informazioni in una pagina.

3.4.3 Sezioni

Per costruire l'Indice, bisogna avere qualcosa con cui costruirlo. Le sezioni nel caso dell'SGML sono l'equivalente dei capitoli nelle pubblicazioni tradizionali. Sono disponibili sezioni multiple e ogni sezione può avere sottosezioni e ognuna di esse può avere sottosezioni eccetera.

Cominciare i documenti con le sezioni è comodo in quanto permette di creare una bozza degli argomenti principali da coprire. Poi si possono spezzare queste sezioni principali in sezioni gradatamente più piccole, fino ad ottenere delle piccole parti di cui si può scrivere in pochi piccoli paragrafi. Ho cominciato così a scrivere questo documento.

Le sezioni sono uno dei pochi insiemi di tag SGML che non richiedono di essere chiusi. Non ci sono tag `</sect>`. E non c'è bisogno di preoccuparsi della numerazione. Ci penserà il processore a gestirla quando l'SGML verrà formattato in un altro formato.

Le sezioni cominciano con il tag `<sect>`. Per ogni tag `<sect>` viene cominciata una nuova sezione. La prima sezione viene numerata con il numero 1.

Le sottosezioni (come 1.1) si creano con il tag `<sect1>`. Anche esse cominciano con 1.

Le sotto-sottosezioni (1.1.1) si creano con il tag `<sect2>`, ed anche esse cominciano con 1.

Quando il processore SGML trova il tag `<toc>`, passa attraverso il resto del documento e costruisce l'Indice basato sul numero dei tag di sezione ivi contenuti. Le sezioni vengono numerate e elencate nella TOC e poi utilizzate nel resto del documento. Le sotto-sottosezioni (1.1.1) non vengono mostrate nella TOC, ma sono mostrate in testo enfatizzato se possibile.

3.4.4 Paragrafi normali

Scrivere paragrafi del contenuto è proprio come l'HTML. Utilizzate un tag `<p>` per specificare una nuova riga e cominciate a scrivere. L'SGML ignora gli spazi bianchi come tabulazioni, spazi multipli e ritorni a capo. Quando l'SGML incontra un tag `<p>` inizia un nuovo paragrafo. Il corretto SGML prevede un tag `</p>` per chiudere il paragrafo.

3.4.5 Testo modificato

Ogni tanto potrebbe servire un poco di testo diverso dal resto, per evidenziare del codice o per un `comando`. Il testo enfatizzato va racchiuso con i tag `<em>` e `</em>`. Il testo dattiloscritto va racchiuso con i tag `<tt>` e `</tt>`.

3.4.6 Liste

Ci sono due modi per fare liste in SGML. Il primo è la lista numerata, dove ogni elemento della lista viene numerato (come le sezioni) cominciando con 1.

1. Questo è il primo elemento della lista numerata.
2. Questo è il secondo.
3. Terzo.

Il codice per la lista sopra appare così;

```
<enum>
<item>Questo è il primo elemento della lista numerata.
<item>Questo è il secondo.
<item>Terzo.
</enum>
```

Il tag <enum> specifica che gli elementi seguenti saranno numerati.

L'altro metodo di scrivere liste è la lista puntata, dove ogni elemento ha una stella, un cerchio, un punto o un altro metodo per puntare gli elementi.

- Questo è il primo elemento della lista puntata.
- Questo è il secondo.
- Terzo.

Il codice sopra appare così in SGML puro:

```
<itemize>
<item>Questo è il primo elemento della lista puntata.
<item>Questo è il secondo.
<item>Terzo.
</itemize>
```

Come potete vedere, il tag <item> è lo stesso per la lista numerata e quella puntata.

Una terza forma di lista è la lista descrittiva. Essa ha un termine descritto e una frase che lo descrive.

LDP

The Linux Documentation Project

SGML

Standard Generalized Markup Language

Il codice per creare le descrizioni sopra è:

```
<descrip>
 <tag>LDP</tag>The Linux Documentation Project
 <tag>SGML</tag>Standard Generalized Markup Language
 </descrip>
```

Non è proprio uguale alle liste puntate o numerate, ma la lista è interamente racchiusa dai tag (<descrip> e </descrip>) ed ogni elemento della linea che è una parola definita va racchiusa nei tag <tag> e <tag>. Il resto della linea viene preso come la definizione della parola.

3.4.7 Testo riportato

Delle volte serve stampare il testo nel modo in cui è scritto. Per far questo, vanno usati i tag `<verb>` e `</verb>` per racchiudere il paragrafo di testo da riportare. Spazi, ritorni carrello, ed altro testo (inclusi i caratteri speciali) sono riportati letteralmente fino al tag `</verb>`.

```
Il testo seguente viene riportato letteralmente       .
```

3.4.8 URL

Anche nell'SGML è possibile gestire Universal Resource Locator (URL) di ogni tipo. Notate che essi funzionano solamente quando esportati in HTML, ma ci possono essere degli usi per questo tag in altri formati (lo usa anche l'RTF?).

Una URL non ha un tag di chiusura, ma mette queste informazioni tra il tag `<url>` stesso. Ecco una URL che punta alla pagina principale dell' LDP: *http://www.linuxdoc.org/* `<http://www.linuxdoc.org/>`. Ed ecco il codice per crearla:

```
<url url="http://www.linuxdoc.org/" name="http://www.linuxdoc.org/">
```

`url="http://www.linuxdoc.org/"` dice al browser dove andare, mentre i contenuti di `name="http://www.linuxdoc.org/"` dicono al browser cosa scrivere sullo schermo. In questo caso, i due sono simili, ma è possibile creare un tag URL che appare così:

```
<url url="http://www.linuxdoc.org/" name="LDP">
```

E sulla pagina appare come: *LDP* `<http://www.linuxdoc.org/>`.

3.4.9 Riferimenti

Mentre le URL sono ottime per collegarsi con i contenuti esterni al documento a cui si sta lavorando, non sono molto comode per collegarsi con il contenuto stesso. Per questo, si usano i tag `<label>` e `<ref>`. Il tag `<label>` crea un punto nel documento a cui riferirsi poi, come un segnalibro. Creare il tag `<label>` è semplice. Posizionatevi nel punto al quale volete riferirvi poi, e inserite il seguente:

```
<label id="Introduction">
```

Avete ora creato un punto nel documento al quale potete riferirvi dopo come "Introduction". Questa etichetta attualmente appare in questo SGML all'inizio del documento. Quando volete riferirvi a quel punto più tardi (ad esempio 1 (qui)), inserite il seguente tag:

```
<ref id="Introduction" name="qui">
```

e l'SGML inserirà un collegamento chiamato "qui" (vedi sopra) che rimanda alla sezione "Introduzione".

L'altro scopo dei riferimenti è l'indicizzazione. Siccome la documentazione LDP viene solitamente pubblicata su carta come una grande raccolta di documenti, serve un metodo per costruire l'indice in fondo al libro, basato su parole e oggetti.

3.4.10 Caratteri speciali

Come in HTML, c'è bisogno di fare l'escape di molti caratteri non-alfanumerici per impedire al processore SGML di interpretarli come codice SGML. Ecco una lista di codici SGML utilizzati. Altri sono nella Guida Utente sgmltools *http://www.sgmltools.org/guide/guide.html* `<http://www.sgmltools.org/guide/guide.html>`.

- Utilizzare & per una e commerciale (&)
- Utilizzare < per un segno di minore (<)
- Utilizzare > per un segno di maggiore (>)
- Utilizzare &etago; per un segno di minore con uno slash (</)
- Utilizzare $ per un segno di dollaro ($)
- Utilizzare # per un cancelletto (#)
- Utilizzare % per un segno di percentuale (%)
- Utilizzare ˜ per una tilde (˜)
- Utilizzare " e " per le citazioni, oppure &dquot; per `"`
- Utilizzare per un trattino di unione (ovvero, una indicazione che quello è un buon posto per spezzare una parola per il ritorno a capo).

3.5 Scrivere l'SGML utilizzando altri strumenti

3.5.1 LyX

Mi sto ancora entusiasmando troppo per LyX. Okay, sono un poco obbiettivo verso questa applicazione perché mi piace veramente. Fornisce la potenza di scrivere in SGML con la facilità di un normale word processor. Non è un programma WYSIWYG, ma un'applicazione più WYGIWYM (What You Get Is What You Mean, quello che ottieni è quello che intendi), in quanto quello che si vede sullo schermo non è necessariamente quello che si ottiene dopo che il processore SGML l'ha elaborato.

Per creare un documento LinuxDoc con LyX, scaricate ed installate l'applicazione. Assicuratevi di avere TeX e gli sgmltools installati (vedere 3.3 (Scaricare ed installare gli strumenti) per maggiori informazioni). Una volta finito, avviate LyX e selezionate "file->new from template..." Selezionate "Templates" poi fate click su `linuxdoctemplate.lyx` e avrete pronto un modello di documento, con la maggior parte delle informazioni di intestazione che dovrebbe avere un documento LDP. Cambiate i dati per soddisfare i vostri bisogni (ovvero, inserite il Titolo, Autore, Data, Sommario, eccetera) e cominciate a scrivere. Il menu a comparsa nell'angolo in alto a destra può essere utilizzato per selezionare tipi di contenuto (standard, liste puntate e numerate, sezioni). Il punto esclamativo viene utilizzato per enfatizzare il testo e potete o farci click sopra e cominciare a scrivere in modo enfatizzato oppure selezionare il testo con il mouse e farci click per enfatizzare il testo selezionato. Molte altre caratteristiche possono essere trovate sotto il menu Insert. Potete inserire url, riferimenti incrociati, elementi dell'indice e altri tipi di dato. Quando la documentazione è completata, potete salvarla in formato LyX, poi esportarla in LinuxDoc ed avere il file salvato con un'estensione .sgml. Questo file è poi pronto per essere verificato con sgmlcheck e convertito nei formati voluti.

3.5.2 Emacs

Ho questa cosa su Emacs. Non lo uso e la cosa non mi irrita. Chiunque con più esperienza di Emacs sarebbe molto utile.

3.5.3 Altri strumenti SGML

Se ci sono altri strumenti SGML, anche commerciali, con le quali possa essere utilizzato il LinuxDoc DTD per creare la documentazione LDP, fatemelo sapere.

3.6 Le basi del CVS

Al momento, l'LDP non ha un deposito condiviso per archiviare i contenuti online. Speriamo che la cosa cambi. Ci sono alcune buoni ragioni per utilizzare CVS:

1. Il CVS mantiene un backup off-site dei documenti. Nel caso cediate un documento ad un altro autore, lui può recuperare il documento dal CVS e continuare. Nel caso ci fosse bisogno di tornare ad una versione precedente di un documento, anche esso può essere recuperato.
2. È comodo se ci sono molte persone che lavorano sullo stesso documento. È possibile farsi dire dal CVS quali cambiamenti sono stati fatti da un altro autore mentre stavate modificando la vostra copia e integrare quei cambiamenti.
3. Tiene traccia dei cambiamenti fatti. Questi log (ed una data) possono essere inseriti automaticamente nel documento utilizzando dei tag speciali che vengono processati prima che il documento venga processato dal processore SGML.
4. Fornisce un metodo per un programma di aggiornare automaticamente il sito web LDP con nuova documentazione appena viene scritta e proposta.

3.7 Distribuire la documentazione

3.7.1 Prima della distribuzione

Prima di distribuire il codice a milioni di potenziali lettori ci sono dei passi da seguire.

Primo, assicuratevi di controllare l'ortografia del documento. Nulla dice "Sì, sono stupido!" più velocemente degli errori di scrittura nella terra di Internet. La maggior parte delle utilità che utilizzate per scrivere in SGML (emacs, LyX, altri editor di testi) hanno dei plug-in per fare un controllo ortografico. Altrimenti, c'è sempre il programma ispell, installato in praticamente ogni distribuzione. Utilizzate anche il comando sgmlcheck degli sgmltools per verificare i tag SGML.

Secondo, fate leggere a qualcuno la vostra documentazione per commenti e correttezza formale. La documentazione pubblicata da LDP deve essere il più possibile corretta, in quanto ci sono milioni di utenti Linux che potrebbero leggerla. Se fate parte di una grossa mailing list che parla dello stesso soggetto, chiedete agli altri della lista di aiutarvi.

Terzo, create un sito web dove poter distribuire la documentazione. Non è obbligatorio, ma è utile per le persone per trovare la locazione originale del vostro documento.

3.7.2 Copyright e Licenze

Perché un documento LDP sia accettato dall'LDP, deve avere una licenza che permetta la libera distribuzione (come la birra) e pubblicazione. Come autore, potete detenere il copyright ed aggiungere altre restrizioni (per esempio, il dover approvare ogni traduzione o lavori derivati). Una licenza d'esempio è disponibile all'indirizzo *http://www.linuxdoc.org/COPYRIGHT.html*. Se scegliete di utilizzare il copyright di quel documento, copiatelo semplicemente nel vostro codice sorgente in una sezione chiamata "Copyright e Licenze" o simile. Includete anche una dichiarazione di copyright propria (in quanto ancora lo possedete). Se siete un nuovo mantenitore per un HOWTO già esistente, dovete includere la dichiarazione di copyright dell'autore/i precedente/i e le date in cui loro mantenevano il documento.

3.7.3 Proposta a LDP

Quando il documento LDP è stato riletto da alcune persone e avete preso in considerazione i loro commenti, potete rilasciare il vostro documento a LDP in generale. Mandate una email a *ldp-submit@lists.linuxdoc.org* con il vostro sorgente. Entro 24 ore dovreste sapere se è stato accettato e inserito nel sito principale di LDP.

4 Suggerimenti di stile

Questa non è (ancora) una veloce guida su come scrivere una buona documentazione, ma consideratela una manciata di consigli per aiutarvi nella scrittura.

- Siate chiari. Ognuno deve sapere di cosa state parlando.
- Usate degli esempi dove possibile. Fa vedere a tutti di cosa state parlando.
- Organizzate. Non saltate tra argomenti slegati tra loro nella stessa sezione.

Potete trovare molti altri consigli dalla guida allo stile LDP situata all'indirizzo *http://www.linuxdoc.org/HOWTO/LDP-Style-Guide.html*.

5 FAQ su LDP

5.1 Voglio aiutare LDP. Come posso fare?

Il modo più facile è trovare qualcosa e documentarlo. Controllate anche gli HOWTO non mantenuti e guardate se c'è un soggetto di cui sapete qualcosa e potete continuare a documentare.

5.2 Voglio pubblicare una raccolta di documenti LDP in un libro. Come è la licenza del contenuto LDP?

Guardate l'indirizzo *http://www.linuxdoc.org/COPYRIGHT.html*.

5.3 Ho trovato un errore in un documento LDP. Posso correggerlo?

Contattate l'autore del documento o il coordinatore di LDP (e-mail?) e menzionate il problema e come pensate possa essere corretto.