HOWTO HOWTO

Mark F. Komarinski (*markk@cgipc.com*)
Adaptation française par Nicolas Duboc (*duboc@essi.fr*) $Id: HOWTO-HOWTO.sgml, v1.4, 17 Mars 2000 markk Exp $

Explique les outils, les procédures, et donne des conseils aux auteurs de HOWTO.

Table des matières

1 Introduction

1.1 Historique

Ce document a été commencé le 26 août 1999 par Mark F. Komarinski (*markk@cgipc.com*) après deux jours de frustration pour faire marcher les outils. Si ne serait ce qu'un auteur du LDP (NdT: "Linux Documentation Project", Projet de Documentation Linux) trouve de l'aide dans ce document, j'aurai atteint mon objectif.

1.2 Nouvelles versions

La version la plus récente de ce document peut être trouvée sur ma page personnelle *http://www.cgipc.com/~markk/* au format SGML. D'autres versions peuvent être trouvées dans différents formats sur le site du LDP *http://www.linuxdoc.org/*.

1.3 Commentaires

Les commentaires à propos de ce HOWTO peuvent être envoyés à l'auteur (*markk@cgipc.com*).

1.3.1 Historique des versions

v1.4 (17 Mars 2000)

- Ajout des commentaires de Gregory Leblanc (*GLeblanc@cu-portland.edu*) à propos des soumissions au LDP et sur psgml sous Emacs. Merci !

1.4 Copyrights

1.5 Contributions et remerciements

Merci à tous ceux qui ont donné leurs commentaires lorsque j'écrivais ceci. Cela inclut Deb Richardson et Daniel Barlow et les autres membres de la liste de diffusion ldp-discuss.

J'ai tiré quelques sections du Howto Index (disponible sur tous les sites du LDP) et de la documentation de sgmltools. Vous trouverez les références à sgmltools et le LDP ailleurs dans ce document.

Les sections sur l'accès réseau à CVS ont été partiellement écrites par Serek (*ser@serek.arch.pwr.wroc.pl*). Les sections sur DocBook ont été en partie écrites par Jorge Godoy (*godoy@conectiva.com.br* `<mailto:godoy@conectiva.com.br>`). Un grand merci à tous les deux pour leur aide.

2 Informations générales sur le LDP et SGML

2.1 Le LDP

Le Projet de Documentation de Linux (LDP) a été commencé pour fournir aux nouveaux utilisateurs un moyen d'obtenir rapidement des informations sur un sujet particulier. Il ne contient pas seulement une série de livres sur l'administration système, le réseau ou la programmation, mais également un grand nombre de petits travaux sur des sujets plus individuels, écrits par ceux qui les ont utilisés. Si vous voulez savoir comment imprimer, vous trouverez le Printing HOWTO. Si vous voulez mettre en place un réseau, récupérez le Ethernet HOWTO; etc.

Au départ, la plupart de ces documents étaient au format texte ou HTML. Avec le temps, il fallait trouver un meilleur moyen de gérer ces travaux. Celui-ci permettrai de les lire depuis une page web, depuis un fichier texte sur un CD-ROM, ou encore depuis votre organiseur de poche. La solution s'est avérer être SGML.

2.2 SGML

Le "Standard Generalized Markup Language" (SGML) est un langage qui est basé sur le balisage du texte. De ce point de vue, il est similaire à HTML, mais toutes les similitudes s'arrètent là. La force de SGML est que contrairement à la philosophie WYSIWYG (What You See Is What You Get, NdT: Ce que vous voyez est ce que vous obtenez), vous ne définissez pas la couleur, la taille de la police, ou d'autres paramètres de formatage. A la place, vous définissez des éléments (paragraphes, sections, listes numérotées) et laissez l'interprète SGML et le programme final s'occuper du placement, des couleurs, des polices, et tout le reste. HTML fait la même chose, puisqu'il est en fait une subdivision de SGML.

Il y a en fait deux parties qui constituent SGML. La première est la Structure, qui est appelée la DTD, ou encore Définition du Type du Document. La DTD définie les relations entre les différents éléments d'un texte. La DTD LinuxDoc, utilisée pour créer ce document, en est un exemple. La DTD fournit un style et une apparence à tous les documents créés qui l'utilisent. La seconde est le Contenu, qui est traité par l'interpréteur SGML et qui est éventuellement vu par l'utilisateur. Ce paragraphe est un contenu, comme pourrait l'être une image, une table, une liste numérotée ou autre chose. Le contenu est entouré de balises pour séparer les différents éléments.

Dans quelques temps, le Projet de Documentation Linux passera de LinuxDoc à DocBook, déjà utilisée par d'autres auteurs, donnant ainsi au LDP un style plus consistent de documents SGML. Quand cela arrivera, nous vous tiendront informés par l'intermédiaire de ce HOWTO ou des listes de diffusion. La différence la plus importante réside dans le fait que DocBook utilise des balises pour les différents types de contenus (comme les commandes, les noms de fichiers, de répertoires, etc.) alors que LinuxDoc utilise des balises basées sur l'apparence que devrait prendre le texte (vous pouvez utiliser un type "insistant" ou machine à écrire).

2.2.1 Pourquoi SGML au lieu de HTML ou d'autres formats ?

SGML fournit d'autres outils que ceux de mise en forme. Vous pouvez créer des index, des tables des matières ou des liens internes ou externes automatiquement. Le paquetage sgmltools vous permet aussi d'exporter (j'appèlerai ça générer à partir de maintenant) vos documents SGML vers LaTeX, info, du texte, HTML ou RTF. Vous pouvez alors créer d'autres types de fichiers (DOC, PostScript, etc.) à partir de ces formats de base. Des programmes comme LyX (mon éditeur WYSIWYM préféré) vous permettent d'écrire au format TeX, d'exporter au format SGML et de générer ce que vous voulez depuis SGML.

En fin de compte, SGML s'intéresse plus au fonctionnement des éléments plutôt qu'à leur apparence. Une différence de taille, en tout cas une qui vous permettra d'écrire plus vite, puisque vous n'avez plus à vous souciez de la justification des paragraphes, des types et tailles des polices, etc.

2.3 Les outils

Dans cette section, je vais survoler les outils dont vous aurez besoin ou voudrez utiliser pour créer votre propre documentation LDP. Je vais les décrire ici, et mieux les explorer plus tard, en même temps que leur procédure d'installation. Si vous utilisez d'autres outils pour écrire des manuels du LDP, faites le moi savoir, j'ajouterai un descriptif ici.

2.3.1 sgmltools

Requis

Le paquetage sgmltools contient les outils SGML nécessaires pour générer les formats dont on a déjà parlé. Il inclut également la DTD LinuxDoc, indispensable aux documents LDP. Pour créer des documents SGML, c'est tout ce dont vous avez besoin. Si vous voulez convertir au format TeX, vous aurez aussi besoin de ce dernier. Le paquetage sgmltools est disponible soit dans votre distribution, soit via *http://www.sgmltools.org/*. Notez que vous aurez besoin de la version 1.0.9 pour utiliser LinuxDoc. Toute autre version utilise DocBook.

2.3.2 TeX

Optionnel

TeX est le langage de balisage de texte favori pour beaucoup, dont les personnes du monde mathématique. Je me rappelle d'examens de maths qui étaient en fait écris avec TeX. C'est aussi un des premiers langages à balises qui est encore utilisé (les autres étant les formats *roff utilisés pour les pages de manuels). En fait, TeX partage une grande partie de ses concepts avec SGML. Néanmoins, TeX génère ses fichiers au format DVI (indépendant du périphérique de sortie) qui peut alors être converti en un autre format. Malheureusement, le format DVI ne peut pas être facilement converti en autre chose que des langages d'impression (PostScript, PCL), le rendant difficile à traduire en HTML. TeX est disponible dans toutes les distributions Linux en tant que LaTeX ou TeTex. Vous pourrez utiliser l'un des deux.

2.3.3 LyX

Optionnel

Le programme LyX est un éditeur de texte graphique WYSIWYM (What You See Is What You Mean – NdT : Ce que vous voyez est ce que vous pensez) et constitue un lien entre les applications graphiques et les règles de SGML quelque fois compliquées. En fait, LyX est utilisé pour écrire des documents TeX, et la plupart dans règles TeX sont présentes dans LyX. Par exemple, puisque les sections sont numérotées automatiquement, vous ne pouvez pas insérer d'espace facilement. Ce serait contraire à la philosophie de

TeX. De même, SGML ignore souvent ces espaces. Le programme LyX supporte la DTD LinuxDoc et fournit un document de base pour vous permettre d'écrire (ou d'éditer) votre documentation LDP comme vous en avez l'habitude, sans devoir utiliser vi et vous souvenir quelles sont les balises pour créer une liste à items. Vous trouverez LyX à l'adresse *http://www.lyx.org/*.

Pour ceux qui utilisent KDE, il existe un portage de LyX basé sur les librairies Qt. Vous trouverez plus d'informations à l'adresse *http://www.devel.lyx.org/~ettrich/klyx.html* `<http://www.devel.lyx.org/~ettrich/klyx.html>`.

2.3.4 Emacs (PSGML)

Optionnel

Il existe un mode Emacs pour écrire des documents SGML ou XML. Vous trouverez plus d'informations à *http://www.lysator.liu.se/projects/about_psgml.html*

2.3.5 WordPerfect 2000

Optionnel

La dernière version de WordPerfect 2000 supportera les modes SGML. Je ne sais pas encore si celà concerne aussi LinuxDoc, ou juste DocBook. Si vous êtes un beta-testeur de WP2000 et si vous pouvez me dire comme il marche, je serais heureux d'inclure vos notes.

2.3.6 DocBook : L'ultime guide

Optionnel (mais recommandé)

Ce livre est édité par O'Reilly depuis Octobre 1999 et c'est une bonne référence pour DocBook. Je n'ai pas trouvé qu'il soit très pratique, et la plupart des "gloires"est pour XML, mais les balises DocBook 3.1 sont toutes listées dans un format concit. Vous pourrez le trouver chez votre vendeurs favori, ou le retrouver en intégralité sur le site *http://www.docbook.org* `<http://www.docbook.org/>`.

3 Démarrer avec DocBook

Cette section couvre les nouvelles méthodes pour écrire de la documentation pour le LDP, en utilisant la DTD DocBook 3.1. Nous verrons comment récupérer, installer et utiliser les outils, ainsi qu'une introduction aux balises DocBook. Etant donné qu'il y a près de 300 balises DocBook, nous ne les verront pas toutes ici. Les lecteurs vraiment intéressés pourront se tourner vers *http//www.docbook.org* `<http://www.docbook.org/>` pour plus d'informations.

3.1 Pour les nouveaux auteurs

Si vous êtes un nouvel auteur au sein du LDP, et que vous voulez prendre en main un HOWTO (ou un Mini-HOWTO) non maintenu ou en écrire un vous même, contactez le coordinateur du LDP à l'adresse *linux-howto@metalab.unc.edu*. Cela lui permet de savoir qui travaille sur quel document. Tous les HOWTO envoyés devront être au format SGML (basés sur la DTD LinuxDoc ou DocBook). Les mini-HOWTO pourront être aux formats SGML ou HTML, mais seuls les documents SGML seront inclus dans les versions imprimées des HOWTO. Si vous écrivez un Mini-HOWTO, en SGML, il sera plus facile de le convertir dans les autres formats, permettant à plus de monde de profiter de votre travail.

3.2 Les listes de discussion

Il y a quelques listes de discussion auxquelles vous pouvez vous abonner pour prendre part au fonctionnement du LDP. La première est *ldp-discuss@lists.linuxdoc.org*, qui est le principal lieu de discussion du LDP. Pour s'abonner, il suffit d'envoyer un message avec "subscribe" dans le champ du sujet à l'adresse *mailto:ldp-discuss-request@lists.linuxdoc.org* `<mailto:ldp-discuss-request@lists.linuxdoc.org>`. Pour se désabonner, même adresse avec "unsubscribe" dans le champ sujet du message.

3.3 Télécharger et installer les outils

3.3.1 sgmltools

Contrairement aux versions précédentes, vous aurez besoin de la version 2.x de sgmltools pour utiliser DocBook. Puisque tous les programmes de base ont changé, vous devrez aussi oublier les programmes du type sgml2xxx (désolé). Puisque toutes les principales distributions arrivent avec sgmltools 1.x, vous devrez supprimer le paquetage sgmltools 1.x et installer une version 2.0 ou une version CVS.

Pour obtenir le source de la dernière version CVS, vous pouvez utiliser les commandes suivantes :

```
CVSROOT=:pserver:cvs@cvs.sgmltools.org:/home/cvs
export CVSROOT
cvs login
cvs -z6 get sgmltools
```

Le mot de passe CVS est '`cvs`'.

Une fois téléchargé, vous avez juste à faire `./compile`, `make`, et `make install` pour installer sgmltools.

Pour les systèmes RedHat (utilisant RPM) vous pouvez utiliser la commande rpmfind pour obtenir le dernier paquetage sgmltools. Le programme rpmfind est disponible sur *http://www.rpmfind.net/* `<http://www.rpmfind.net/>`. Assurez-vous que vous avez recupéré sgmltools et non sgml-tools, car ce dernier est sgml-tools 1.0.9.

Pour les systèmes Debian, apt-get récupèrera le bon paquetage pour vous :

```
# apt-get install sgmltools
```

Comme pour les systèmes RedHat, il faut récupérer sgmltools et non sgml-tools.

3.4 Ecrire du SGML à la main

Puisque ceci est largement traité dans d'autres ouvrages, les personnes intéressées pourront se réferrer à *http://metalab.unc.edu/godoy/using-docbook/using-docbook.html* `<http://metalab.unc.edu/godoy/using-docbook/using-docbook.html>` pour plus d'informations sur savoir comment produire du SGML à la main avec votre éditeur de texte favori.

4 Démarrer avec LinuxDoc

Cette section décrit comme être au point pour écrire vos propres documents LDP. Récupérer et configurer les outils, prendre contact avec le LDP en général, et partager vos connaissances avec tous les utilisateurs de Linux.

4.1 Pour les nouveaux auteurs

Si vous êtes un nouvel auteur au sein du LDP, et que vous voulez prendre en main un HOWTO (ou un Mini-HOWTO) non maintenu ou en écrire un vous même, contactez le coordinateur du LDP à l'adresse *ldp-discuss@lists.linuxdoc.org*. Cela lui permet de savoir qui travaille sur quel document. Tous les HOWTO envoyés devront être au format SGML (basés sur la DTD LinuxDoc ou DocBook). Les mini-HOWTO pourront être aux formats SGML ou HTML, mais seuls les documents SGML seront inclus dans les versions imprimées des HOWTO.

4.2 Les listes de discussion

Il y a quelques listes de discussion auxquelles vous pouvez vous abonner pour prendre part au fonctionnement du LDP. La première est *ldp-discuss@lists.linuxdoc.org*, qui est le principal lieu de discussion du LDP. Pour s'abonner, il suffit d'envoyer un message avec "subscribe" dans le champ du sujet à l'adresse *mailto:ldp-discuss-request@lists.linuxdoc.org* `<mailto:ldp-discuss-request@lists.linuxdoc.org>`. Pour se désabonner, même adresse avec "unsubscribe" dans le champ sujet du message.

4.3 Télécharger et installer les outils

4.3.1 sgmltools

Récupérez le paquetage sgmltools depuis *http://www.sgmltools.org/*, ou directement depuis votre distribution. Les fichiers de sgmltools.org sont le code source de l'application, vous devrez donc les compiler pour votre machine. Utiliser un paquetage pré-compilé pour votre distribution est plus facile, puisque vous n'aurez pas à le compiler et éviterez les problèmes de compilation (sauf si vous êtes un programmeur).

Les outils sgmltools sont inclus dans la distribution RedHat. Si ce n'est pas le cas, vous pouvez le télécharger depuis *le site ftp de RedHat* ou un des sites miroirs.

Si vous utilisez une Debian, sgmltools est également inclus en standard. Dans le cas contraire, vous pourrez utiliser apt-get pour télécharger et installer le paquetage à votre place :

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

Pour plus d'informations sur le paquetage Debian, regarder à l'adresse *http://www.debian.org/Packages/stable/text/sgml-tools.html*

Pour la compilation des sources, la marche à suivre est :

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

Remplacez `sgmltools-x.x.x` par la version du paquetage que vous utilisez. A la date où j'écris ces lignes, la version qui supporte LinuxDoc est la 1.0.9, et celle qui supporte DocBook est 2.0.2. Ces deux versions sont disponibles sur le site web déjà vu ci-dessus.

Une fois les outils installés, vous avez plusieurs commandes à disposition.

`sgmlcheck file.sgml` - Vérifie la syntaxe du document `sgml2html file.sgml` - Convertit un fichier SGML en HTML. Le fichier `file.html` contiendra la table des matières, les fichiers `file-x.html` contiendront les sections numérotées x.

`sgml2rtf file.sgml` - Convertit un fichier SGML en deux fichiers Rich Text Format (RTF). Le fichier `file.rtf` contiendra la table des matières et `file-0.rtf` contiendra toutes les sections.

`sgml2txt file.sgml` - Convertit un fichier SGML en texte ASCII. La table des matières et les sections sont toutes dans le fichier `file.txt`.

`sgml2info file.sgml` - Blah SGML blah INFO, utilisé par la commande info. Tout est contenu dans `file.info`.

`sgml2latex file.sgml` - Blah SGML blah TeX.

`sgml2lyx file.sgml` - SGML converti pour l'éditeur graphique LyX. Intéressant si vous avez généré un fichier SGML et voulez le convertir pour l'utiliser avec LyX.

4.4 Ecrire du SGML à la main

Tout comme le HTML, vous pourrez écrire du SGML à la main, une fois que vous connaîtrez toutes les balises dont vous aurez besoin. Vous trouverez ici une description pour la plupart des balises, avec des exemples d'utilisations. Le code source SGML de ce document, disponible sur le site web vu dans la section 1 (Introduction), est un bon exemple pour démarrer l'apprentissage. J'essairai de donner les indications sur les interprétations des balises dans les différents formats de conversion.

4.4.1 Début du document

Pour commencer un nouveau document, créez un nouveau fichier avec votre éditeur ASCII favori et commencez comme ça :

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

Cela précise le type de document (LinuxDoc dans notre cas) que l'interprète SGML utilisera pour la conversion vers les autres formats. Cette balise ne génère aucun texte en elle même.

Ensuite vous devez entourez le reste de votre travail entre `<article>` et `</article>`. Cela indique le début du contenu (ou de l'article, eh?). Si vous connaissez HTML, ces balises sont équivalentes aux balises `<html>` et `</html>`

4.4.2 Les informations de l'entête

La première partie du document devrait inclure des informations générales à propos du contenu. Elle est similaire aux premières pages d'un livre qui contiennent le titre de l'ouvrage, l'auteur, la date de publication, la table des matières, etc.

Le titre est indiqué entre les balises `<title>` et `</title>`. De même on utilise `<date>` et `</date>` pour la date.

Les deux sections qui restent sont les balises `<abstract>` et `</abstract>` qui donnent un résumé du document, et la balise `<toc>` qui indique l'emplacement de la table des matières. Celle ci est automatiquement générée par l'interprète SGML. Nous reviendrons sur les sections plus tard.

Maintenant, à quoi tout cela ressemble-t-il ? En regardant le début du code source de ce document, vous trouverez : (NdT : voir la version originale)

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

```
<!-- LinuxDoc file was created by LyX 1.0 (C) 1995-1999 by <markk> Tue Dec 14 16:17:42 1999
 -->

 <article>
 <title>HOWTO HOWTO
 </title>
 <author>Mark F. Komarinski &lt;markk@cgipc.com&gt;
 </author>
 <date>v1.1, 14 décembre 1999
 </date>
 <abstract>List the tools, procedures, and hints to get HOWTO authors up to speed
 and writing.
 </abstract>
 <toc>
```

Cette partie du document est utilisée pour créer la page principale que vous voyez une fois convertie au format RTF ou HTML.

4.4.3 Les sections

Pour générer la table des matières, il vous faut de quoi la construire. Les sections de SGML sont ce que les chapitres sont aux publications traditionnelles. Il peut y avoir plusieures sections, et chaque section peut avoir des sous-sections, qui peuvent elles aussi avoir des sous-sections etc.

Démarrer la rédaction de vos documents par les sections est une bonne idée puisque cela permet de dresser la liste des sujets que vous voulez traiter. Vous pouvez alors subdiviser ces sections principales en d'autres de plus en plus petites, jusqu'à ce que vous obteniez l'information pure que vous pourrez écrire en quelques paragraphes. J'ai moi même commencé ce document en utilisant cette méthode.

Les sections fait partie des quelques balises qui n'ont pas besoin d'être fermées. Ainsi, il n'y a pas de balise `</sect>`. Vous n'avez pas besoin non plus de vous occuper de la numérotation des sections. L'interprète SGML s'en chargera lors de la génération vers d'autres formats.

Les sections sont amorcées par la balise `<sect>`. Chaque nouvelle section commence par `<sect>`. La première est numérotée 1.

Les sous-sections (comme 1.1) se créent par la balise `<sect1>`. Elles commencent aussi à 1.

Les sous-sous-sections (1.1.1) se créent par la balise `<sect2>` et commencent aussi à 1.

Quand l'interprète SGML arrive à la balise `<toc>`, il parcourt le reste du document et construit la table des matières à partir des balises des sections qu'il rencontre. Les sections sont numérotées et listées dans la table des matières et, bien sûr, utilisées dans le reste du document. Les sous-sous-sections (1.1.1) n'apparaissent pas dans la table des matières, mais sont mises en valeur dans le texte si c'est possible.

4.4.4 Les paragraphes

L'écriture de paragraphes est la même qu'en HTML. Utilisez une balise `<p>` pour indiquer une nouvelle ligne, et commencez à écrire. SGML ignore les espaces tout comme les tabulations, les espaces multiples et les sauts de lignes. Quand SGML rencontre une balise `<p>`, il commence un nouveau paragraphe. Un document SGML correct devrait contenir les balises `</p>` pour finir les paragraphes.

4.4.5 Texte avancé

Vous aurez besoin de différencié des parties de texte par rapport à d'autres. Soit le mettre en valeur, soit pour `donner un nom de commande`. Le premier de ces deux cas se résout par les balises et . Quant au style machine à écrire, on utilise les balises <tt> et </tt>.

4.4.6 Les listes

Il existe deux types de listes sous SGML. La première est la liste numérotée, où chaque item de la liste est numéroté (comme les sections), en commençant à 1.

1. Voici la première entrée de la liste numérotée.
2. Voici la seconde.
3. Et la troisième.

Le code pour cette liste est le suivant :

```
<enum>
 <item>Voici la première entrée de la liste numérotée.
 <item>Voici la seconde.
 <item>Et la troisième.
</enum>
```

La balise <enum> indique que les entrées qui la suivent doivent être numérotées.

L'autre type de liste est la liste à items simple, où chaque entrée à une étoile, un cercle, ou un point, ou tout autre symbole pour indiquer chaque item.

- Voici la première entrée de la liste à items.
- Voici la seconde.
- Et la troisième.

Le code de la liste ressemble à ça en SGML :

```
<itemize>
 <item>Voici la première entrée de la liste à items.
 <item>Voici la seconde.
 <item>Et la troisième.
</itemize>
```

Comme vous le voyez, la balise <item> est la même pour les listes numérotées et les listes à items.

Une troisième forme de liste est la liste de définition. Elle comporte un terme à définir, et la phrase de définition.

LDP

Le Projet de Documentation Linux

SGML

Standard Generalized Markup Language

Le code qui permet de créer cette liste est :

```
<descript>
 <tag>LDP</tag>Le Projet de Documentation Linux
 <tag>SGML</tag>Standart Generalized Markup Language
</descript>
```

Ce n'est pas tout à fait la même syntaxe que pour les listes à items et numérotées, mais la liste est aussi entourée par des balises (`<descrip>` et `</descrip>`) et chaque item qui est un mot à définir est entouré par `<tag>` et `</tag>`. Le reste de la ligne est alors considéré comme la définition de ce dernier.

4.4.7 Texte verbatim

De temps en temps, on a besoin d'afficher du texte comme on l'écrit. Pour cela, vous pouvez utiliser les balises `<verb>` et `</verb>` pour entourer du texte qui doit apparaître tel quel. Les espaces, les retours à la ligne, et tout autre caractère spécial sont préservés jusqu'à la balise `</verb>`.

```
Ceci est du texte verbatim.

Et voici un autre texte verbatim.
```

4.4.8 Les URL

SGML fournit de quoi utiliser des URL (Universal Resource Locators , NdT: sortes de pointeurs vers des ressources extérieures) de tous types. Cela ne sera utilisé que par les versions HTML, mais d'autres formats pourrait également y avoir recours. La meilleure utilisation en sera faite pas HTML, mais d'autres formats, tels que PDF, pourront aussi en tirer avantages.

Les URL n'ont pas de balise de terminaison, mais toutes les informations sont contenues dans la balise `<url>` elle-même. Voici un URL qui pointe vers la page du LDP : *http://www.linuxdoc.org/*. Et voici le code pour le créer :

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

La partie `url="http://www.linuxdoc.org/"` donne la destination du pointeur, et la partie `name="http://www.linuxdoc` indique au navigateur ce qu'il doit afficher en réalité. Dans ce cas, les deux parties sont identiques, mais on pourrait créer une balise url qui ressemblerai à ça :

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

Ce qui affichera ceci dans le texte : *LDP*. Toutefois, un bon principe consiste à dupliquer l'URL dans la partie name. La raison est que si vous utilisez un format du type texte ou RTF, ces balises n'auront pas de significations. Le lecteur ne connaîtra pas l'URL à utiliser.

4.4.9 Les références

Alors que les URL sont adaptés pour faire référence aux ressources externes à votre document, ce n'est pas le cas pour les références au sein du texte lui-même. Pour cela, les balises `<label>` et `<ref>` sont préférables.

La balise <label> indique un endroit dans le document auquel vous voudrez faire référence ailleurs dans le texte, comme un signet. La création du <label> est simple. Insérez la ligne suivante à l'endroit voulu :

```
<label> id="introduction">
```

Vous avez alors créé un point dans le texte auquel vous pourrez vous référer en tant que "introduction". En effet, cette balise est utilisée dans le document SGML que vous lisez. Quand vous voulez pointer vers cet endroit (comme 1 (ici)), vous insérez le code SGML suivant :

```
<ref id="introduction" name="ici">
```

et SGML insérera le mot "ici" dans le texte qui sera un lien vers la section indiquée par le label "introduction".

L'autre utilisation des références est l'indexation. Puisque les documents du LDP sont souvent publiés sur papier sous forme d'un grand nombre de documents, les références servent à générer un index qui apparaîtra à la fin du livre, basé sur les mots et les sujets.

4.4.10 Les caractères spéciaux

Tout comme pour HTML, vous devrez interdire l'interprétation de certains caractères non alphanumériques pour éviter que SGML les voit comme du code. Voici une liste des codes utilisés. Vous en trouverez d'avantage dans le Guide de l'utilisateur de sgmltools à l'adresse *http://www.sgmltools.org/guide/guide.html*

- Utilisez & pour le et commercial
- Utilisez < pour le caractère inférieur à (<)
- Utilisez > pour le caractère supérieur à (>)
- Utilisez &etago; pour un inférieur à suivi d'une barre oblique (</)
- Utilisez $ pour le signe dollar ($)
- Utilisez # pour un dièse (#)
- Utilisez % pour le pourcent (%)
- Utilisez ˜ pour un tilde (˜)
- Utilisez " et " pour les apostrophes, ou utilisez &dquot; pour "
- Utilisez pour un trait d'union (c'est à dire une indication où le mot peut être coupé pour la justification)

4.5 Ecrire du SGML avec d'autres outils

4.5.1 LyX

Je vais encore chanter les louanges pour LyX. Je favorise cette application car je l'apprécie vraiment. Il donne la possibilité d'écrire en SGML avec la facilité d'un traitement de texte standard. Ce n'est pas un programme WYSIWYG, mais plutôt une application WYSIWYM (What You See Is What You Mean, NdT: Ce que vous voyez est ce que vous pensez), puisque ce que vous voyez n'est pas forcement ce que vous obtiendrez une fois que l'interpréteur SGML aura fait son travail.

Pour créer un document LinuxDoc avec LyX, téléchargez et installez l'application. Assurez-vous d'avoir déjà installé TeX et sgmltools (voir 2.3 (Télécharger et installer les outils) pour plus d'informations à ce propos). Ensuite, lancez LyX et sélectionnez "file->new from template...". Cliquez sur "Templates" et sélectionnez `linuxdoctemplate.lyx` et vous obtiendrez un document de base, avec tous les en-têtes d'information qu'un document du LDP se doit d'avoir. Modifiez les données selon vos besoins (c'est à dire compléter les champs

Titre, Auteur, Date, Abstract, etc.) en commencez la rédaction de votre document. Le menu dans le coin en haut à gauche vous permet de sélectionner le type du texte (standard, liste énumérées ou à items, sections). Le point d'exclamation est utilisé pour mettre le texte en valeur, et vous pouvez soit cliquer dessus et commencer à taper le texte, soit sélectionner du texte et cliquer dessus pour mettre en valeur du texte déjà écrit. D'autres spécificités de SGML peuvent être trouvées dans le menu Insertion. Vous pouvez insérer des URL, des références, des entrées d'index, et d'autres types de données. Une fois votre document terminé, vous pouvez le sauvegarder au format LyX, et l'exporter au format LinuxDoc et obtenir ainsi un fichier avec l'extension .sgml. sgmlcheck peut alors vérifier ce fichier, qui est prêt à être converti vers d'autres formats.

4.5.2 Emacs

Emacs dispose d'un mode spécial pour écrire en SGML appelé psgml. Ceux qui ont une expérience de ce mode sont invités à donner des informations à l'auteur par courrier électronique. psgml est un module majeure pour Emacs conçu pour éditer des documents SGML et XML. Il permet une coloration syntaxique et un joli affichage qui font ressortir les balises SGML, il fournit une méthode dínsertion des balises sans les taper à la main, et est capable de valider la syntaxe de votre document lors de sa rédaction. Pour les utilisateurs d'Emacs, c'est un excellent outil. Je pense qu'il permet une plus grande versatilité que tout autre éditeur de code SGML. Il fonctionne aussi bien avec DocBook, LinuxDoc et d'autres DTD. La documentation psgml est disponible à *http://www.lysator.liu.se/˜lenst/about_psgml/* `<http://www.lysator.liu.se/~lenst/about_psgml/>`.

4.5.3 Les autres outils SGML

S'il y a d'autres outils permettant d'utiliser la DTD LinuxDoc pour générer des documents du LDP, faites le moi savoir.

4.6 Les bases de CVS

Le LDP est en train de mettre un place un accès CVS pour les auteurs. Il y a, en effet, de bonnes raisons d'utiliser CVS :

1. CVS gère une sauvegarde des documents. Si vous passer un document à un autre auteur, il peut récupérer le document depuis CVS et continuer à travailler dessus. Si vous avez besoin de revenir sur une ancienne version, vous pouvez aussi la récupérer.
2. C'est formidable si plusieurs auteurs travaillent sur le même document. Vous pouvez demander à CVS de vous indiquez qu'elles modifications ont étées faites pendant que vous travailliez sur le document, et directement intégrer ces changements.
3. CVS garde un rapport des modifications du document. Ce rapport peut être placé automatiquement dans le fichier si vous utilisez certaines balises qui seront analysées avant l'interpréteur SGML.
4. Il peut servir, grâce à un programme, à mettre à jour le site web du LDP automatiquement, dès qu'un document a été terminé et reçu. Ce n'est pas encore en place, mais ça ne va pas tarder.

Si CVS est quelque chose de nouveau pour vous, voici quelques pages web qui pourront vous aider :

1. *http://www.sourcegear.com/CVS/Docs/blandy* `<http://www.sourcegear.com/CVS/Docs/blandy>`
2. *https://wroclaw.art.pl/˜ser/docs/cvs.html* `<https://wroclaw.art.pl/~ser/docs/cvs.html>`

4.7 Obtenir un compte CVS

D'abord, il vous faudra obtenir un compte dans le repository CVS du LDP (NdT: lieu de stockage et de dépôt des documents pour CVS). C'est souvent le répertoire racine qui est utilisé par CVS, où chaque projet (HOWTO, Mini-HOWTO, ...) dispose d'un sous-répertoire.

Vous devrez créer un mot de passe crypté et un identifiant d'utilisateur pour votre compte. Ce mot de passe vous permet d'envoyer un mot de passe crypté au groupe CVS sans qu'ils aient besoin de connaître votre mot de passe. Vous pouvez le faire par les commandes suivantes, depuis un shell bash (ou sh) :

```
$ echo votre_mot_de_passe | perl -e "print crypt(<>, join
'',('.', '/', 0..9, 'A'..'Z', 'a'..'z')[rand 64, rand 64]),\"n\""
```

(NdT: je conseille un

```
; echo
```

si vous exécutez ceci depuis un shell et que la réponse ne semble pas s'afficher)

Envoyez la sortie de cette commande avec l'identifiant d'utilisateur à *cvsadmin@cvslist.linuxdoc.org* `<mailto:cvsadmin@cvslist.linuxdoc.org>`. Votre CVSROOT unique sera créer et vous recevrez un e-mail avec la réponse.

Quand vous obtiendrez la réponse, connectez vous sur votre CVSROOT et vérifiez que tout est configuré correctement :

```
$ export CVSROOT=:pserver:your_userid@cvs.linuxdoc.org:/cvsroot
$ cvs -d $CVSROOT login
```

(Remplacez CVSROOT par ce qui vous a été indiqué dans la réponse.)

On vous demandera votre mot de passe, et vous aurez accès au repository CVS en mode lecture-écriture. Une fois que vous aurez utilisé login et obtenu accès au système, votre mot de passe est stocké dans .cvsroot et vous n'aurez plus besoin d'utiliser cvs login. Positionner CVSROOT correctement et c'est parti.

Vous pouvez obtenir le repository linuxdoc en entier avec cette commande :

```
$cvs get linuxdoc
```

Ou vous pouvez obtenir le fichier source SGML de votre propre document par :

```
$ cvs get linuxdoc/src/VOTRE-HOWTO.sgml
$ cvs get linuxdoc/minisrc/VOTREDOC.sgml
```

Une liste des changements est également disponible. C'est un e-mail envoyé à chaque changement dans le repository. Remarquez que celà peut devenir une liste à très grand débit. Vous pouvez vous y abonner en envoyant un e-mail vide à *commits-subscribe@cvslist.linuxdoc.org*. Vous pouvez annuler votre abonnement en envoyant un e-mail vide à *commits-unsubscribe@cvslist.linuxdoc.org* .

4.8 Autres informations sur CVS

4.8.1 Accès CVS anonyme

L'accès CVS anonyme (en lecture seule) est disponible par :

```
$ cvs -d :pserver:cvs@anoncvs.linuxdoc.org:/cvsroot login
```

Utilisez "cvs" comme mot de passe. Vous pouvez alors accèder aux modules linuxdoc comme décrit ci-dessus. Notez que les changement apparaissent sur le site cvs anonyme environ une demi-heure après le site principal.

4.8.2 Fichiers CVS via le web

Vous pouvez accèder au repository CVS par le web à l'adresse *http://cvsweb.linuxdoc.org/index.cgi/linuxdoc* `<http://cvsweb.linuxdoc.org/index.cgi/linuxdoc>`.

4.8.3 Accès graphique à CVS

Il existe des interfaces graphiques pour CVS, et vous en trouverez une liste sur le site *http://freshmeat.net/appindex/* `<http://freshmeat.net/appindex/>`. Cherchez CVS.

4.9 CVS et la mise à jour des fichiers

CVS reconnait une balise spéciale que vous pouvez utiliser pour insérer la date et la version automatiquement dans votre document. C'est la balise Id. En mettant cette balise dans la section <date> (par exemple), elle sera modifiée à chaque changement du document, permettant une incrémentation automatique de la version.

Quand vous voulez copier votre fichier modifié sur le serveur CVS, utilisez la commande `cvs ci -m "commentaires" YOUR-HOWTO.sgml`. Le paramètre `-m "commentaires"` n'est pas obligatoire, mais si vous ne le mettez pas, vous serez ammené dans votre éditeur (certainement vi, ou l'éditeur indiqué par la variable d'environnement EDITOR) et devrez taper un commentaire à propos des changements. Si vous décidez d'utiliser un éditeur plutôt que de spécifier le commentaire sur la ligne de commande, il se peut qu'il faille quelques (plus de 5) secondes pour finir ce qu'il y avait à faire et effectuer la mise à jour.

Vous pouvez suivre toutes les discussions à propos de CVS sur la liste ldp-discuss. Pour l'instant, les soumissions LDP doivent toujours être envoyées à ldp-submit.

5 Distribuer votre documentation

5.1 Avant la distribution

Avant de distribuer votre prose à des millions de lecteurs potentiels, il y a quelques petites choses à faire.

D'abord, vérifiez le bon français de votre texte. La plupart des outils que vous utiliserez pour écrire en SGML (Emacs, LyX, et les autres éditeurs de texte) disposent de modules de vérification de l'orthographe. Si ce n'est pas le cas, il y a toujours ISPELL, disponible dans toutes les distributions. Utilisez aussi sgmlchek pour vérifier la cohérence de vos balises SGML.

Deuxièmement, faites relire votre document par quelqu'un d'autre que vous pour obtenir des commentaires et d'éventuelles corrections. La documentation publiée par le LDP a besoin d'être la plus correcte possible, car des millions d'utilisateurs de Linux peuvent avoir besoin de la lire. Si vous faites partie d'une liste de discussion parlant du sujet traité, demandez aux autres intervenants de vous aider.

Troisièmement, créez un site web où vous pourrez distribuer vos réalisations. Ce n'est pas indispensable, mais très utile pour retrouver la source originale du document.

5.2 Notions de copyright et de licence

Pour qu'un document soit accepté par le LDP, il doit utilisé une licence qui en autorise une distribution et une publication libre. En tant qu'auteur, vous pourrez conserver le copyright et ajouter d'autres restrictions (par exemple, vous devez approuver toutes traductions ou travaux dérivés). Une version simple d'une licence est disponible à l'adresse *http://www.linuxdoc.org/COPYRIGHT.html*. Si vous choisissez d'utiliser ce copyright, placez le dans votre code source dans une section intitulée "Copyright et licence" où quelque chose dans ce genre. Incluez également une clause de propriété (puisque vous en serez le propriétaire). Si vous êtes un nouveau responsable d'un HOWTO qui existait déjà, vous devez inclure les clauses de copyright de(s) l'ancien(s) auteur(s) et la période où ils ont maintenu le document, à moins qu'ils aient distribué leur document dans le Domaine Publique quand ils en ont arrêté la maintenance.

5.3 Soumission au LDP Une fois que votre document a été relu et que vous avez pris en compte les commentaires des relecteurs, vous pouvez soumettre votre texte au LDP. Envoyez un e-mail à *ldp-submit@lists.linuxdoc.org*. Indiquez le nom du HOWTO dans le sujet du mail, et décrivez les changements que vous avez fait (réécriture majeure ou corrections de fautes d'orthographe ...) dans le corps du message, en y joignant votre document. Celà permet aux mainteneurs de faire leur travail plus rapidement, et de ne pas avoir à trop attendre avant que votre HOWTO ne soit disponible sur le site web du LDP. Si vous n'avez pas de réponse dans les 7 jours suivant, renvoyez un mail pour savoir si votre HOWTO à bien été traité.

6 Conseils de style

Ceci n'est pas un guide complet sur comment écrire une bonne documentation (pour l'instant), mais considérez le comme une liste rapide de conseils pour vous aider tout au long de la rédaction.

- Etre clair. Tout le monde doit comprendre de quoi vous parlé.
- Utiliser des exemples quand c'est possible. Cela aide à comprendre de quoi vous parlé.
- Etre organisé. Ne pas sauter à des sujets sans lien dans une même section.

Vous pourrez obtenir beaucoup d'autres conseils sur le style du LDP à l'adresse *http://www.linuxdoc.org/HOWTO/LDP-Style-Guide.html*.

7 FAQ à propos du LDP

7.1 Je désire aider le LDP. Comment puis-je le faire?

La façon la plus simple, c'est de trouver un sujet et d'en faire un document. Regardez également la liste des HOWTO non maintenus et voyez s'il n'y en a pas un que vous pourriez continuer.

7.2 Je voudrais publier un ensemble de documents du LDP dans un livre. Comment est protégé le contenu du LDP ?

Référez vous à la page *http://www.linuxdoc.org/COPYRIGHT.html*.

7.3 J'ai trouvé une erreur dans un document du LDP. Puis-je le corriger ?

Contactez l'auteur du document, où le coordinateur et informez le du problème en lui donnant une éventuelle correction.