# HOWTO HOWTO

作者: Mark F. Komarinski *markk@cgipc.com*
譯者: 蘇江文 *softsprite@yeah.net*
v0.13, 19 September 1999 翻譯日期: 09 Feb 1998

成為一個新的 LDP 作者，掌握 LDP 使用的工具、思路和約定。

## 目錄

# 1 概述

## 1.1 歷史

本文是1999年8月26日 Mark F. Komarinski *markk@cgipc.com* <mailto:markk@cgipc.com> 在花費了兩天找尋工具的代價後開始寫作的。如果任何一個 LDP 作者能從中得到幫助，我就達到我的願望了。

## 1.2 新版本

本文的最新版本可以在我的主頁 *http://www.cgipc.com/~markk* <http://www.cgipc.com/~markk> 找到 SGML 源文件。 其它不同格式的版本可以在 LDP 主頁 *http://www.linuxdoc.org/* <http://www.linuxdoc.org/> 找到。

### 1.2.1 版本歷史

v0.25, 20 September 1999

- 修復了一些斷開的鏈接
- 爲新作者增加章節
- 從最初發行版本增加一些雜項修改
- 增加版權説明取代以前的 URL 鏈接

v0.12, 2 September 1999

- 完成大多數章節
- 討論列表的綜合修改

v0.10, 27 August 1999

- Got up to section 3.4 written
- Added to the outline some
- Changed location of LDP mailing list to lists.debian.org from thepuffingroup.com.

v0.01, 27 August 27 1999

- 初步完成，上傳主頁，簡單大綱寫作
- Take some of what I wrote with a grain of salt. Some things need to be verified.

## 1.3 版權和商標

### 1.4 感謝

感謝所有在我寫作過程中給予我幫助的人，包括 Deb Richardson 和 Daniel Barlow 以及 LDP 討論列表上的其它人。

部分章節我取自 HOWTO Index 和 sgmltools 文檔。在本文的其它部分有指明到何處取得 sgmltools 和 LDP。

## 2 LDP 和 SGML 背景資料

### 2.1 LDP

Linux 文檔計劃 (LDP) 是用來提供給新用户以快速取得某個詳細主題相關信息。它並不僅僅包含一系列關於管理、網絡、編程等的書籍，而是包容了大量的由使用過的人書寫的關於某個個別主題的解決方法。如果你想查找關於打印的資料，你可以查閱 Printing HOWTO。如果你想查找關於網絡的資料，你可以查閱 Ethernet HOWTO，等等。

剛開始，許多這些文檔是文本或 HTML 格式。隨著時間的推移，必須找出一個管理這些文檔的好方法。一種能讓你從 Web 網頁上閱讀、從光盤的文本文件閱讀，甚至從你的個人數字助理上閱讀的方法。答案已經有了，就是 SGML。

### 2.2 SGML

Standard Generalized Markup Language (SGML) 是一種基於記號文本的語言。在這方面，它類似於 Tex 或者 groff 或者 HTML。 SGML 的強大之在於它不象 WYSIWYG (所見即所得)那樣，你不需要定義象顏色、字體等格式，你只需定義元素(段落、章節、編號列表)而 讓最終處理程序來處理象位置、顏色、字體等煩心的事情。HTML 也是這麼做的，實際上它是 SGML 的一個子集。

SGML 包含兩部分。首先是一個稱爲 DTD (或稱爲文檔類型定義 Document Type Definition) 的結構。DTD 定義每一個元素之間的關系。The LinuxDoc DTD, used to create this document, is an example of this. DTD 爲每一個使用 DTD 創建的文檔一個通用的視覺和感覺效果。其次是內容(content)，就是將要由 SGML 處理器處理並被用户閱讀的部分。本段落就是內容，但也可以包含圖像、表格、編號列表等等。內容被標記(tag)包圍用來分開不同的元素。

隨著時間的推移，LinuxDoc DTD 將被 DocBook DTD 所取代，爲其它的 SGML 文檔提供一個統一的視覺和感覺效果。 如果實現了這一步，我們將通過本 HOWTO 或者郵件列表爲你升級。 LinuxDoc 和 DocBook 之間最大的不同在於 DocBook 分配標記給不同類型的內容(就象命令、文件名、地址目錄等等)，而 LinuxDoc 分配標記基於文本將要顯示的方式(比如你可以分配強調或打字)

#### 2.2.1 爲什麼用 SGML 取代 HTML 或其它格式？

SGML 提供的不僅僅是格式。你可以自動建立索引、內容表以及文檔中的鏈接或者是大綱。sgmltools 包也可以讓你輸出 LaTeX、 info、 text、 HTML 和 RTF。從這些基本格式，你可以創建其它格式文檔(DOC、PostScript等等)。SGML 不會象 HTML 那樣冗長。我不認爲你很快會在 SGML 中看到閃亮的標記。這有助於編碼不僅僅易於處理，而且易於書寫。象 LyX (我至今使用的所見即所得編輯器)這樣的軟件可以幫助你書寫 TeX格式文檔並輸出成 SGML 格式，然後你就可以從 SGML 生成任何你需要的格式。

最後，SGML 更注重元素的工作性能而不是元素的顯示。這是一個顯著的特性，有助你更快地書寫，因爲你不必關心段落的位置、字體大小等等。

## 2.3 工具

在本章節，我將介紹一些用來創建你自已的 LDP 文檔的工具。我將先簡單介紹，然後詳細說明，包括如何安裝它們。如果你使用其它的一些工具來書寫 LDP，請告訴我，我會在這裡爲它們做廣告的。

### 2.3.1 sgmltools

必備

sgmltools 包包含用來處理 SGML 成以上所說格式的 SGML 工具。它也包含創建 LDP 文檔所需的 LinuxDoc DTD。如果只創建 SGML 文檔，這些已經足夠了。如果你想處理成比如 TeX 格式，你還需要一些工具包。sgmltools 包在你的 linux 發行版本中可以找到，或者到 *http://www.sgmltools.org/* <http://www.sgmltools.org/> 下載。

### 2.3.2 TeX

可選

TeX (rhymes with blech!) 是一種可供廣泛選擇的標記語言，甚至包括那些數學世界中的人們。 我還記得許多微積分考試是由 TeX 寫成的。它也是至今仍在使用的最早的標記語言之一。 TeX 實際上和 SGML 具備許多同樣的概念。TeX 處理它的文件成 DVI (設備無關)從而可能處理成其它格式。 不幸的是，DVI不能容易地轉換成打印語言(PostScript、PCL)以外的其它格式，很難生成 HTML。 TeX 在多數 Linux 發行版本中都已安裝或被提供。

### 2.3.3 LyX

可選

LyX 程序是一個圖形化所見即所得軟件，提供非常必需的易用圖形應用與 SGML 處理器之間的聯系。LyX是用於書寫 TeX 文檔，實現了許多 TeX 規則。舉個例子，當章節被自動編號，你無法輕易地插入空格。這被 TeX 設計成禁止。而 SGML 通常忽略同樣的空格。LyX能夠讀入 LinuxDoc DTD 並且提供文檔模板以幫助你用習慣的方式創建你的 LDP 文檔，而不需要象 vi 那樣要記住許多標記。 LyX 可以在 *http://www.lyx.org/* <http://www.lyx.org/> 找到。

在 KDE 中有一個使用 Qt 庫的 LyX。 你可以從以下地址得到更多信息： *http://www.devel.lyx.org/˜ettrich/klyx.html* <http://www.devel.lyx.org/~ettrich/klyx.html>。如果你使用 KLyX 來寫 SGML，請通過 郵件 <mailto:markk@cgipc.com> 告訴我你的使用經驗。

# 3 讓我們開始

本章節顯示如何獲取關於如何書寫你自己的 LDP 文檔的信息。獲取並安裝工具，與 LDP 聯系，從那裡向所有 Linux 用户發布你所掌握的知識。

## 3.1 寫給新作者

如果你剛剛了解 LDP 並且想要選擇一個沒人維護的 HOWTO 或者想寫一個新的 HOWTO 或者 mini-HOWTO 文檔，請聯系 *linux-howto@metalab.unc.edu* <mailto:linux-howto@metalab.unc.edu>。

這將讓 HOWTO 協調人掌握誰在維護哪篇文檔。同時要注意所有的 HOWTO 都使用 SGML 格式(當前使用 LinuxDoc DTD)。mini-HOWTO 使用 SGML 或者 HTML 格式，但只有 SGML 格式將被包含進 HOWTO 的打印版本中。

## 3.2 郵件列表

有一些郵件列表可以訂閱以讓你了解 LDP 目前工作情況。首先是 *ldp-discuss@lists.linuxdoc.org* <mailto:ldp-discuss@lists.linuxdoc.org>，這是 LDP 的主要討論組。如果要訂閱，發送主題爲 "subscribe" 的郵件到 *ldp-discuss-request@lists.linuxdoc.org* <mailto:ldp-discuss-request@lists.linuxdoc.org>。退訂則發送主題爲 "unsubscribe" 的郵件到 *ldp-discuss-request@lists.linuxdoc.org* <mailto:ldp-discuss-request@lists.linuxdoc.org>。

## 3.3 下載並安裝工具

### 3.3.1 sgmltools

從 *http://www.sgmltools.org/* <http://www.sgmltools.org/>下載 sgmltools 包，或者直接從你的 Linux 發行版本中取得。從 sgmltools.org 取得的軟件是源碼格式，因此你必須在你的機器上編譯這些源碼。從你的 Linux 發行版本中取得的預建立好的包更容易些，你可以不用編譯而直接開始編輯工作。(特別是如果你不是一個程序員)

RedHat 中包含 sgmltools。 如果沒有，你可以從 ftp.redhat.com 或者其它鏡像站點下載。

如果你使用 Debian，在標準發行版本中也包含 sgmltools。如果你沒有已安裝的包，你可以使用 apt-get 命令下載並安裝：

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

要了解更多關於 Debian 包的信息，你可以查閱 *http://www.debian.org/Packages/stable/text/sgml-tools.html* <http://www.debian.org/Packages/stable/text/sgml-tools.html>

如果從源碼編譯，所有你需要做的是：

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

替換 `sgmltools-x.x.x` 成你當前 sgmltools 的版本號。支持 LinuxDoc 的當前版本號是 1.0.9。支持 DocBook 的當前版本號是 2.0.2。兩者在以上所述站點中都可以找到。

一旦工具安裝好後，你就可以使用很多命令了。

`sgmlcheck file.sgml`- 檢查給定文檔的句法。

`sgml2html file.sgml`- 轉換 SGML 文件成 HTML。 創建一個名爲 `file.html` 的文件包含內容表，然後創建一系列名爲 `file-x.html` 的文件，這裡 `x` 是章節號。

`sgml2rtf file.sgml`- 轉換 SGML 文件成 Rich Text Format (RTF)。創建兩個文件，第一個是 `file.rtf` 包含 TOC，以及 `file-0.rtf` 包含所有的章節。

`sgml2txt file.sgml`- 轉換 SGML 文件成 ASCII 文本。所有內容放在 `file.txt`中。

`sgml2info file.sgml`- Blah SGML blah INFO, 能被 info 命令使用。所有的輸出傳送到 `file.info`。

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

`sgml2lyx file.sgml`- SGML yadda LyX graphical editor. This is great if you have pre-generated SGML files and want to convert them for use in LyX.

## 3.4 手工書寫 SGML

非常象 HTML，你可以手工書寫 SGML , 只要你掌握了所有需要使用的標記代碼本章節將講述盡可能多的這些代碼，並爲每一個提供例子。開始的最好的地方就是本文的源碼，可以從這裡看看： 1 (Introduction). 因爲 SGML 的處理方式依賴於文件格式的不同而不同，我將盡可能說明在文件寫作中應該知道的事情。

### 3.4.1 出發

現在開始寫作了，首先用你喜愛的編輯器創建一個新文件，文件的開頭如下：

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

這將定義 SGML 處理輸出文件時使用的文檔類型(我們這個例子使用 LinuxDoc)。這個標記本身不產生任何輸出。

接下來你需要把你其余的內容包含在標記 `<article>` 和 `</article>` 之中。這意味著內容的開始。如果你熟悉 HTML 的話，這類似於將所有內容包含在標記 `<html>` 和 `</html>` 之間。

### 3.4.2 頭部信息Header information

內容的第一部分應該包含內容其余部分的概括信息。這類似於一本書的頭幾頁，那裡有標題頁(書本的標題、作者、出版日期、內容目錄等等)。

內容的標題包含在標記 `<title>` 和 `</title>` 之間。作者包含在標記 `<author>` 和 `</author>` 之間。日期則使用標記 `<date>` 和 `</date>`。

有兩個保留的章節是： `<abstract>` 和 `</abstract>` 標記描述內容摘要；`<toc>` 標記指出內容目錄的位置。內容目錄(TOC)是由 SGML 自動生成的。我們後面將引入章節。

現在我們所做的一切看起來是什麼樣子呢？把剛才說過的 SGML 代碼組合起來，就象：

```
<!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>Aug 27, 1999
</date>
<abstract>Getting a new LDP author up and running with tools, ideas,
 and conventions used by the LDP
</abstract>
<toc>
```

當你使用 RTF 或 HTML 格式看這段創建主頁面的內容時，所有的信息都在一頁中。

#### 3.4.3 章節 (Sections)

爲了建立內容目錄，你必須做一些標記。 在 SGML 中的章節(Sections)類似於傳統出版中的章的概念。你可以有多個章節，每個章節可以有子章節，子章節下還可以有子章節。

在你的文檔中使用章節，有助於你建立主題大綱。你可以將主題分隔成多個小章節。我在寫本文時就是這樣做的。

章節是 SGML 中少數幾個不需要成對出現的標記。也就是說，沒有 </sect> 標記。 你不用關於章節的編號，SGML 在輸出成其它格式時會自動處理的。

章節是使用 <sect> 標記作爲開始的。每一個新章節開始於一個 <sect> 標記。 第一個章節被編號爲 1。

創建子章節 (比如 1.1) 可以使用 <sect1> 標記。它也從 1 開始編號。

子章節 (1.1.1) 使用 <sect2> 標記，並且它也從 1 開始編號。

當 SGML 處理器處理到 <toc> 標記時，它開始掃描文檔的其余部分，利用其中的章節標記編號創建文檔目錄。章節被編號排列在內容目錄(TOC)中，然後被文檔的其余部分使用。子章節 (1.1.1) 並不出現在 TOC 中，但如果可能的話會被置成強調文本格式。

#### 3.4.4 標準段落

寫內容段落就象 HTML一樣。使用一個 <p> 標記指明一個新行的開始。SGML 會忽略 TAB、空格、空行。當 SGML 檢查到 <p> 標記時，它開始一個新行。放置 </p> 標記結束該段落。

#### 3.4.5 增強文本

你偶而會需要讓一些文本明顯不同於其它文本。可能是重要代碼或者命令列表。強調文本可以使用 <em> 和 </em> 標記。斜體字可以使用 <tt> 和 </tt> 標記。

#### 3.4.6 列表

在 SGML 中有兩種形式的列表。第一種是列舉列表，從1開始爲每一個列表項編號。

1. This is the first entry in the enumerated list.
2. This is the second.
3. Third.

以上的代碼如下：

```
<enum>
<item>This is the first entry in the enumerated list.
<item>This is the second.
<item>Third.
</enum>
```

<enum> 標記指出緊隨其後的列表項要開始編號。

另一種形式是逐項列表，每一個列表項前面只放一個星或圓圈或點或其它東西。

- This is the first entry in the itemized list
- This is the second
- Third

以上的代碼如下：

```
<itemize>
<item>This is the first entry in the itemized list
<item>This is the second.
<item>Third.
</itemize>
```

就象你所看到的，<item> 標記在兩種形式列表中都一樣。

第三種形式是描述列表，包括被描述的術語以及描述該術語的短語。

**LDP**

The Linux Documentation Project

**SGML**

Standard Generalized Markup Language

以上的代碼如下：

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

這種形式不同於前面兩種，整個列表包含在標記 <descrip> 和 </descrip> 中，每一行中的每一項，也就是定義的短語，被包含在 <tag> 和 </tag> 中。行的剩余部分是對短語的定義。

#### 3.4.7 逐字文本

有時你需要完全顯示你所寫的內容，你可以使用 <verb> 和 </verb> 標記來包含段落。在</verb>標記之前的空格、回車以及其它文本(包括特別字符)都將被保留。

以下就是逐字文本

#### 3.4.8 統一資源定位(URL)

SGML 也允許統一資源定位(URL)。注意這在輸出成 HTML 格式時才有用，其它格式可能也有一些用，比如說 RTF 格式。

一個 URL 沒有結束標記，所有信息都放在 <url> 標記裡面。這是一個指向 LDP 主頁的 URL：*http://www.linuxdoc.org/* <http://www.linuxdoc.org/>。以下是代碼：

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

url="http://www.linuxdoc.org/" 告訴瀏覽器目標地址，而 name="http://www.linuxdoc.org/" 則告訴瀏覽器如何顯示。在本例中兩者是一樣的，但我可以創建一個如下的 URL 標記：

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

在頁面上的顯示就象這樣： *LDP* <http://www.linuxdoc.org/>.

#### 3.4.9 參考(References)

URL 適合於鏈接到 LDP 文檔之外的內容，但不適合於鏈接到文檔內容自身。爲實現這一功能，可以使用 <label> 和 <ref> 標記。<label> 標記創建一個將來能跳轉回來的記號，非常象一個書簽。創建 <label> 很簡單。找到以後需要跳轉回來的位置，然後插入以下：

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

你現在已經在文檔內容中創建了一個可以跳轉回來的記號 "Introduction"。這個標簽確實出現在文檔的開頭。當你以後需要跳轉回來時，插入以下：

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

SGML 就能明白放入一個叫 "here" 的鏈接可以跳轉至 Introduction 章節的位置。

參考的另一部分是索引 (indexing)。由於 LDP 文檔經常是大量文檔，必須在書的後面建立基於單詞和主題的索引。

#### 3.4.10 特殊字符(Special characters)

非常象 HTML，你必須避免許多非文字字符以防止 SGML 將其解釋成 SGML 代碼。這裡有一個 SGML 代碼的列表。更多的可以查閱 sgmltools 用户指南：*http://www.sgmltools.org/guide/guide.html* `<http://www.sgmltools.org/guide/guide.html>`。

- Use & for the ampersand (&)
- Use < for a left bracket (<)
- Use > for a right bracket (>)
- Use &etago; for a left bracket with a slash (</)
- Use $ for a dollar sign ($)
- Use # for a hash (#)
- Use % for a percent (%)
- Use ˜ for a tilde (˜)
- Use “ and ” for quotes, or use &dquot for "
- Use  for a soft hyphen (that is, an indication that this is a good place to break a word for horizontal justification).

## 3.5 使用其它工具書寫 SGML

#### 3.5.1 LyX

我一直十分喜愛 LyX，因此我偏愛這個軟件。它提供常規字處理器易於使用的方式書寫 SGML。它不是一個所見即所得程序，而是所得即所想應用。

要用 LyX 創建一個 LinuxDoc 文檔，下載並安裝應用程序。確保你已安裝 TeX 和 sgmltools (參見 3.3 (安裝工具) )。然後啓動 LyX 並且選擇 "file->new from template..."。選擇 "Templates" 然後點擊 `linuxdoctemplate.lyx`，這樣你將得到一個帶 LDP 文檔頭信息的文檔模板。更改合適你的數據(也就是，填寫標題、作者、日期、摘要等等)，然後開始寫作。左上角的下拉菜單可以用來選擇內容的類型(標準、列表、章節)。感嘆號用來強調文本。你既可以點擊它進入強調模式，也可以用鼠標高亮選擇文本再點擊它，將選擇的文本置爲強調文本。許多其它的 SGML 特性可以在 Insert 菜單條中找到。 你可以插入 URL、交叉參考、索引以及其它類型的數據。當完成你的文檔後，你可以保存成 LyX 格式，然後輸出到 LinuxDoc 得到 .sgml 後綴的文件。這個文件可以被 sgmlcheck 檢查和輸出成你想要的格式。

#### 3.5.2 Emacs

我不使用 Emacs，但我並不排斥它。有 Emacs 使用經驗的人會覺得它非常有用。

#### 3.5.3 其它 SGML 工具

如果有除此之外的其它 SGML 工具，或者即使是商業版本，只要是能被 LinuxDoc DTD 用來創建 LDP 文檔，請告訴我。

### 3.6 CVS 基礎

到現在爲止，LDP 還沒有一個共享倉庫可供你在線保存你的文檔內容。非常希望改變這種狀況。有很多需要使用 CVS 的原因：

1. CVS 將保存一個你的文檔的外部備份。在你將文檔傳給另外一個作者時，他只需要從 CVS 取得文檔就可以繼續工作。在你需要取得一個文檔的早期版本，你也可以很快得到。
2. 在多人維護同一文檔時非常有用。你可以從 CVS 得知在你編輯你的拷貝時其他作者做了哪些修改，從而將所有的修改很好地結合在一起。
3. 保存修改的記錄。在 SGML 處理器處理之前，你可以使用一些特殊的標記，這些記錄(以及一個時間戳)會被自動地插入到文檔中。
4. 提供一個途徑，在新文檔完成並提交後通過一個程序自動地更新 LDP 站點。

### 3.7 發布你的文檔

#### 3.7.1 發布之前

在你將文檔發布給成千上萬的潛在讀者之前，我必須做一些事情。

首先，對你的文檔做拼寫檢查。沒有什麼比拼寫錯誤更快讓人覺得是在互聯網世界中說“嘿，我是傻瓜”。許多你可以用來寫 SGML 的應用程序(emacs、LyX...)都內置有拼寫檢查功能。如果沒有，就在所有的linux發行版本中都安裝了一個 ispell 程序。同樣要使用 sgmltools 中的 sgmlcheck 命令檢查 SGML 標記。

其次，找一些人評價你的文檔。由 LDP 發布的文檔應盡可能保証眞實、正確，因爲有成千上萬的Linux用户可能會讀它。如果你參加了一個相關主題的較大規模的郵件列表，可以請求列表中的其他人幫助你。

第三，建立一個發布你的文檔的網站。這並不是必須的，但有助於人們找到你的文檔的原始出處。

#### 3.7.2 版權和許可

In order for an LDP document to be accepted by the LDP, it must be licensed to allow for free (as in beer) distribution and publishing. As an author, you may retain the copyright and add other restrictions (for example, you must approve any translations or derivative works). A sample license is available at *http://www.linuxdoc.org/COPYRIGHT.html* `<http://www.linuxdoc.org/COPYRIGHT.html>`. If you choose to use the boilerplate copyright, simply copy it into your source code under a section called "Copyright and

Licenses" or similar. Also include a copyright statement of your own (since you still own it). If you are a new maintainer for an already-existing HOWTO, you must include the previous copyright statements of the previous author(s) and the dates they maintained that document.

#### 3.7.3 遵從 LDP

一旦你的文檔被一部分人看過並且你已吸收了他們的意見，你就可以將文檔送往 LDP。發送郵件到 *ldp-submit@lists.linuxdoc.org* <mailto:ldp-submit@lists.linuxdoc.org>。24小時之內你就可以知道它是否已被接受並送往主要的 LDP 站點。

# 4 風格指南

這並不是一個寫出色文檔的強制和快速的指南，但對你的寫作有所幫助。

- 清晰。所有人需要知道你在說什麼。
- 在可能的地方使用例子。這讓所有人看到你在說什麼。
- 組織。不要在同一章節中，在兩個不相關主題間跳來跳去。

你可以從以下地址得到更多的 LDP 風格指南： *http://www.linuxdoc.org/HOWTO/LDP-Style-Guide.html* <http://www.linuxdoc.org/HOWTO/LDP-Style-Guide.html>.

# 5 關於 LDP 的常見問題

## 5.1 我想爲 LDP 出一份力，我該如何去作？

最簡單的辦法就是找出一些東西然後寫出來。也可以查找沒人維護的 HOWTO 然後看看是否有一個你了解的主題可以繼續寫下去。

## 5.2 我想出版 LDP 文檔。LDP 的文檔許可是怎樣規定的？

請參閱 *http://www.linuxdoc.org/COPYRIGHT.html* <http://www.linuxdoc.org/COPYRIGHT.html>.

## 5.3 我在一個 LDP 文檔中找到一個錯誤。我能修正它嗎？

聯系文檔的作者，或者 LDP 組織者，告訴他們存在的問題以及如何修正。