LE BON OUVRIER SE PROCURE DE BONS OUTILS (LIBRES ET GRATUITS DE SURCROÎT) :

Travail informatique sous Windows

avec Notepad++, Ruby et AsciiDoctor

Article mis en ligne le 27 avril 2016

par Laurent Bloch

Cet article a été inspiré par celui de Benoît Benedetti, Administrateur Système Linux à l'Université de Nice Sophia Antipolis, « AsciiDoc et AsciiDoctor pour soigner votre documentation », dans « Gnu/Linux Magazine France » n° 190 de février 2016.

Le système Windows n'a pas été conçu pour faire de l'informatique, mais pour utiliser de l'informatique faite par d'autres, ce qui a aussi pou à voir avec l'informatique que le permis de conduire a à voir avec la thermodynamique. Windows est fait pour ceux qui ne veulent *surtout pas* savoir comment ça marche et de quoi il retourne.

Si l'on veut faire de l'informatique, il vaut donc mieux utiliser un système fait pour cela, dont l'archétype est Linux/Unix. Il y a trois circonstances qui justifient de faire de l'informatique sous Windows :

- être un salarié de Microsoft ou d'un de ses sous-traitants ;
- développer du logiciel ou du matériel destiné à fonctionner sous Windows ;
- administrer un parc de machines Windows, en assurer la sécurité, un audit...

Créer un texte sans mise en page inutile

Cela dit, il y a toute une collection de situations intermédiaires, qui, sans être du travail proprement informatique, demandent des outils un tant soit peu professionnels. Ainsi, pour écrire le texte d'un article pour publication dans un CMS (Spip, Drupal, Wordpress...) ou dans un Wiki, Word (ou LibreOffice, ou OpenOffice) ne sont pas de bonnes solutions, parce que ces logiciels font beaucoup d'autres choses en sus d'enregistrer le texte, lequel ne représentera souvent que 5% de la taille finale du fichier créé, et ce au prix de tas de complications épuisantes, pour un résultat décevant parce que le texte apparaîtra avec toute une mise en page qui gênera plutôt qu'autre chose.

Régulièrement je suis amené à travailler avec des utilisateurs de Windows, pour rédiger une page Web, un énoncé d'exercices de TP, voire un article ou un livre, cela se finit toujours par un échange de documents Word, c'est improductif et horripilant. En outre, souvent, le logiciel qui a servi à cet usage est volé, ce qui est mal.

Ainsi, créer un texte : un jour un lecteur de mon manuel de programmation m'a reproché de ne pas avoir expliqué comment on créait le texte d'un programme. Rétrospectivement, je suis encore offusqué de cette remarque, mais je devrais plutôt la considérer comme le symptôme des dégâts intellectuels provoqués par la généralisation des interfaces graphiques, les cliquodromes, qui sont bien destinés à ne pas faire d'informatique mais à utiliser celle que d'autres ont faite. Mon père et mon grand-père, littéraires à l'état pur, se donnaient la peine de comprendre le fonctionnement des machines à vapeur, des centrales hydro-électriques et des fours Martin, ils avaient du mal mais ils en faisaient l'effort, il me semble que les non-informaticiens d'aujourd'hui pourraient entreprendre une démarche analogue, d'autant mieux que c'est plus facile à comprendre.

Donc, créer un texte : contrairement à l'opinion de plus en plus répandue, cela ne se fait pas avec Word ou un de ses succédanées, LibreOffice ou OpenOffice, qui sont des logiciel de *traitement* de texte, mais avec un *éditeur* de texte. Dans le monde informatique les éditeurs les plus convenables se nomment Emacs, vi ou ed, qui soit dit en passant existent sous Windows, mais qui sont sans doute d'un emploi rebutant pour la victime des cliquodromes. Aussi existe-t-il des outils d'un usage plus facile.

Notepad++

Tout système Windows vient avec un logiciel nommé Notepad, destiné à cet usage, mais en fait peu pratique, et WordPad n'est pas meilleur. L'utilisateur de Windows qui n'aura pas encore sauté le pas et installé Emacs trouvera son bonheur avec l'excellent Notepad++, qui lui permettra de créer simplement toutes sortes de documents, y compris des programmes correctement indentés, ce qui est la première condition de leur justesse (il y a même un style Scheme qui met bien les mots-clé en couleur, les parenthèses alignées et tout et tout). C'est vraiment un bon outil, installé en deux minutes, et doté de toutes les fonctions nécessaires tout en restant facile d'usage.

De grâce, pour tout ce qui n'est pas destiné à une mise en page avec polices et styles, laissez Word et ses congénères, vive Notepad++ !

Ruby

Une fois que l'on sait créer des textes simplement, sans toutes les complications de Word/LibreOffice/OpenOffice, on aimerait bien pouvoir en faire une page HTML, ou un document PDF, ou même un document LaTeX. C'est possible avec AsciiDoc, ou aussi AsciiDoctor. Ces deux logiciels utilisent le même format très simple pour décrirela mise en page, AsciiDoc est peut-être plus rapide à installer sous Windows et peut produire du HTML (4 ou 5), du DocBook, du LaTeX, des planches de présentation et du WordPress. AsciiDoctor produit les mêmes formats, plus du PDF : un moyen simple de produire du PDF, cela ne se refuse pas, alors ce sera AsciiDoctor, mais avant d'installer AsciiDoctor il faut installer Ruby, ce qui fait l'objet de la présente section.

Le moyen le plus simple d'installer Ruby sous Windows me semble RubyInstaller. Alors allons-y : une fois l'installateur téléchargé c'est l'affaire de quelques secondes. Ruby se lance dans la fenêtre de terminal moche de Windows, c'est Windows, pour vous dégoûter de l'informatique, mais là c'est pour la bonne cause alors on continue.

Si vous utilisez un autre système que Windows, vous pouvez aussi installer Ruby (les paquets de votre distribution habituelle sont probablement trop anciens pour supporter AsciiDoctor).

```
wget https://cache.ruby-lang.org/pub/ruby/2.3/ruby-2.3.1.tar.gz
tar xvzf ruby-2.3.1.tar.gz
cd ruby-2.3.1/
./configure
make
sudo make install
```

Installation de AsciiDoctor

C'est très simple (expliqué ici aussi), dans la fenêtre de texte qui s'ouvre lorsque l'on lance Ruby, taper :

```
gem install asciidoctor
```

Puis on peut installer les outils qui produisent les différents formats de sortie :

```
gem install --pre asciidoctor-pdf
```

```
gem install asciidoctor-latex
```

Mise en page AsciiDoc

Ce qui est séduisant avec le format AsciiDoc, c'est qu'il est simple et pourtant riche de possibilités.

C'est un langage de balises, comme LaTeX ou Html, et à l'inverse de Word/LibreOffice/OpenOffice, c'est-à-dire que les indications de mise en forme figurent dans la texte lui-même, sous une forme syntaxique qui les distingue du texte destiné à la publication. Voici un exemple :

```
:source-highlighter: coderay
:author:    Laurent Bloch
:email:     lb@laurentbloch.org
:revdate:   27 avril 2016
:lang:      fr

= Pour AsciiDoctor et AsciiDoc (ceci est un titre)

== Plan (ceci est un intertitre)

== Un système simple et puissant pour la documentation (ceci est un intertitre)

* Système technique et révolution industrielle (une liste de thèmes)
* Révolution cyberindustrielle en cours
* Production du système technique contemporain

----
sudo bash
gem install --pre coderay
----

[source, clojure]
----
(define (fact n)
   (if (zero? n)
       1
       (* n (- n 1))))
----

... etc.
```

On distingue facilement la syntaxe, en tête, des attributs (ou métadonnées, `coderay` est le nom d'un style de mise en forme, les noms des autres attributs ne prêtent pas à confusion).

On trouvera sur le site AsciiDoc la description complète de la syntaxe.

Produire un document

Une fois que le système de production est installé et que l'on a saisi son document avec Notepad++, on peut l'éditer au format désiré :

```
asciidoctor -r asciidoctor-pdf -b pdf mon-document.adoc
```

```
asciidoctor -r asciidoctor-latex -b latex mon-document.adoc
```

La sortie par défaut est en Html. On peut aussi avoir du code colorié en fonction de la syntaxe :

```
[source, clojure]
----
(define (fact n)
   (if (zero? n)
       1
       (* n (- n 1))))
----
```

Chez moi, les fichiers de style selon les langages sont ici :

```
/opt/rubies/ruby-2.3.0/lib/ruby/gems/2.3.0/gems/coderay-1.1.1/lib/coderay/scanners
```

Le résultat n'est pas mal, en Pdf comme en Html, je ne saurais trop vous conseiller ces logiciels d'usage aisé pour de bons résultats.

Voici les résultats :

Je devrais pouvoir assez facilement produire un style Scheme à partir de celui de Clojure. Voici le corps de la sortie Html :

[code]

```
<body class="article">
<div id="header">
<h1>Pour AsciiDoctor et AsciiDoc (ceci est un titre)</h1>
<div class="details">
<span id="author" class="author">Laurent Bloch</span><br>
<span id="email" class="email"><a href="mailto:lb@laurentbloch.org">lb@laurentbloch.org</a>
</span><br>
<span id="revdate">26 avril 2016</span>
</div>
</div>
<div id="content">
<div class="sect1">
<h2 id="_plan_ceci_est_un_intertitre">Plan (ceci est un intertitre)</h2>
<div class="sectionbody">

</div>
</div>
<div class="sect1">
<h2 id="_un_système_simple_et_puissant_pour_la_documentation_ceci_est_un_intertitre">Un système
simple et puissant pour la documentation (ceci est un intertitre)</h2>
<div class="sectionbody">
<div class="ulist">
<ul>
<li>
<p>Système technique et révolution industrielle (une liste de thèmes)</p>
</li>
<li>
<p>Révolution cyberindustrielle en cours</p>
</li>
<li>
<p>Production du système technique contemporain</p>
</li>
</ul>
</div>
<div class="listingblock">
<div class="content">
<pre>sudo bash
gem install --pre coderay</pre>
</div>
</div>
<div class="listingblock">
<div class="content">
<pre class="CodeRay highlight"><code data-lang="clojure">(define (fact n)
  (<span class="keyword">if</span> (<span class="keyword">zero?</span> n)
      <span class="integer">1</span>
      (<span class="keyword">*</span> n (<span class="keyword">-</span> n <span
class="integer">1</span>))))</code></pre>
</div>
</div>
</div>
</div>
</div>
<div id="footer">
<div id="footer-text">
Last updated 2016-04-27 09:17:26 CEST
</div>
</div>
</body>
```

P.S. :

Cet article a été inspiré par celui de Benoît Benedetti, Administrateur Système Linux à l'Université de Nice Sophia Antipolis, « AsciiDoc et AsciiDoctor pour soigner votre documentation », dans « Gnu/Linux Magazine France » n° 190 de février 2016.

Répondre à cet article