Introduction à l'environnement UNIX

R&I 1

Antoine Rollet

Bordeaux INP - Enseirb-Matmeca

LaTeX

Introduction

Documentation

Lien vers le cours :

https://rollet.zzz.bordeaux-inp.fr/

Un peu de lecture :

`https://rollet.zzz.bordeaux-inp.fr/IntroUnix/documentation/refcard-latex.pdf`

`https://rollet.zzz.bordeaux-inp.fr/IntroUnix/documentation/Apprends_LaTeX.pdf`

Historique

- TEX (Donald E. Knuth) : programme pour des documents (scientifiques)
- TEX grande stabilité, portabilité, absence quasi totale de bogues, respect des usages des typographes et composeurs de textes.
- TEX se prononce "tech" (comme dans tech-nologie), X → χ (chi).
- LATEX (Leslie Lamport) sur-couche basée sur TEX comme système de mise en page avec de nombreux modules

Introduction

LaTeX vs. WYSIWYG

- LaTeX : outil très puissant pour rédiger tous types de documents, notamment scientifiques
- L'auteur → structure logique du document (section, sous-section, etc)
  LaTeX → calcul la mise en page optimale avec les bonnes règles de typographie.
- Information de structure insérée dans le texte avec des **commandes** : distinction entre le fond du document et sa forme.
- Démarche opposée aux traitements de texte WYSIWYG (What You See Is What You Get) : mélange de structure du document et de mise en forme. L'auteur voit à l'écran le résultat final du document.
- LaTeX → pas de résultat final immédiat, nécessite une phase de compilation. **compilateur : programme qui permet de transformer un langage vers un autre langage**. 'code source latex' → LaTeX→ (dvi, ps, pdf, html, etc)

☞ Remarque : ces slides sont rédigés en LaTeX !
☞ Extrait des retours d'année R&I1 précédents : *"LaTeX est vraiment utile..."*

Caractéristiques de LaTeX

Arguments en faveur de LaTeX par rapport aux logiciels WISIWYG

- mise en page professionnelle / respect des règles typographiques
- facilité de la composition des formules mathématiques
- on ne se préoccupe pas de la mise en page, on peut donc se concentrer sur le fond
- gestion automatique des structures complexes telles que : tables des matières, notes de bas de pages, liste des figures, références croisés, bibliographies, index, et bien d'autres.
- possibilité de programmer/très extensible
- multi-plateforme / libre
- extensions conventionnelles : `.tex` (LaTeX), `.bib` (fichier de bibliographie), `.aux` (fichiers auxiliaires), `.sty` (fichiers de style), etc

Remarque

Création de bibliographie : voir le système `bibtex` (fichier `.bib`)

☞ Remarques :

- les autres éditeurs de documents connus (e.g. libreoffice) ont aussi des qualités. A l'avenir, ça sera à vous de faire votre choix en fonction du contexte.
- il est possible d'utiliser d'autres éditeurs de fichiers qu'`emacs` pour rédiger des documents en LaTeX.

Un mot sur Markdown

Une autre alternative à LaTeX, très à la mode

`markdown` = langage de balise, syntaxe simple, facile à lire et à écrire

Un document `markdown` peut être converti en divers formats (e.g. sous linux, on peut utiliser la commande `pandoc`), notamment pdf ou html.
→ `pandoc file.md -o file.pdf` ou `pandoc file.md -o file.html`

Utile pour du protoypage rapide ou des mises en page simples, beaucoup moins de possibilités que LaTeX

Plus d'informations

`https://www.markdownguide.org/getting-started/`
Et un peu de pratique :
`https://www.markdowntutorial.com`

Structure d'un fichier source

LaTeX Langage balisé à base de commande

- **Commande**
  - commence par `\` suivi du nom de la commande et est sensible à la casse
  - syntaxe : `\commandname[options]{parameters}`
  - parfois : `\commandname*[options]{parameters}` action particulière (ex. `\section{}, \section*{}`)
- **Squelette d'un fichier**

```
\documentclass[options]{parameters}
\usepackage[package-options]{package-name} %optionnel
% >>préambule<<
\begin{document}
  le document commence ici
\end{document}
```

- `\documentclass[options]{paramaters}`
  `parameters` : article, report, book, etc ; `options` : 10pt, twocolumn, etc
- `\usepackage[package-options]{package-name}`
  `package-name` : graphicx, amsmath, etc ; `package-options` (personnalisation)
- ligne commençant par `%` = commentaires
  ☞ *pour le utilisateurs d'*emacs*, essayer la commande* `comment-region`
- **Gros documents, plusieurs fichiers**
  `\input{file.tex}` ou `\include{file}` (idem avec saut de page, ⚠sans le `.tex`)

Compilation / visualisation

LaTeX : Langage compilé qui nécessite un compilateur (programme informatique)
Code source LaTeX→ compilateur (`pdflatex`) → fichier visualisable, imprimable

- **Compilation standard avec le programme `pdflatex`**
  - comprend les commandes et instructions du fichier source.
  - phase de compilation crée un ensemble de fichiers : `.pdf` (mise en forme), `.aux` (numéro de pages, figures, références, etc), `.log` (messages d'informations ou d'erreur), `.toc` (table des matières), ...
  - il est souvent nécessaire de compiler le fichier `tex` deux fois pour que les références s'affichent correctement
  - visualisation : `evince` (a priori tout lecteur de pdf peut convenir)
    ☞ `evince` se met automatiquement à jour en cas de changement dans le fichier pdf

  Généralement, on édite le code LaTeX dans son éditeur de texte favori, on le compile dans un terminal (ou avec l'éditeur s'il le permet), tout en maintenant une fenêtre ouverte qui visualise le fichier pdf.

Compilation / visualisation

- **Prise en charge des figures/images**
  - `pdflatex` : `pdf`, `png`, `jpeg`
  - Privilégier les figures/images vectorielles par rapport aux images bitmap
- Edition partagée sur `http://www.overleaf.com`
- **Outil de manipulation des pdf :** `pdftk` / `pdfjam`

  ex : concaténer des pdf
  - avec pdftk : `pdftk a.pdf b.pdf cat output c.pdf`
  - avec pdfjam : `pdfjam a.pdf b.pdf -o c.pdf`

    (`pdfjam` est accessible sur les machines de l'école ici : `/opt/pdfjam/bin/pdfjam`)

Exercices

Récupérer le fichier
`https://rollet.zzz.bordeaux-inp.fr/IntroUnix/ressources/first-doc.tex`
(avec `wget`) et le mettre dans une arborescence locale du type
`~/IntroUnix/latex/compil/first-doc.tex`

1. Compiler le fichier, il y a des erreurs de compilation, interpréter, corriger les erreurs jusqu'à compilation **totale**
2. Vérifier dans le répertoire courant la présence des fichiers auxiliaires. De quels types sont-ils ?
3. Visualiser le document produit. Modifier le fichier source et vérifier la modification dans le document final.
4. Produire un fichier `pdf`. Le visualiser avec `evince`.
5. Effacer tous les fichiers sauf le fichier source. Modifier le fichier source. Compiler avec `pdflatex`. Vérifier la génération des fichiers auxiliaires et du `pdf`. Visualiser le `pdf`. Modifier le fichier source. Recompiler. Visualiser la modification.
6. Créer deux petits fichiers `.tex` contenant chacun une ligne de texte. Utiliser la commande `\input` pour intégrer le 1er, et la commande `\include` pour intégrer le 2è.

first-doc.tex

```
\documentclass{article}
% My first document using a professionnal editor!

\begin{document
This is my first \LaTeX{}} document with a spaces
test and.



A new    \ \ paragraph.

Today is \today.



end{document}
```

Quelques commandes

- Sauts de ligne, de page et césure : `\\`, `\newline`, `\newpage`, `ce\-su\-re`
- Chaînes toutes prêtes : `\today`, `\date`, `\TeX`, `\LaTeX`
- Caractères spéciaux :
  `\c{c}`, `\oe`, `\^o`, `\"i`, `\dots`, `\slash`, `\textbackslash`...
- Espaces : LaTeX calcul l'espacement optimal que les mots doivent avoir.
  - espace insécable, espacement : `~`,`\␣`, `\hspace{m}`, `\vspace{m}`,
  - mesure automatique : `\textwidth`, `\textheight`
- Polices : `\emph{}`, `\textbf{}`, `\texttt{}`, `\textit{}`, `\textsc{}`
- Tailles :
  `\tiny`, `\scriptsize`, `\footnotesize`, `\small`, `\normalsize` (default),
  `\large`, `\Large`, `\LARGE`, `\huge`, `\Huge`
  → Utilisation de blocs, par ex : `{\small petit} normal`

Exercice : Ajouter le texte dans votre fichier qui se rapproche le plus possible du rendu suivant :
The small text is written in big and conversely ;
italic is written in *bold* and conversely.

Quelques commandes

- Structure logique d'un document (toutes ont une version avec `*`)
  - `\part{title}` (`book`)
  - `\chapter{title}` (`book` et `report`)
  - `\section{title}`
  - `\subsection{title}`
  - `\subsubsection{title}`
- Un paragraphe : unité typographique commençant par un alinéa (**obtenu par une ligne vide dans le code source**) = pensée cohérente ou développement d'une idée. Lorsque la réflexion se poursuit, un saut de ligne `\\` suffit (pas d'alinéa).
- Références croisées : `\label{nom}`, `\ref{nom}`, `\pageref{nom}`, `\eqref{nom}`
  ⚠ Ne jamais écrire une référence en toutes lettres !
  Ecrire : `Section~\ref{labelCorrespondant}` et non `Section 1` (pourquoi ?)
- Listes automatiques : `\tableofcontents`, `\listoffigures`, `\listoftables`

→ Créer deux sections dans votre fichier précédent, et faites référence à la première dans la deuxième.

Environnements

Permet de composer du texte dans un contexte particulier
syntaxe générale `\begin{env} contenu \end{env}`

- `center` pour centrer le texte
- Listes : `enumerate, itemize`, utilisation avec `\item`
- Impression verbatim : `verbatim` (on peut aussi utiliser `\verb|text|` où `|` peut être remplacé par n'importe quel caractère)
- Tableau : `\begin{tabular}{description}`
  - `description` : droite, gauche, centré, trait vertical : `r, l, c, |`
  - séparateur de colonne, saut de ligne : `&`, `\\`
  - trait horizontal, trait horizontal de la colonne `i` à `j` : `\hline`, `\cline{i-j}`
  - fusion de colonnes : `\multicolumn{n}{description}{texte}`
    → `n` : nombre de colonnes fusionnées, `description` : idem ci-dessus

Dans votre fichier précédent, ajouter une énumération de trois items, puis donner une grille centrée du jeu de *morpion* de votre choix, par exemple celle-ci dessous à gauche, et le modifier ensuite pour obtenir la version de droite :

X	O	X
O	X	O
O	X	O

	OXO	
X	O	X
O	X	O
O	X	O

Objets flottants

Objets ne pouvant être coupés par un saut de page (figure, tableau, algorithme, etc). LaTeX place automatiquement les objets flottants, là où il y a de la place dans la page.

- **Squelette standard**

```
\begin{...} %figure ou table
  % commandes pour une figure, le tableau ou autre chose
  \caption{legende}
  \label{labelName} % \label se trouve apres \caption
\end{...} % figure ou table
```

- **Important**, un flottant (une figure, un tableau, etc), doit toujours
  - comporter une légende explicative ;
  - être cité, référencé dans le texte, par ex.
    `The figure~\ref{fig:nomdulabel} shows`... ou encore
    `The results are summarized by Table~\ref{tab:nomdulabel}`....
- Par ex : figure, tableau : `\begin{figure}[pref-placement]`, `\begin{table}[pref-placement]`
- Ordre de préférence de placement : `h t b p` (here, top, bottom, page)

Inclusion de figures

- dans le préambule : `\usepackage{graphicx}`, attention au `x`, `graphics` existe aussi
- commande : `\includegraphics[options]{filename}`
  - `options` : permet de régler la taille de la figure
  - préférer les tailles relatives, préservation de l'aspect/ratio
  - jouer sur la largeur par rapport à la largeur du texte : `\includegraphics[width=0.5\textwidth]{filename}`
  - `filename` (avec ou sans l'extension)

Exercice :

→ Dans votre fichier précédent, intégrez le logo de l'Enseirb-Matmeca(disponible à `https://rollet.zzz.bordeaux-inp.fr/IntroUnix/ressources/logo_em.pdf`)

→ L'intégrer dans un environnement `figure`, et y faire référence

Et le français dans tout ça ?

Prise en charge des caractères français, il y a plusieurs façons de faire :

- utiliser les caractères spéciaux `\c{c}`, `\oe`, `\^o`, `\"i`, ...
  (mais un peu pénible à la longue...)
- utiliser `\usepackage[utf8]{inputenc}` en encodant le fichier en `utf8`.
  → indique à LaTeX comment les caractères du fichier .tex sont codés
  → considéré par défaut si non précisé normalement

☞ Pour modifier l'encodage d'un fichier sous `emacs` :

- commande `set-buffer-file-coding-system`

Et le français dans tout ça ?

Prise en charge des caractères français, il y a plusieurs façons de faire :

- utiliser les caractères spéciaux `\c{c}, \oe, \^o, \"i, ...`
  (mais un peu pénible à la longue...)
- utiliser `\usepackage[utf8]{inputenc}` en encodant le fichier en `utf8`.
  → indique à LaTeX comment les caractères du fichier .tex sont codés
  → considéré par défaut si non précisé normalement

☞ Pour modifier l'encodage d'un fichier sous `emacs` :

- commande `set-buffer-file-coding-system`

Le package `fontenc` définit l'encodage de la police :

- `\usepackage[T1]{fontenc}` → T1 est le plus adapté pour les caractères français

Et le français dans tout ça ?

Prise en charge des caractères français, il y a plusieurs façons de faire :

- utiliser les caractères spéciaux `\c{c}, \oe, \^o, \"i, ...`
  (mais un peu pénible à la longue...)
- utiliser `\usepackage[utf8]{inputenc}` en encodant le fichier en `utf8`.
  → indique à LaTeX comment les caractères du fichier .tex sont codés
  → considéré par défaut si non précisé normalement

☞ Pour modifier l'encodage d'un fichier sous `emacs` :

- commande `set-buffer-file-coding-system`

Le package `fontenc` définit l'encodage de la police :

- `\usepackage[T1]{fontenc}` → T1 est le plus adapté pour les caractères français

Le package `babel` apporte un support multilingue (titres (e.g. chapter → chapitre)) et adapte les règles typographiques (césures, etc...) :

- `\usepackage[french]{babel}`

Exercice : reprendre votre fichier précédent, le traduire, et le mettre en forme en langue française. Remarquer que la mise en forme de la date a changé.





Documentation

Un peu de lecture :

`https://rollet.zzz.bordeaux-inp.fr/IntroUnix/documentation/refcard-amslatex.pdf`

AMS-LaTeX

- AMS-LaTeX : collection d'extensions et de classes pour saisir des formules mathématiques
- développé par l' *American Mathematical Society*.
- `\usepackage{amsmath, amssymb}`
- symboles mathématiques dans le texte `$...$`, par ex. $a+b=c$: `$a+b=c$`
- hors texte : environnement (flottant) `\begin{equation}...\end{equation}`, par ex : `\begin{equation} a+b=c\end{equation}`

$$a+b=c \tag{1}$$

- pour supprimer la numérotation : `\begin{equation*}...\end{equation*}`

$$a+b=c$$

  Equivalent à `\[ ... \]` ou `$$ ... $$`

- référencer une équation
  1. déposer une étiquette (label) `\label{eq:addition}`
  2. la référencer dans le texte `\eqref{eq:addition}`

  En résumé :

  1. `\begin{equation} \label{eq:addition}a+b=c\end{equation}`
  2. `L'équation \eqref{eq:addition} est une addition`
     → *l'équation* (1) *est une addition*

Commandes en vrac

- `^ _ ^{} _{}` : exposant, indice
- `\sum, \int, \lim, \prod` : somme, intégrale, limite, produit
- `\frac{}{}` : fraction
- les symboles grecs, syntaxe du nom, e.g., `\pi, \epsilon, \Gamma` , ...
- opérateurs, etc. : `=, \neq, \forall, \in, \notin,`...
- `\, \; \quad \qquad` gestion des espaces
- `\sqrt{}, \sqrt[]{}`, racine carré, racine n^{ieme}.
- `\cdots, \ldots, \vdots, \ddots`, différents points de suspensions
- `\text{...}` texte (droit) en mode math
- rq : variables en italique, les fonctions en police droite, e.g., `\sin, \cos, etc`.
- `\partial, \| \| , | |` : dérivée partielle, norme, valeur absolue
- etc.

Lien utile : `http://detexify.kirelabs.org`

Exercice : → écrire sous forme d'équation : $E = MC^2$ puis $a + \beta = 1 + \frac{\beta}{a}$ si $a \neq 0$, et enfin

$$\int_a^b f(x)dx = -\int_b^a f(x)dx$$

LaTeX

Afficher du code

Afficher du code : le package `listings`

Permet d'afficher du code de façon formatée

- `\usepackage{listings}` à ajouter en début de document
- Utilisation : `\begin{lstlisting} some code \end{lstlisting}`
- Nombreuses options possibles, notamment le langage affiché (C, Java, etc...) :
  `\begin{lstlisting}[language=Java]...`
- Possible de configurer une fois pour toutes :
  `\lstset{language=C#, ...}`
- Utilisable intégré dans du texte : `\lstinline|some small code|`
  (les "|" peuvent être remplacés par n'importe quel caractère)
- Intégrer un (morceau de) fichier de code :
  `\lstinputlisting[language=C, firstline=32, lastline=47]{source.c}`

Plus de détails :

`https://ctan.mines-albi.fr/macros/latex/contrib/listings/listings.pdf`

Exercice : afficher le code suivant à la suite de votre ficher LaTeX :

```
int factorielle (int x) {
  if (x == 1)
    return x;
  else
    return x*factorielle(x-1);
}
```

Exercices

1. Avant de commencer penser
   - à activer dans `emacs` le mode `flyspell-mode`
   - à activer dans `emacs` le mode `auto-fill-mode`
   - à utiliser plusieurs bureaux
   - à mettre votre souris de coté
2. Récupérer le fichier `https://rollet.zzz.bordeaux-inp.fr/IntroUnix/ressources/rapport.tex` (avec `wget`), et le compléter pour essayer de rédiger un document le plus proche possible de la mise en page suivante : `https://rollet.zzz.bordeaux-inp.fr/IntroUnix/ressources/rapport_enseirb_matmeca.pdf`. Terminer par la page de garde. Le logo est diponible ici : `https://rollet.zzz.bordeaux-inp.fr/IntroUnix/ressources/logo_em.pdf`.
   ☞ Ce rapport pourra vous servir comme base pour vos rapports futurs
3. Mettre sous forme d'article "papier" le contenu en ligne suivant : `https://interstices.info/la-machine-a-perspective-de-jan-van-eyck/` [1].
4. Rédigez votre CV en LaTeX

1. sous licence Creative Common