Pourquoi je déteste MS Word © ?

Ou, la puissance combinée de LaTeX, Git et Vim

dernière version disponible :

https://scarpet42.gitlab.io/lagivim/lagivim.pdf

Stéphane Carpentier

lagivim@fiat-linux.fr

31 décembre 2020

Avant propos

Avant de commencer à critiquer Word et à encenser LaTeX, Vim et Git, j'ai besoin de faire quelques précisions. D'abord, quand je dis que Word est une bouse infâme, je ne méprise aucunement ses utilisateurs. J'utilise Word au boulot. Sous la contrainte soit, mais je l'utilise quand même. Je ne deviens pas plus ou moins méprisable en changeant d'outil. Ensuite, je ne cherche absolument pas à inciter qui que ce soit à abandonner Word pour passer à LaTeX. Chacun fait ce qu'il veut et par ce document je revendique mon droit de faire ce que je veux. J'entends trop souvent des esprits limités critiquer mes choix pour tomber dans le même travers. Pourquoi je parle d'esprits limités ? Parce que le raisonnement est le suivant. La personne connaît Word et pas LaTeX. La personne a donc plus de facilités à gérer des documents avec Word. Donc, si c'est plus facile pour la personne, c'est donc plus facile pour moi. Et donc, si je ne choisis pas la solution la plus simple pour la personne, c'est que je suis con, c'est que j'aime me compliquer la vie, c'est par esprit de contradiction ou pour toute autre mauvaise raison. Ce n'est pas parce que quelqu'un est plus à l'aise avec une solution que tout le monde doit être plus à l'aise. Refuser cette évidence est faire preuve d'un esprit limité. Je veux montrer que mon choix est le bon pour moi, même si tout le monde ne peut pas l'adopter.

Ce n'est ni un tutoriel, ni une initiation, ni une documentation. Je vais considérer que le lecteur ne connait pas les outils dont je vais parler. Je vais donc les présenter, puis montrer ce que leur puissance peut apporter une fois le stade d'apprentissage passé. Il est très facile de trouver des documents pour ces logiciels, mais je n'ai pas trouvé de comparatif sérieux. Les seuls comparatifs que j'ai vu montraient la supériorité typographique de LaTeX sur Word. Ensuite, ils donnaient l'impression que la difficulté d'utilisation de LaTeX était compensée par le rendu des documents. C'est faux. Le rendu des documents est effectivement meilleur avec LaTeX. Mais une fois la difficulté d'apprentissage passée, bien

utilisé, LaTeX est bien plus pratique à utiliser que Word. C'est ce que je veux montrer ici, même si tout le monde est persuadé de l'inverse. Avec tout ce que j'ai pu me prendre dans les dents sur mon choix, il ne pourra pas m'être reproché de ne plus pouvoir mâcher mes mots. Aujourd'hui, c'est à mon tour de me défouler. Si vous voulez que je mette des gants pour me défouler, pas de problème : je vais mettre des gants de boxe.

Comparaison entre Word et LaTeX

Un peu de typographie

Je ne vais pas m'y attarder parce que la littérature montrant la supériorité de LaTeX sur Word est abondante. De plus, personne ne s'intéresse à la typographie, sinon, Word ne serait pas aussi mal utilisé. Cependant, le but ultime d'un document est d'être lu. Et c'est une bonne typographie qui permet à un document d'être lu dans les meilleurs conditions possibles. Je ne peux donc pas passer ce point sous silence.

D'abord, la bonne typographie ne se remarque pas. C'est ce que les gens ont souvent du mal à comprendre. C'est pour ça qu'il y a tant de documents avec des parties soulignées ou en gras à tort et à travers. Le pire est la partie à la fois soulignée et en gras. Ce que l'œil remarque, ce sont les erreurs de typographie. Quand il n'y a pas d'erreur de typographie, l'œil lit de façon optimale et ne remarque rien. Même si, après avoir lu des milliers de pages wordiques rédigées n'importe comment, vous avez l'impression inverse.

Tenez ce document à bout de bras et regardez le de loin. Vous devez voir des blocs homogènes, avec les titres qui ressortent et deux parties. L'une des deux parties que vous devez voir est la partie soulignée. Le soulignement avec LaTeX, contrairement au soulignement wordique, ne coupe pas les jambages. Le soulignement wordique est tellement nul et répandu qu'une légende se répand de plus en plus. La légende veut que le soulignement soit typographiquement mauvais. C'est faux, c'est le mauvais soulignement, ou le soulignement dans de mauvaises conditions qui est mauvais. L'autre partie que vous devez voir est la partie en gras. **La mise en gras avec LaTeX doit être sensiblement égale à celle avec Word.** Par contre, vous ne devez pas remarquer la partie en italique. *La partie en italique ne se remarque que quand vous la lisez.* De même, vous ne voyez pas « la partie entre guillemets » avant de la lire.

Le paragraphe précédant était typographiquement mauvais pour montrer mon point. L'œil ne lit pas les lettres une par une. Toutes les études sont unanimes là-dessus. L'œil lit des parties autour de la lettre à lire. Si, au dessus ou en dessous de la partie à lire, il y a des parties qui ressortent, son attention est détournée et il a plus de mal à lire. Si vous voyez une partie en gras ou soulignée lorsque vous tenez un texte à bout de bras, lorsque vous lirez la phrase au dessus ou en dessous, l'œil la verra. Si une partie est soulignée sur un texte de mille pages, vous ne fatiguerez pas vraiment plus. Mais si presque la moitié des mots est en gras ou soulignée alors, vous fatiguerez plus sans savoir pourquoi. Le gras et le souligné ne sont pas faits pour être lus au fil de l'eau mais pour être trouvés

facilement. Par exemple, dans un dictionnaire, vous devez voir les mots directement pour les trouver facilement. De même que les titres doivent ressortir pour pouvoir trouver rapidement la partie intéressante. C'est pour ne pas perturber l'œil que les titres sont séparés du reste du texte. Ils doivent ressortir pour être remarqués rapidement, mais ne doivent pas gêner la lecture du paragraphe. Puisque vous ne me croyez pas, lisez cette partie :

Je suis en retard. Je vois mon bus qui arrive. Je vais avoir du mal à l'avoir. Il faut que je coure. Je cours. Le plus vite possible. Il s'arrête. Je suis encore loin. Je n'arrive plus à respirer. Mais je dois accélérer. Il m'a vu. Il m'attend. Je l'ai eu de justesse, c'est bon, maintenant je peux me reposer, il faut que je reprenne ma respiration, mais je n'y arrive pas, je suis essoufflé.

Maintenant, lisez ça :

Je suis en retard, je vois mon bus qui arrive, je vais avoir du mal à l'avoir, il faut que je coure, je cours, le plus vite possible, il s'arrête, je suis encore loin, je n'arrive plus à respirer, mais je dois accélérer, il m'a vu, il m'attend, je l'ai eu. De justesse. C'est bon. Maintenant, je peux me reposer. Il faut que je reprenne ma respiration. Mais je n'y arrive pas. Je suis essoufflé.

Vous avez vu la différence ? Dans le premier cas vous lisez lentement lorsque l'action est rapide et rapidement lorsqu'elle est lente. Dans le second cas, votre lecture est plus en adéquation avec l'action. La seule différence entre les deux textes est la ponctuation. Vous ne le remarquez pas, mais votre œil est attentif à beaucoup de choses. Vous avez lu tellement de documents wordiques que vous ne remarquez même plus les aberrations. Vous êtes même capable de trouver des qualités à des erreurs. Par exemple, si vous trouvez que mes marges sont trop larges, ça vient d'une perturbation wordique qui incite à faire les lignes les plus longues possibles. Lorsque vous lisez, si la ligne est trop longue, votre œil fatigue. Ensuite, à la fin de la ligne, il a du mal à retrouver le début de la ligne suivante.

Lorsque les espaces entre les mots sont trop variés, votre œil est gêné. Et ça, c'est le premier et plus gros reproche fait à Word. Il fait n'importe quoi au niveau de l'espacement : ce n'est pas son problème. Si le dernier mot ne peut par rentrer sur la ligne, il le met à la ligne suivante. Ensuite, il espace les mots de la ligne en conséquence. Mais il ne se demande pas si de descendre le mot de la ligne du dessus permettrait un rendu plus homogène. Alors que LaTeX a un algorithme de calcul très compliqué sur lequel il prend l'affichage de la page en compte. Il est même capable de modifier la largeur des caractères (de moins de deux pourcents c'est invisible à l'œil) pour améliorer le rendu. C'est de la micro-typographie que Word ne gère absolument pas. Au niveau micro-typographie, il y a autre chose que j'aime. LaTeX me le permet mais pas Word. Regardez la fin des lignes de ce texte qui finissent par un signe de ponctuation. Le signe est un peu dans la marge. Comme il est plus petit qu'une lettre, la fin des lignes est moins droite, mais l'œil a l'impression inverse.

Il y a autre chose que Word ne gère pas, mais qu'il donne l'impression de permettre de gérer. En fait, c'est tellement compliqué à gérer avec Word que je considère que ça n'existe pas. Ce sont les ligatures. Regardez les « f » dans « souffle fier ». Ils sont liés avec

le « l » et le « i ». Ce n'est pas juste les caractères qui sont rapprochés, mais ce sont de nouveaux caractères. Là, je ne parle que de différences que vous pouvez voir vous-même. Mais il y en a beaucoup d'autres, la littérature sur le sujet est abondante.

Vous allez me dire que la typographie (micro ou pas) n'a aucune importance. Permettez-moi de ne pas être d'accord. Vous avez le droit de ne pas vous y intéresser. De même que certaines personnes ne s'intéressent ni à la grammaire ni à l'orthographe. Pour moi, une typographie correcte est un respect du lecteur. Au même titre qu'une grammaire et qu'une orthographe soignées. Même si le lecteur ne le remarque pas. Pour moi, utiliser un logiciel qui m'oblige à avoir une typographie médiocre revient à utiliser un logiciel qui me rajoute des fautes de grammaire et d'orthographe.

Maintenant, je vais vous parler des logiciels que j'utilise pour avoir ce rendu. Si la littérature est abondante sur ce que j'ai écrit au dessus, elle a un défaut. C'est qu'elle donne l'impression que LaTeX est plus dur à utiliser mais que ça vaut le coup pour la typographie. Je vais donc vous montrer pourquoi ce n'est pas plus dur à utiliser. C'est peut être plus dur à apprendre, mais je n'en suis pas si sûr. Pour être honnête en comptant le temps d'apprentissage de Word, il faudrait ajouter le temps passé à s'y retrouver à chaque nouvelle version. Alors que LaTeX ne s'apprend qu'une fois.

Pour vous aider à mesurer la portée de cette dernière phrase, je me sent obligé de montrer quelques éléments de comparaison. Le TeXbook a été écrit en 1984 et est, encore aujourd'hui, considéré comme la référence absolue pour celui qui veut maîtriser les arcanes intimes du logiciel (donc pas le débutant, ou alors le débutant vraiment motivé). Je possède plusieurs livres sur LaTeX qui ont été écrits entre 1994 et 2000. Tous ces livres pourraient encore être très utiles à quelqu'un souhaitant découvrir LaTeX. Bien sûr, il y a eu des nouveautés apportées à LaTeX qui n'étaient pas décrites lors de leurs rédactions. Mais tout ce qui y est indiqué doit encore fonctionner sur une version de LaTeX à jour. Je n'ai jamais vu un livre sur Word qui conservait le moindre intérêt dès qu'une nouvelle version de Word sortait. Si vous en connaissez un, dites-le moi. Je serais très intéressé pour me le procurer, il pourrait me servir à rester calme au boulot.

Présentation de LaTeX

D'abord, un mot sur sa prononciation. Le « x » final vient du grec, il s'écrit généralement « ch » et se prononce « k » en français, comme dans technique. En français, LaTeX se prononce donc « latèque » et pas « latèxe ». Maintenant que je l'ai écrit, vous le prononcez comme vous le voulez, ça ne me gêne pas.

Word est un environnement de « rédaction » intégré. C'est à dire que même si c'est nul et que plein de trucs ne vous plaisent pas dedans, vous devez tous les prendre. Pour créer un document Word, vous devez ouvrir Word et tout faire dedans. De la rédaction du document à son impression en passant par sa mise en page et sa correction orthographique.

Ce qui est fondamentalement différent de LaTeX. En effet, LaTeX n'intervient qu'à la fin de la rédaction, à la phase qui s'appelle la compilation. C'est à dire que LaTeX prend un fichier avec tout plein de commandes en entrée et le transforme en un beau document

prêt à être lu <A>. Je ne veux pas dire que vous êtes obligé d'avoir écrit les deux mille pages de votre roman avant de le compiler. Ce que je veux dire, c'est que pour obtenir votre document prêt à compiler, vous employez le moyen que vous voulez. Dans la suite, j'appellerai le fichier qui contient tout plein de commandes le fichier source. Le source est le document de travail, le document final est le document que vous distribuez à vos lecteurs. Lorsque vous écrivez votre document pour LaTeX, vous ne voyez donc pas votre document final. Vous avez donc, de fait, une séparation entre le fond et la forme du document. C'est quelque chose qui perturbe les non habitués. Mais je montrerai plus loin en quoi c'est un atout une fois l'effet de surprise passé.

Voici un exemple de source en LaTeX et de son rendu.

Exemple de source et de son rendu

```
Un tout premier exemple de texte
↪ \LaTeX{}.
Pour \emph{mettre un texte en
↪ emphase}.
Ou alors, pour faire tourner des
↪ textes :
↪ \rotatebox{15}{penché}.
```

Un tout premier exemple de texte LaTeX. Pour *mettre un texte en emphase*. Ou alors, pour faire tourner des textes : penché.

Par exemple, vous voulez écrire « XX^e siècle » proprement. Pour respecter les règles de typographie, le siècle est écrit en petites capitales et le « e » en exposant. Avec Word vous écrivez « xxe siècle ». Vous sélectionnez les deux « x ». Vous farfouillez dans le menu pour trouver les petites capitales. Vous cliquez pour mettre les « x » en petites capitales. Vous sélectionnez le « e ». Vous farfouillez encore dans le menu pour trouver les exposants. Vous cliquez pour mettre le « e » en exposant. Vous pouvez passer à la suite.

Avec LaTeX, rien de tout ça.

Premier exemple de siècle

```
Vous écrivez \textsc{xx}\ieme{}
↪ siècle et c'est tout.
```

Vous écrivez XX^e siècle et c'est tout.

Et là, vous trouvez que ce n'est pas génial. Ce n'est effectivement pas génial, mais mes raisons sont différentes des vôtres, nous y reviendrons. Vous êtes en train de vous dire que c'est plus simple avec Word. Mais c'est seulement parce que vous ne connaissez pas LaTeX. Vous retombez dans votre travers de comparer l'utilisation de commandes que vous ne connaissez pas avec l'utilisation de Word que vous connaissez. Pour quelqu'un comme moi qui connaît LaTeX et pas Word, il est plus rapide d'écrire les commandes LaTeX quand elles sont connues que de chercher au hasard dans les menus de Word.

Une grosse différence entre l'utilisation de LaTeX et de Word est la façon d'appréhender les documents. Dans les deux cas, lorsque vous commencez un nouveau document, vous

<A>. Je simplifie un peu, LaTeX est surtout une surcouche de TeX, mais avec les moteurs récents, la différence n'intéresse que les spécialistes.

avez une page blanche. Mais avec Word, vous regardez les menus pour savoir ce qu'il vous propose. Ensuite, vous cliquez en fonction de ce que vous trouvez bien. Avec LaTeX, vous devez commencer par vous demander ce que vous voulez. En fonction de ce que vous voulez, vous allez utiliser les bonnes commandes pour l'obtenir. Il y a des cas très tordus, mais en général, si vous savez ce que vous voulez, vous pouvez l'obtenir facilement avec LaTeX. Alors qu'avec Word, si vous voulez quelque chose qui diffère un peu des menus par défaut, ça peut être très dur à obtenir. LaTeX vous offre une liberté que Word ne permet pas. Cette liberté a un coût, c'est que vous devez réfléchir à ce que vous voulez. Dès le début, vous devez vous demander si vous voulez écrire une lettre, un CV, un article, un livre ou autre chose avec LaTeX. Beaucoup de gens préfèrent se laisser guider sans se poser de question, mais cette liberté est très importante pour moi.

Une fois que les documents sont en cours de rédaction, la différence dans la gestion des erreurs apparaît. Lorsque je m'y prends mal avec Word, j'ai autre chose que ce que je veux. Il essaye de faire ce que je veux, mais comme il n'y arrive pas, il me donne autre chose. Il ne me prévient pas, bien sûr. Je ne comprends pas pourquoi et je suis frustré. Alors qu'avec LaTeX, si je m'y prends mal, j'ai une grosse erreur de compilation. Il va essayer de passer outre mon erreur et de me donner quelque chose quand même. Mais au moins, je sais pourquoi il fait autre chose que ce que je veux et je peux le corriger. Les erreurs de compilation font peur à certains, mais moi, elles me permettent de trouver mes erreurs. Elles me permettent d'obtenir ce que je veux plus facilement.

Maintenant que nous avons vu les principes, nous allons pouvoir en étudier les avantages. Le premier avantage, qui ne saute pas aux yeux de tout le monde, est que la phase de lecture et la phase de rédaction n'ont pas les mêmes contraintes. Pour lire un livre dans les meilleurs conditions, il est préférable de l'avoir sous forme papier (Les liseuses ne sont pas encore très utilisées en France). Alors que pour l'instant, seul l'écran permet de lire le document en cours de rédaction (la liseuse ne le permet pas non plus, mon argument n'est donc pas ébranlé par la liseuse). Or, une feuille de papier n'a pas les mêmes caractéristiques qu'un écran. Il est beaucoup plus reposant pour l'œil de lire en clair sur fond sombre. Mais, si vous voulez imprimer un livre sur fond sombre, ça va vous coûter une fortune en cartouches d'encre. Sans compter l'aspect écologique. Avec Word, vous êtes obligé, de choisir un aspect visuel, il sera forcément utilisé pour la rédaction comme pour la lecture. Alors qu'avec LaTeX, je choisis un aspect visuel très différent pour la rédaction et pour la lecture. Que ce soit le choix des couleurs ou celui des polices de caractères. Alors que Word est néfaste pour mes yeux en m'obligeant à rédiger des documents dans de mauvaises conditions.

Un autre avantage est que je ne cherche pas à voir les mêmes choses pendant les phases de rédaction et de lecture. Par exemple, quand je rédige, j'aime bien voir mes fautes de frappe. Par contre, je n'apprécie pas trop qu'elles soient soulignées quand je fais lire mon texte. Surtout que ce n'est pas forcément une faute, ça peut être une limitation du correcteur, ça fait néanmoins brouillon. De même, lorsque je rédige, je peux avoir envie de mettre des commentaires à mon usage unique. Par exemple, je peux indiquer qu'il faudra que je développe mieux un paragraphe plus tard. Lorsque je fais lire mon texte, ces commentaires n'apparaissent pas, alors qu'avec Word, ils sont aussi envoyés au lecteur.

C'est bien une limitation de Word qui n'apparait pas avec LaTeX. Un autre exemple, c'est que quand je rédige, j'aime bien savoir d'un coup d'œil si mes phrases sont trop longues. Avec LaTeX, c'est simple, j'écris une phrase par ligne, j'aperçois instantanément la longueur de ma phrase. LaTeX se chargera tout seul de faire de beaux paragraphes. Avec Word, je n'ai aucun moyen d'estimer les longueurs de mes phrases. Allez, un dernier exemple pour la route. Pour rédiger un gros document, il est souvent pratique d'utiliser des fichiers différents. Alors que pour la lecture, il est plus agréable d'avoir tout ensemble. J'ai bien entendu parler de fichiers Word sur plusieurs documents, mais je n'en ai jamais vu. Soit c'est une légende, soit c'est tellement compliqué que seule une élite s'en sert. Alors qu'avec LaTeX, l'inclusion de fichiers est indiquée dans tous les manuels d'initiation que j'ai pu voir.

Si l'intérêt de la séparation des outils peut sembler théorique, il a vraiment des avantages concrets. L'utilisation de Word, non seulement l'empêche pour certains cas, mais en plus ne l'incite pas quand c'est possible.

Séparation entre le fond et la forme

C'est un problème qui est souvent mal compris et donc trop souvent sous-estimé. Lorsque vous rédigez un document, vous avez deux choses à traiter. D'un côté, les mots que vous voulez utiliser, avec éventuellement des illustrations, c'est le contenu, ou le fond du document. De l'autre côté, vous avez la façon dont ces mots et illustrations doivent s'afficher, c'est la présentation ou la forme du document. Avec Word, vous ne pouvez pas y échapper, le fond et la forme sont emmêlés, vous ne pouvez pas les séparer. Avec LaTeX, comme votre document est composé de commandes, il est possible de séparer le fond de la forme, c'est même conseillé. Revenons à notre siècle que nous voulions écrire correctement.

Problème du siècle

```
Quand j'écris \textsc{xx}\ieme{}
↪ siècle je ne sépare pas le
↪ fond de la forme.
```

Quand j'écris XX^e siècle je ne sépare pas le fond de la forme.

C'est ça qui me gêne, pas le fait que ces commandes semblent complexes. Pour séparer le fond de la forme, c'est très simple. Il suffit de définir une commande `\siecle{}` dans le préambule du document.

Bonne version du siècle

```
Ensuite, dans le code du texte, il
↪ devient possible d'écrire
↪ \siecle{xx} siècle pour avoir
↪ le même résultat.
```

Ensuite, dans le code du texte, il devient possible d'écrire XX^e siècle pour avoir le même résultat.

Et là, c'est bien, le fond et la forme sont séparés. Les avantages qui découlent de cette séparation sont nombreux.

Le premier, c'est que lorsque je m'occupe de savoir ce que je dois écrire, je n'ai pas besoin de m'occuper de sa représentation. Ça me permet de me concentrer sur mes phrases, je n'ai pas à m'inquiéter de savoir si le rendu me plait ou pas. Quand je rédige, je ne suis pas perturbé par une mise en page qui pourrait ne pas me plaire. Je m'occuperai de ma mise en page en temps voulu. Au moment de la rédaction de ma commande `\siecle{}`, là je m'occupe de l'affichage. Je ne suis pas perturbé par le contenu des paragraphes à me demander si ma phrase est bien tournée. Je suis focalisé sur la rédaction de la commande. Ce qui me permet de voir que le 1er siècle s'écrit un peu différemment des autres siècles. Il est toujours en petites capitales, mais il y a un exposant différent. Aucun problème, j'ai juste à faire un test pour connaitre le numéro du siècle à afficher et afficher le bon exposant en fonction du résultat du test. En gros, il faut moins d'un quart d'heure pour écrire une commande en prenant son temps. Mais une fois, ce qui vous paraît comme une perte de temps, passé, je peux l'utiliser autant de fois que je veux dans tout mon document. Et même dans tous mes documents.

Et c'est là le second avantage de la séparation du fond et de la forme. C'est que ma commande peut être utilisée dans tous mes documents et même dans les documents d'autres personnes. Et là, c'est beau. Par exemple, il se pourrait que je sois incapable d'écrire la moindre commande mais que je connaisse quelqu'un capable d'en écrire. J'écris une commande vide que j'utilise dans mon document et je continue à le rédiger. En même temps, je vais demander à quelqu'un qui sait comment faire. J'envoie un fichier avec les descriptions des commandes, la personne rempli mon fichier, me le renvoie complété, je l'intègre et c'est tout bon. Si j'ai besoin d'aide pour la rédaction d'une commande, un simple copier/coller de la réponse marche. Avec Word, c'est plus délicat. Il faut commencer par connaître la version de Windows, les paramètres régionaux, la version de Word et probablement la couleur du fond d'écran. Et ensuite, avec quarante deux captures d'écran, nous avons réussit à résoudre notre problème. La réponse apportée marche bien… jusqu'à la prochaine mise à jour de Word où tous les menus ont été chamboulés.

Oui, parce qu'un autre intérêt de la séparation du fond et de la forme, c'est pour le suivi. Parce que s'il s'avère qu'une commande ne fonctionne plus à cause d'une mise à jour, ce n'est pas un gros problème avec LaTeX. D'abord, je n'ai jamais vu ce cas se produire, mais c'est envisageable. Il n'y a que dans la définition de la commande qu'il faut faire une modification. De même, si en plein milieu de la rédaction de mon document j'ai une nouvelle idée. Par exemple, de changer la police de caractère utilisée pour l'affichage du numéro du siècle. Je le fais juste dans la définition de ma commande, pas dans tout mon document. Alors qu'avec Word, il m'aurait fallu rechercher tous les siècles pour les modifier un par un.

Je sais, avec Word, il existe des feuilles de style. D'abord, pratiquement personne ne sait les utiliser en dehors des titres. C'est donc que ça n'a pas la simplicité annoncée par ceux qui disent que Word est plus simple que LaTeX. Ensuite, ce que j'en ai vu semble à peu près correct pour les titres, mais il ne faut pas en demander trop non plus. Par exemple, quand j'ai vu le service de communication donner sa nouvelle charte graphique. Les collègues qui devaient modifier leurs anciens documents riaient beaucoup moins

que moi. Si je riais, c'est parce que je n'avais pas de document à modifier. Pour aller plus loin, je parle des siècles parce que c'est un exemple que je trouve intéressant. Mais quelqu'un qui fait un document sur la musique pourrait apprécier d'avoir des commandes `\auteur{}` et `\instrument{}` par exemple pour les mettre en évidence. Le rédacteur de ce document pourrait aimer avoir un index listant les auteurs et un autre index listant les instruments. Il est super simple de demander à LaTeX d'ajouter automatiquement l'auteur ou l'instrument dans le bon index à chaque appel de la commande. La même chose avec les feuilles de style de Word, au mieux, c'est réservé aux super experts. Bref, la séparation du fond et de la forme permet de faire évoluer la forme le plus simplement possible. Word ne me permet pas de faire la même chose aussi simplement.

La gestion de documents

J'ai déjà un peu abordé le sujet, mais maintenant, je vais y aller à fond. Lorsque j'écris un document que je considère comme important, je veux qu'il dure. Avec Word, pendant combien d'années pouvez-vous continuer à ouvrir un document ? Et en rajoutant en plus un espoir de ne pas trop y perdre dans la mise en page que vous vous étiez énervé à peaufiner ? Vous allez me dire, que ça dépend du nombre de versions de Word et de la complexité du document. Mais dans l'ensemble, espérer conserver un document Word pendant dix ou quinze ans sans y toucher est illusoire. Le mieux est de l'ouvrir avec la dernière version de Word et de l'enregistrer dans le nouveau format pour ne pas tout perdre. À chaque changement de version. Parce que si vous n'arrivez pas à l'ouvrir, vous perdez tout. Alors qu'avec LaTeX, comme le source est du texte brut, même si LaTeX disparaît, je pourrai toujours lire mon source. Quoiqu'il arrive d'ici trente ou quarante ans, je pourrai toujours avoir accès au fond de mon document (sauf si je meurs avant, mais là n'est pas la question). Moyennant quelques modifications, je serai même capable de l'afficher sous un autre format, du html par exemple. Ou un autre format auquel je ne peux pas penser parce qu'il n'existe pas encore mais qui sera mieux. LaTeX m'apporte donc un confort inimaginable sur Word quant à la pérennité de mes documents.

Mais bon, un document que je considère important, c'est beau, mais il y a deux points à prendre en compte. D'abord, s'il est important, il ne faut pas que je le perde. Je dois donc le stocker à plusieurs endroits. Imaginez que mon ordinateur lâche et que je n'ai pas de sauvegarde, je perds tout en quelques secondes. Il faut donc qu'il soit sur mon ordinateur, sur au moins une clé USB et sur le plus d'endroits possibles. Ensuite, un document important ne s'écrit pas en trois minutes. Sinon, en cas de perte, il n'y a pas de problème, il me suffit de le réécrire. Et donc, qui dit document long à écrire, dit document qui évolue. Vous avez donc un document situé à plusieurs endroits, mais qui n'a pas évolué partout de la même façon. Par exemple, lorsque vous faites évoluer le document sur votre ordinateur. Au moment où vous l'enregistrez, il n'est pas dans le même état que ceux qui sont sur vos sauvegardes. Il faut donc bien s'y prendre pour gérer les versions du document principal et ses sauvegardes.

La façon la plus basique est de mettre la date dans le nom du document. Ou alors, de mettre les documents relatifs dans un répertoire avec une date dans son nom. Lorsque

votre document arrive à un état stable et que vous voulez le faire évoluer, vous recopiez le répertoire en changeant la date. Comme ça, si vous n'êtes pas satisfait de l'évolution, vous revenez sur le répertoire précédent. Mais il y a une limite, c'est pour connaitre la différence entre les deux versions. Il y a bien la possibilité d'ouvrir vos documents et de les comparer ligne à ligne. C'est pas glorieux comme solution, surtout si les documents sont très longs et que vous n'avez aucun outil pour vous aider. Pour jouer, nous allons complexifier la tâche. Imaginons que nous soyons plusieurs à faire évoluer le même document. Lorsque deux personnes ont fait évoluer le document, je ne connais pas de méthode simple pour récupérer les évolutions des deux personnes dans le même document. Dans les entreprises, c'est un cas fréquent et pour cela, il a été développé la « GED ». C'est un acronyme, il se prononce donc « gède » et signifie Gestion Électronique de Documents. Le technique est de mettre le document dans un espace commun entre les utilisateurs. Lorsqu'un utilisateur veut modifier son document, il bloque l'accès pour empêcher les autres utilisateurs de le modifier. C'est tellement limité comme solution que j'en rie encore. Je veut dire qu'au moment où j'écris ces lignes, je suis en train de rire, mais au moment où vous les lisez je suis encore en train d'en rire.

C'est là que Git (c'est un « G » dur et toutes les lettres se prononcent, ça se prononce donc « guite ») entre en jeu. Il est à l'origine créé pour les développeurs qui avaient besoin de gérer les versions de Linux. De la même façon que LaTeX utilise des fichiers textes pour créer un beau document à lire, les développeurs utilisent un fichier texte pour avoir un logiciel. Et donc, la gestion des sources des document LaTeX étant la même que la gestion des sources des logiciels, il est possible d'utiliser Git pour gérer les sources de LaTeX.

Et là, toutes les réponses aux questions précédemment posées apparaissent. Je veux gérer mon document avec Git ? C'est facile, je vais dans mon répertoire et j'utilise la commande `git init`. Je veux faire une sauvegarde ? Je vais sur ma clé usb et j'utilise `git clone`, et hop, j'ai une sauvegarde sur ma clé usb qui est gérée par git. Je veux enregistrer mes modifications en indiquant que j'atteins une étape à noter ? Je fais un petit `git commit` et j'ai deux versions différentes dans mon même répertoire. Je peux basculer d'une version à l'autre avec un petit `git checkout`. Je veux comparer deux versions d'un document, récupérer les modifications faites par quelqu'un d'autre et intégrer ses modifications dans mon document ? Rien de plus simple, `git diff`, `git fetch` et `git merge` sont là pour ça. Quand je parle de comparaison, il m'affiche tous les fichiers qui ont été modifiés. Avec pour chaque fichier, les lignes ajoutées, les lignes enlevées et les lignes modifiées. Par rapport à la comparaison des documents Word où vous devez comparer toutes les lignes sans aucune aide, ici, vous ne comparez que les modifications. L'apport est énorme. Il m'est possible d'accepter une modification et d'en refuser une autre. Bref, que du bonheur.

Imaginons que je sois en vacances et que je fasse lire un texte à quelqu'un. La réponse est sans appel, il y a une faute d'orthographe ou une phrase pas claire. Avec word, il faut que je sois sûr d'avoir la dernière version sur ma clé usb. Il faut aussi que la personne ait la même version de Word pour ne pas perdre ma mise en page (et encore, ce n'est pas forcément suffisant en fonction des pays et des langues des utilisateurs j'ai déjà eu

des surprises). Si toutes les conditions ne sont pas réunies, j'ai besoin d'un pense-bête. Avec LaTeX, aucun problème. Je modifie mon source tout de suite sur ma clé usb et je n'y pense plus. S'il peut lire ma clé usb, j'ai forcément la possibilité de modifier mon source. Que ce soit avec un bloc note Windows ou un notepad dans le pire des cas, ou avec un éditeur plus évolué, c'est possible. Une fois chez moi, `git commit`, `git fetch` et `git merge` et c'est fini. Même si je n'ai pas la dernière version de mon document sur ma clé usb, Git m'aidera à intégrer les modifications proprement. Comme Git se contente d'ajouter un répertoire pour lui dans mon répertoire de travail, la personne n'a pas besoin d'avoir Git pour que ce soit pris en compte. Il suffit de modifier mon document LaTeX pour que Git prenne les modifications en compte une fois chez moi. Il n'est pas possible de gérer un document Word avec Git. Parce qu'un document Word est un document qui intègre à la fois du texte et de la mise en page sous forme binaire <B>. Git est capable d'extraire le contenu d'un document Word, mais je perds la mise en page. Word m'empêche donc de gérer mes documents de façon simple.

Et la petite cerise sur le gâteau, c'est pour la mise à jour. En plus de mes sauvegardes, j'utilise gitlab qui m'offre un espace partagé sur Internet. Lorsque j'arrive à un résultat que je veux publier, je mets mes modifications sur gitlab. C'est simple, un petit `git push` fait l'affaire. Mais en plus, une fois que j'ai déployé ma version, il compile lui-même mon document pour que le fichier pdf soit téléchargeable sur mon site Internet. Vous pouvez automatiquement récupérer la dernière version de mon document sans que je n'ai rien à faire de spécial. Allez expliquer à Word que quand vous voulez le rendre public il doit se déplacer lui même sur votre site Internet. Quand vous y arriverez, nous en reparlerons et vous comprendrez la puissance de Git.

Vous allez me dire que c'est compliqué et que ce sont des commandes à écrire ? D'abord, le fait que ce soient des commandes permet des automatisations simples. Ensuite, il est possible de n'écrire que le début des commandes, des mécanismes aident à les compléter. Enfin, il existe des interfaces graphiques pour faire la même chose et éviter d'écrire des commandes. Mais je préfère me passer de la souris, je vais expliquer pourquoi maintenant.

La saisie de documents avec LaTeX

La supériorité du clavier sur la souris

Pour bien comprendre mon titre, il faut se situer dans le contexte. Il s'agit d'écrire des textes. Par exemple, je sais bien que pour certains jeux ou pour faire de la retouche d'images, la souris est irremplaçable. Il doit exister d'autres exceptions, mais ici, je me concentre sur la rédaction de documents. Le fait est, que pour écrire des textes, il faut utiliser un clavier. Je sais, il existe des claviers virtuels qui permettent d'écrire à la souris, mais ils ne sont pas productifs.

<B>. Ce n'est plus tout à fait vrai avec le format docx qui peut être utilisé par les dernières versions de Word.

Dans des textes avec des images, la souris peut sembler un avantage. C'est vrai que pour faire un panneau publicitaire avec des images à déplacer, la souris est pratique. Elle permet de voir directement le résultat du déplacement quand il se fait au feeling. Mais pour des textes écrits avec Word, c'est une autre paire de manches. J'ai de très mauvais souvenirs d'images que j'ai dû déplacer à la souris dans Word. Ou pour les redimensionner aussi. Et je ne parle pas de souvenirs d'images qui se déplaçaient sans que je comprenne pourquoi. Avec LaTeX, j'ai le nom de mon image que je déplace comme je le veux. Je lui dit aussi quelle taille elle doit faire si je veux la redimensionner. Par exemple, la commande `\includegraphics[width=\textwidth]{image}` permet simplement d'inclure une image en lui disant de faire la largeur du texte. Sans avoir à m'énerver si la capture d'écran dépasse la largeur de ma feuille. Non, vraiment, l'inclusion d'images dans LaTeX est reposant pour moi.

Mais le plus gros problème avec la souris est que j'ai horreur de basculer entre la souris et le clavier. C'est une perte de temps et de concentration. Lorsque j'écris, mes yeux savent où est le curseur, mes mains sont sur le clavier.

Si je veux utiliser la souris, j'ai besoin de lever ma main et de la mettre sur la souris. Ensuite, mes yeux doivent chercher le pointeur de la souris à l'écran, puis rechercher l'endroit où je veux le mettre, le déplacer et faire ce que je veux avec ma souris. Une fois que c'est fait, je dois remettre ma main sur mon clavier, me demander où j'en étais avant d'utiliser ma souris et recommencer à écrire. C'est un processus qui est assez rapide, mais qui est quand même une perte de confort et de concentration. Surtout si je dois le répéter souvent. Tout procédé qui me permet de me passer de la souris est donc une diminution de ma fatigue. C'est donc un confort supérieur, même si vous ne le remarquez pas avec l'habitude. Word m'oblige à basculer constamment entre le clavier et la souris. C'est, par nature, contre-productif.

Vous considérez probablement que j'exagère. Vous avez l'habitude de passer de la souris au clavier et vous ne remarquez pas de problème. Le fait que vous vous soyez habitué à augmenter votre fatigue ne signifie pas qu'elle n'augmente pas. Pour vous faire prendre conscience du changement, vous pouvez faire un test simple. Vous comptez le nombre de « m » dans cette page. C'est un exercice que vous n'avez pas l'habitude de faire, au début vous aurez du mal, ensuite vous serez plus rapide. Une fois que vous avez fini, vous comptez le nombre de « r » dans cette page. Là, vous allez comprendre le problème. En changeant d'exercice, vous avez deux difficultés. La première est de vous entrainer à repérer les « r ». La seconde est de vous forcer à ignorer les « m » que vous vous étiez habitué à remarquer. Le changement d'exercice est une augmentation de la fatigue au même titre que le passage de la souris au clavier. Ce n'est pas un effort énorme, réalisé une fois. Mais il est trop souvent répété dans l'utilisation de Word. L'utilisation de LaTeX, me permet d'utiliser le moyen que je veux pour saisir mon texte. Ce qui me permet de choisir un outil limitant au maximum l'utilisation de la souris. C'est donc une augmentation considérable de mon confort.

De tous les outils que j'ai vu, il y en a beaucoup qui sont prévus pour aider la saisie de documents LaTeX. Je reconnais que Kile et TexMaker ont beaucoup de qualités pour saisir les commandes des documents LaTeX. Mais seuls Vim et Emacs permettent de se

passer entièrement de la souris. Oui, vous avez bien, lu, avec Vim et Emacs, il est possible d'oublier la souris. Et donc, ce sont les deux seuls éditeurs que je vais prendre en compte. Ces deux éditeurs sont très différents dans leur conception. Vim est très déroutant, car en plus d'apprendre des raccourcis clavier, il faut assimiler son fonctionnement. Il possède un fonctionnement basé sur des états. Une même combinaison de touches va avoir des effets différents en fonction de l'état dans lequel Vim est. J'adore cette façon de procéder, ça apporte une puissance énorme, mais certains ne s'y font jamais. Alors qu'Emacs offre un fonctionnement standard, en contrepartie, il y a besoin d'utiliser des combinaisons de touches beaucoup plus complexes. Certains adorent, pour ma part, je ne m'y suis jamais habitué. Les deux permettent de faire la même chose, mais de façons très différentes. Certains détestent les deux, peu adorent les deux. Leur seul point commun est la possibilité de bannir la souris (ils sont aussi tous les deux des logiciels libres destinés à des informaticiens).

Pouvoir se passer de la souris est vraiment bien. Car en plus du confort de saisie déjà évoqué, se passer entièrement de la souris offre des avantages supplémentaires. D'abord, s'il n'y a plus besoin de souris, il n'y a plus besoin de bouton à cliquer. Et donc, tous les menus et boutons peuvent disparaître et vous laisser toute la place pour votre texte. Quand je vois le gaspillage d'écran pris par les boutons de Word, c'est énorme. Surtout sur un ordinateur portable avec un petit écran. Pour continuer avec le portable, débrancher la souris a plusieurs avantages. D'abord, ça augmente le temps d'utilisation de la batterie. Puis dans certaines conditions, par exemple dans un train, la souris est loin d'être pratique.

Pour en finir avec mes critiques de la souris, c'est mauvais pour la santé. Beaucoup de personnes qui utilisent la souris intensément ont des problèmes de poignet. C'est le syndrome du canal carpien : il y a un nerf dans le poignet qui est trop comprimé et qui finit par engourdir la main. Il n'y a rien de très grave, il suffit de se faire opérer et de prendre trois semaines d'arrêt maladie. C'est donc mauvais et pouvoir n'utiliser que le clavier protège mon poignet et est meilleur pour ma santé.

Maintenant, je vais vous montrer en quoi Vim me facilite la vie. Parce que, non seulement Vim me permet de me passer de la souris, mais en plus il me permet des choses inimaginables avec Word. Disons que si vous savez le faire, je ne suis pas capable d'imaginer réussir à le faire. Ce qui est le plus important, puisque c'est moi qui utilise LaTeX et c'est vous qui affirmez que Word est plus simple pour moi. Après avoir montré les avantages de LaTeX sur Word, je vais parler de la saisie du texte. Je vais montrer pourquoi ce qui resemble à une contrainte n'en est pas une avec un bon éditeur de texte.

La puissance de Vim

Contrairement à tous les autres éditeurs de texte, il est absolument impossible d'utiliser Vim (toutes les lettres se prononcent, ça se prononce « vime ») sans l'apprendre. Je ne vais pas vous apprendre à vous en servir, mais je vais vous en présenter le principe. Pour pouvoir ensuite montrer ce que ces principes ont de puissant une fois assimilés. Le côté contre-intuitif de Vim est évident. Imaginons que vous vouliez écrire « bonjour » dans un

nouveau document en utilisant Vim. Vous ouvrez Vim, vous écrivez « bonjour » et vous regardez le résultat. Vous avez écrit « njour » sur la deuxième ligne du document. Toute personne normalement constituée doit se demander ce qu'il s'est passé. Que sont devenus les deux premières lettres ? Pourquoi sur la deuxième ligne ? La réponse est simple, je l'expliquerai, mais pour ça, j'ai besoin de commencer par présenter son fonctionnement.

Vim est un utilitaire modal, c'est à dire qu'il possède des modes, ou états. Il a un comportement différent en fonction du mode dans lequel il se trouve. Pour simplifier, je dirai qu'il possède quatre modes : normal, insertion, ligne de commandes et visuel. Si nous prenons la touche [w] par exemple, elle va avoir les comportements suivants. Si vous êtes en mode normal, le curseur va se déplacer jusqu'à la première lettre du mot suivant. Si vous êtes en mode insertion, vous allez écrire un « w » dans votre texte. Si vous êtes en mode ligne de commandes, c'est pour enregistrer votre document. Si vous êtes en mode visuel, vous allez sélectionner le texte jusqu'à la première lettre du mot suivant. Vous avez donc : une touche, quatre actions possibles. Alors qu'avec n'importe quel autre éditeur de texte, vous êtes obligé d'avoir quatre combinaisons de touches distinctes pour pouvoir distinguer les actions. Comme il est possible d'utiliser des combinaisons de touches avec Vim, cela augmente le nombre de possibilités de contrôler votre texte avec un clavier. Je suis donc d'accord sur le côté perturbant, mais il est possible de s'y habituer. Je n'en parlerai plus, je me contenterai de montrer ce que sa maîtrise peut apporter.

Commençons par le mode normal. C'est le mode par défaut, c'est celui dans lequel se trouve Vim lors de son ouverture. Il n'est pas utilisé pour écrire mais pour se déplacer et modifier du texte. Par exemple, la touche [}] permet d'aller à la fin du paragraphe, la touche [G] à la fin du document et la touche [*] sur la prochaine occurrence du mot sous le curseur. Oui, c'est puissant. Si dans votre texte vous avez plusieurs mots « toto », vous allez sur l'un d'entre eux par le moyen qui vous convient le mieux. Ensuite, en appuyant plusieurs fois sur [*], vous allez vous déplacer sur chacun d'entre eux les uns après les autres. Pour le côté modification, vous avez généralement besoin de combiner des touches. Par exemple, en faisant [d] [d] vous supprimez la ligne courante. En faisant [d] [}] vous supprimez tout depuis le curseur jusqu'à la fin du paragraphe. Il suffit d'appuyer sur deux touches consécutives : avec Word, c'est plus délicat. Mais il existe quand même des possibilités avec une seule touche. Comme la touche [J] qui sert à joindre deux lignes. Ou la touche [p] qui sert à copier une sélection dans le texte. Il y a aussi la commande [.] qui est trop souvent négligée. Cette simple touche permet de ré-exécuter la dernière modification qui a été faite. Par exemple, vous avez un mot que vous voulez masquer pour anonymiser votre document. Vous faites « ciwxxxxx » pour remplacer votre mot par « xxxxx ». Ensuite, plus loin dans votre texte, vous voulez faire la même chose avec un autre mot. Là, c'est très simple, vous vous placez sur cet autre mot et vous faites [.]. Et c'est fait. Vous pouvez recommencer avec d'autres mots si vous voulez.

Ensuite, vous avez le mode insertion qui est celui dans lequel vous écrivez votre texte. La majorité des touches se contente d'insérer une lettre dans le texte. C'est ce que vous avez l'habitude de faire quand vous écrivez, même Word est capable de le gérer. Il n'y a donc pas grand chose à en dire, mais il y a quand même des trucs intéressants. Par exemple en faisant [Ctrl]+[w] permet d'effacer le mot à gauche du curseur (ça me manque

beaucoup dans Word). De même, Ctrl+u permet d'effacer tout jusqu'au début de la ligne. L'un des aspects les plus intéressants du mode insertion est l'utilisation des digraphes. Avec un clavier, vous avez accès à certaines touches, mais en français, il y a des caractères plus complexes à saisir que d'autres. Par exemple, le « Ç », le « œ » ou le « À ». Chez moi, avec Linux, c'est super simple, mais au boulot, avec Windows, je ne sais pas faire. Dans Vim, les digraphes permettent, comme leur nom l'indique, de saisir n'importe quel caractère à l'aide de deux touches. C'est aussi utilisable pour des caractères de l'alphabet grec, arabe, hébreux ou russe. Il y a beaucoup de caractères déjà définis, mais il est possible d'enregistrer ses propres combinaisons de touches. Les digraphes se saisissent à l'aide de la combinaison Ctrl+k. Par exemple, en faisant Ctrl+k C , vous avez « Ç ». De même, en faisant Ctrl+k o e vous avez « œ » et Ctrl+k A ‘ vous avez « À ». C'est aussi simple à écrire qu'à retenir tellement c'est logique. Alors qu'avec Word, c'est toujours l'horreur, il y a des combinaisons de touches que je n'arrive jamais à mémoriser. La seule chose que j'ai trouvée est de saisir un mot avec le mauvais caractère et de demander au correcteur de Word de me mettre le bon (il y a aussi la possibilité d'insérer un caractère spécial en fouillant dans les menus, mais c'est très lourd). Ce qui est quand même ridicule.

Maintenant, nous pouvons comprendre le problème de la saisie initiale de « bonjour ». Lorsque vous ouvrez Vim, vous êtes en mode normal. En mode normal, b déplace le curseur sur la première lettre du mot (ou du mot précédent si vous êtes déjà sur la première lettre du mot). Donc, sur un document vide, elle ne fait rien. Ensuite, o passe en mode insertion en insérant une ligne sous le curseur. Donc, vous vous trouvez sur la deuxième ligne du fichier en mode insertion. Puis, les autres touches s'affichent normalement à l'écran. C'est perturbant, mais ça n'a rien de magique.

Pour en revenir plus particulièrement à LaTeX, le mode insertion est aussi celui dans lequel les commandes LaTeX doivent être écrites. Par exemple, si nous en revenons à l'exemple du siècle. J'ai séparé le fond de la forme et je dois maintenant écrire `\siecle{iii} siècle` pour écrire IIIe siècle. Cela vous semble lourd, Vim est là pour m'aider. À chaque fois que j'écris « SIE » Vim me remplace ces trois lettres par `\siecle{<+siecle+>} siècle<++>` en positionnant le curseur sur « <+siecle+> ». Je n'ai plus qu'à écrire « iii » qui est le numéro du siècle et à taper Ctrl+J pour mettre le curseur sur le « <++> » derrière siècle et continuer ma saisie. Pour récapituler, pour écrire « IIIe siècle » j'ai eu besoin d'appuyer sur huit touches (dont deux simultanément). Et je peux continuer ma saisie. C'est à la fois simple, efficace et ça ne perturbe pas ma concentration. Avec Word, il faut appuyer sur onze touches (les trois « i », le « e », l'espace et les six lettres de « siècle »). Et il reste la mise en forme des « i » en petites capitales et du « e » en exposant. La mise en forme m'obligeant à penser à autre chose pour pouvoir chercher dans les menus. Pour me redemander ce que je voulais écrire une fois la mise en forme faite. Avec Word, c'est donc plus long qu'avec LaTeX. Alors que tout le monde est persuadé de l'inverse. C'est surtout une méconnaissance des bons outils qui donne cette impression.

Le mode ligne de commande a surtout pour but d'exécuter des traitements plus complexes. Par exemple, « :1,.g/toto/m$ » déplace toutes les lignes qui contiennent « toto »,

entre le début du fichier et le curseur, à la fin du fichier. Ou alors, « :%g/toto/s/titi/tata/ » remplace tous les « titi » par des « tata » sur les lignes qui contiennent « toto ». Un petit dernier avec « %g/toto/-1j » qui joint toutes les lignes qui contiennent « toto » avec la ligne qui la précède. Vous trouvez que c'est compliqué comme commandes et que j'aime me compliquer la vie. Mais c'est parce que vous ne les connaissez pas et ne les comprenez donc pas. Mais réfléchissez un peu différemment et acceptez le fait que je les comprenne et puisse les adapter. Comparez le nombre de touches que j'ai besoin d'utiliser avec le résultat obtenu. Avec Word, c'est facile de remplacer « titi » par « tata » dans tout le fichier. Mais comment faites-vous pour ne faire le remplacement que si la ligne contient « toto » ? En appuyant sur seulement quelques touches ? Ou en quelques clics de souris ? Ce sont des exemples qui semblent tordus, car ils ne sont pas souvent utilisés. Mais ils illustrent la possibilité de faire des modifications très puissantes en fonction de mes besoins.

Et enfin, le mode visuel, qui est le dernier de la série. C'est le mode qui permet de sélectionner du texte en le surlignant, comme vous le feriez à la souris. Une fois le texte sélectionné, il peut être copié ou coupé. Mais il permet surtout de faire des traitements intéressants en le combinant avec les autres modes. Par exemple, si vous faites [v] [i] [p], vous sélectionnez le paragraphe sur lequel se trouve le curseur. Ensuite, vous pouvez faire « !sort » et ça trie toutes les lignes du paragraphe par ordre alphabétique. Dans un roman, ça a peu d'intérêt, mais sur certaines listes, ça peut être très pratique. Ou alors, si vous faites [Ctrl]+[v] vous passez en mode bloc, c'est à dire que vous sélectionnez un rectangle. Si, une fois votre rectangle sélectionné vous faite [I] « toto » [Esc] alors, vous avez inséré « toto » sur chaque ligne de votre sélection. Pareil, dans un roman, c'est rarement utile, mais il existe beaucoup de cas où c'est intéressant. Un cas simple, c'est que pour LaTeX pour mettre une ligne en commentaire, il faut ajouter un « % » au début. Cela offre un moyen très simple de commenter un paragraphe. Une autre séquence de touches : « yypVr= » permet de souligner la ligne actuelle avec des signes « = ». C'est pareil, ça semble compliqué, mais je n'ai besoin d'appuyer que sur six touches pour faire ça. Si j'en ai souvent besoin, je peux mapper ces touches pour n'en avoir plus qu'une à utiliser. Je peux même personnaliser en mettant « map <F5> yypVr » dans mon fichier de configuration. Et ensuite, à chaque fois que j'appuierai sur [F5] [=] je soulignerai avec des signes « = ». Quel est l'intérêt d'utiliser deux touches au lieu d'une ? C'est pour pouvoir choisir mon caractère de soulignement. Maintenant, je peux appuyer sur [F5] [-] pour souligner ma ligne avec des tirets. Vous faites ça aussi simplement avec Word ? Moi pas.

Vous allez me dire que ce n'est pas parce que je ne sais pas faire que ce n'est pas possible. Je sais. Mais je répondrai d'abord que de savoir que d'autres personnes savent peut-être le faire ne m'aide pas à le faire. Or, dans le cas présent (et comme dans beaucoup d'autres cas), c'est moi qui ait besoin de le faire et personne ne sait m'aider. Je répondrai ensuite que si ni moi, ni personne autour de moi, ne savons faire, c'est que ça n'a pas la simplicité annoncée. Tout le monde se répète à me dire que Word est plus simple que LaTeX puisqu'il n'y a pas besoin d'apprendre Word. Cet argument tombe donc dès que je suis incapable de faire quelque chose avec Word et que personne ne peut m'aider.

Je répondrai enfin que si par bonheur quelqu'un est capable de m'aider à le faire avec la version de Word que j'utilise actuellement, je suppose fortement que la solution ne fonctionnera plus avec la future version de Word. Je sais, cette dernière phrase est un procès d'intention, mais c'est aussi le résultat de plus de vingt-cinq ans d'expérience de Word.

D'une façon générale, il y a souvent besoin de revenir en arrière lors de la rédaction de documents. Vous écrivez quelque chose, ça ne vous plaît pas, vous effacez. Mais des fois vous voulez revenir dessus. Avec Word, vous pouvez revenir en arrière sur certaines modifications, mais pas sur toutes. Par exemple, vous écrivez « toto », « titi », « tata ». Ensuite, vous effacez « tata », vous effacez « titi » et vous écrivez « tutu ». Si vous effacez « tutu » puis « toto », vous pouvez revenir en arrière sur « toto » et sur « tutu ». Mais vous ne pourrez jamais revenir en arrière sur « titi » ou sur « tata ». Alors que Vim, en conservant les branches, vous permet de revenir dessus. Mais Vim offre surtout un moyen performant pour retrouver ses modifications, c'est le voyage dans le temps. Vous passez en ligne de commandes en écrivant « :earlier 3m » et là, vous vous retrouvez avec votre document tel qu'il était il y a trois minutes. Vous vous rendez compte que vous êtes revenu un peu trop en arrière ? Pas de problème, en tapant « :later 15s », vous vous retrouvez avec votre document tel qu'il était 15 secondes plus tard. C'est à la fois simple et puissant, sans aucune commune mesure avec ce que Word apporte.

Vous allez me dire, OK, Vim, LaTeX et Git c'est puissant, mais bon, c'est lourd. Il faut trois environnements différents et la compilation avec LaTeX, c'est compliqué. Et bien non. D'abord, la compilation LaTeX n'a pas besoin de longs discours. Dans Vim j'appuie sur F2 et la compilation se lance. Si je veux voir le résultat de la compilation, j'appuie sur F3 et le texte pdf s'ouvre dans mon éditeur favori. Dire qu'appuyer sur une touche est compliqué est vraiment de la mauvaise foi. Entendons-nous bien. Vim n'est pas un compilateur, c'est un éditeur de texte, il m'offre juste une interface vers d'autres logiciels.

Il y a aussi l'interface vers Git qui est pratique. Passer par Vim pour gérer mes versions n'est pas forcément la meilleur solution. De même que vous ne gérez pas vos versions et vos sauvegardes des documents dans Word, il n'est pas forcément idéal de les gérer dans Vim. Mais il y a des cas pour lesquels utiliser Git à travers Vim est très pratique. Lorsque le rôle d'une commande de Git est d'afficher un résultat, alors la gestion de son résultat par Vim est intéressant. Vim n'est pas puissant pour gérer des documents mais pour afficher et modifier du texte. Par exemple, avec « Gitblame » je peux voir pour chaque ligne du fichier qui l'a écrite, quand et dans quelle version du document. Ou alors, avec « Gitdiff » je peux comparer de manière très claire les modifications apportées au fichier dans les différentes versions.

Il y a des interfaces écrites par d'autres, mais je peux me faire les miennes si besoin. Des fois c'est assez simple. Lorsque vous écrivez, il vous arrive d'avoir des soucis avec les mots que vous voulez écrire ? Par exemple, il vous arrive de vous demander comment conjuguer ou orthographier un mot ? Ou alors, vous n'êtes pas sûr de sa définition ou vous cherchez des synonymes ? Une petite recherche DuckDuckGo (je n'aime pas Google) est pratique, non ? Avec cette petite ligne : `:map <C-G> :!qutebrowser <cword> >& /dev/null<CR><CR> &` la recherche se simplifie. Elle permet de lancer qutebrowser avec une recherche Duck-

DuckGo sur le mot sous le curseur en faisant juste Ctrl+G. Vous n'êtes pas un peu impressionné ? Sachant que j'utilise DuckDuckGo, mais j'aurais pu choisir Wikipedia, le conjugueur, Google ou n'importe quel autre site. Je ne suis pas non plus obligé de choisir : je pourrais avoir trois combinaisons de touches pour les trois sites. Vous faites ça avec Word ?

Pour information, qutebrowser est un navigateur, au même titre que Fireofox ou Internet Exploreur, qui permet aussi de se passer le la souris. Avec des commandes inspirées de vim, il y a donc beaucoup de touches en commun. Ce qui simplifire considérablement l'apprentissage, je n'en parlerai pas plus car ce n'est pas le sujet ici.

Quand je veux comparer LaTeX et Word, vous trouvez que je passe beaucoup plus de temps à parler de vim que de LaTeX et de Git réunis. C'est normal, c'est l'outil que j'utilise le plus. Pour un document qui me demande dix heures de rédaction, le cumul des compilations doit me prendre moins d'un quart d'heure. De même, le cumul de la gestion des versions doit me prendre moins d'un quart d'heure. C'est donc l'éditeur de texte qui va être déterminant pour ma productivité. Comme je vous l'ai montré, vim me fait gagner beaucoup de temps lors de la rédaction. Alors que Word m'en fait perdre.

Éventuellement une section pour Emacs

Comme je l'ai écrit au dessus, je n'ai jamais réussit à m'adapter à Emacs. Cependant, dans mon infinie bonté, je daigne reconnaître Emacs comme un éditeur de texte digne de ce nom. N'ayant aucune envie ni capacité à dire du bien d'Emacs (mon infinie bonté a quand même des limites), je laisse cette partie. Si, un jour, quelqu'un veut rédiger cette section, il est possible que j'accepte son texte, sous certaines conditions. Cependant, je ne chercherai pas spécialement, il faudra que la personne me le demande.

Pour finir

J'espère que maintenant, vous acceptez que LaTeX n'est pas si compliqué que ça. Vous avez pu voir que bien maîtrisé, LaTeX est beaucoup plus pratique à utiliser que Word mal maîtrisé. Ce qui est mon cas. Vous devez comprendre que si vous préférez utiliser Word, j'ai de bonnes raisons de préférer LaTeX. Vous voyez peut-être une mauvaise raison, c'est en cas de problème. Vous allez me dire que tout le monde utilise Word alors que personne n'utilise LaTeX. Donc, en cas de blocage j'aurais plus facilement de l'aide avec Word qu'avec LaTeX. En fait, dans la pratique, c'est la conclusion inverse que j'ai remarquée. En cas de problème avec Word, je n'ai jamais pu trouver quelqu'un pour m'aider. Au boulot, aucun de mes collègues ne maîtrise Word et aucun support n'est fourni. Je n'ai jamais trouvé de réponse intéressante sur internet. À chaque fois que j'ai demandé de l'aide, j'ai dû me débrouiller pour faire autrement. Alors qu'avec LaTeX, j'ai quasiment toujours trouvé mes réponses sur internet. Les rares fois où je n'ai pas trouvé, j'ai pu poser ma question sur les forums usenet et j'ai toujours eu une réponse.

Après avoir montré tous LES inconvénientS de Word, maintenant, voyons quels sont

LE_ avantagE_ de Word. Dans certaines anciennes versions tout au moins, je ne sais pas si ça existe encore. Il y avait la possibilité d'écrire un texte bien défini. Puis de faire quelques combinaisons de touches précises. Et là, magie, devant nos yeux ébahis, nous avions accès à un flipper. Alors que je ne vois pas comment changer LaTeX en console de jeu, Word le permettait. C'est un peu ce que je lui reproche. Au lieu de faire peu de choses, mais bien, il fait tout, mais mal (le flipper de Word était nul, pour y jouer pendant des heures, il fallait vraiment être désespéré).

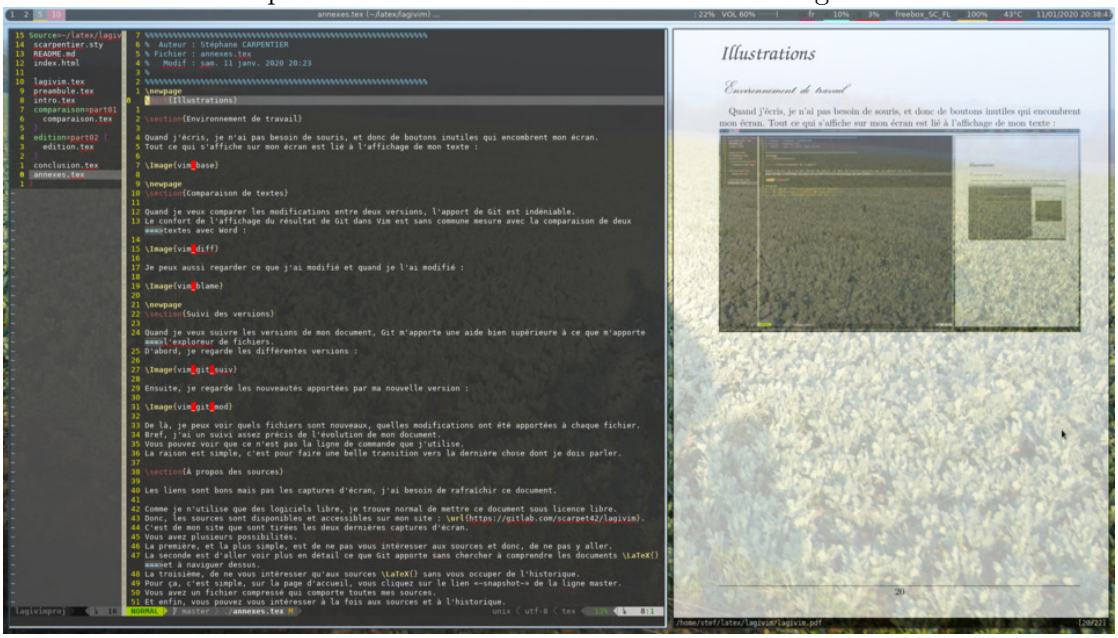
Je ne cherche pas à vous convaincre de passer à LaTeX. Si vous avez des arguments pour continuer à utiliser Word, tant mieux pour vous. Continuez à l'utiliser. Mais ce n'est pas pour ça que j'ai tort d'utiliser LaTeX. En dehors du milieu professionnel où je suis payé pour être contre-productif, bien sûr. Au boulot, je suis payé pour utiliser Word. Entre autre, ce n'est pas ma tâche principale, loin de là. C'est ça que je déteste le plus dans Word, j'ai beaucoup de travail et Word me fait perdre du temps sur des tâches secondaires. Mais chez moi, je fais ce que je veux. Et Word est banni. Je l'ai banni, mais il faut dire qu'il s'est banni un peu tout seul aussi.

D'abord, pour que je l'utilise, il faudrait que je commence par payer sa licence. Une fois acheté, pour le faire tourner, il faudrait que je paye une licence Windows. Une fois les licences Word et Windows payées, il faudrait que j'achète un nouvel ordinateur. Parce que mon ordinateur qui fait tourner LaTeX, vim et Git sous Linux est bien trop vieux pour faire tourner une version à jour de Windows. Par comparaison, LaTeX, Git, vim et Linux (et même Emacs) sont gratuits. J'ai le droit de donner de l'argent aux projets si j'en ai envie, mais je n'en ai aucune obligation. C'est pas pour dire, mais comme pour Linux : LaTeX, Git et vim, il y a moins bien, mais c'est plus cher.

Illustrations

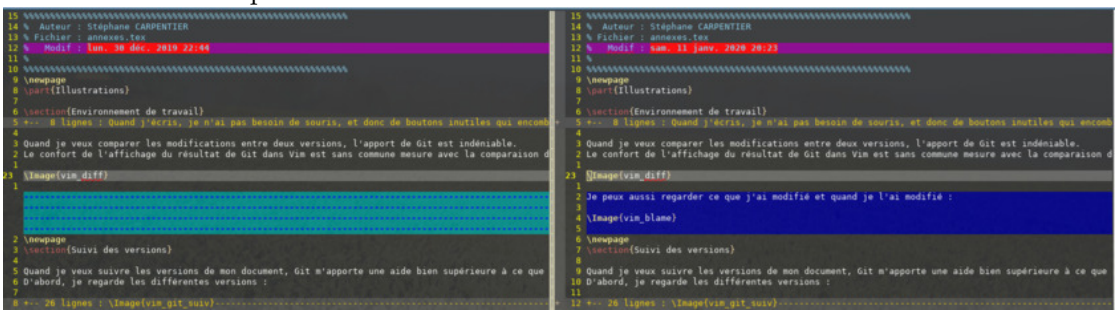
Environnement de travail

Quand j'écris, je n'ai pas besoin de souris, et donc de boutons inutiles qui encombrent mon écran. Tout ce qui s'affiche sur mon écran est lié à l'affichage de mon texte :

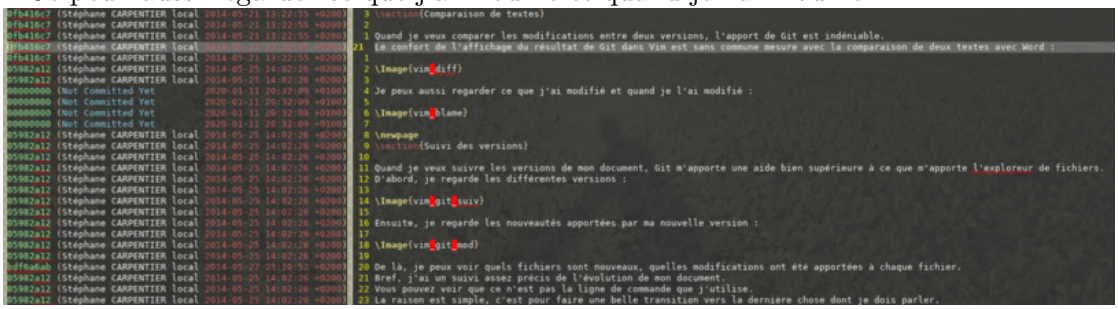


Comparaison de textes

Quand je veux comparer les modifications entre deux versions, l'apport de Git est indéniable. Le confort de l'affichage du résultat de Git dans Vim est sans commune mesure avec la comparaison de deux textes avec Word :

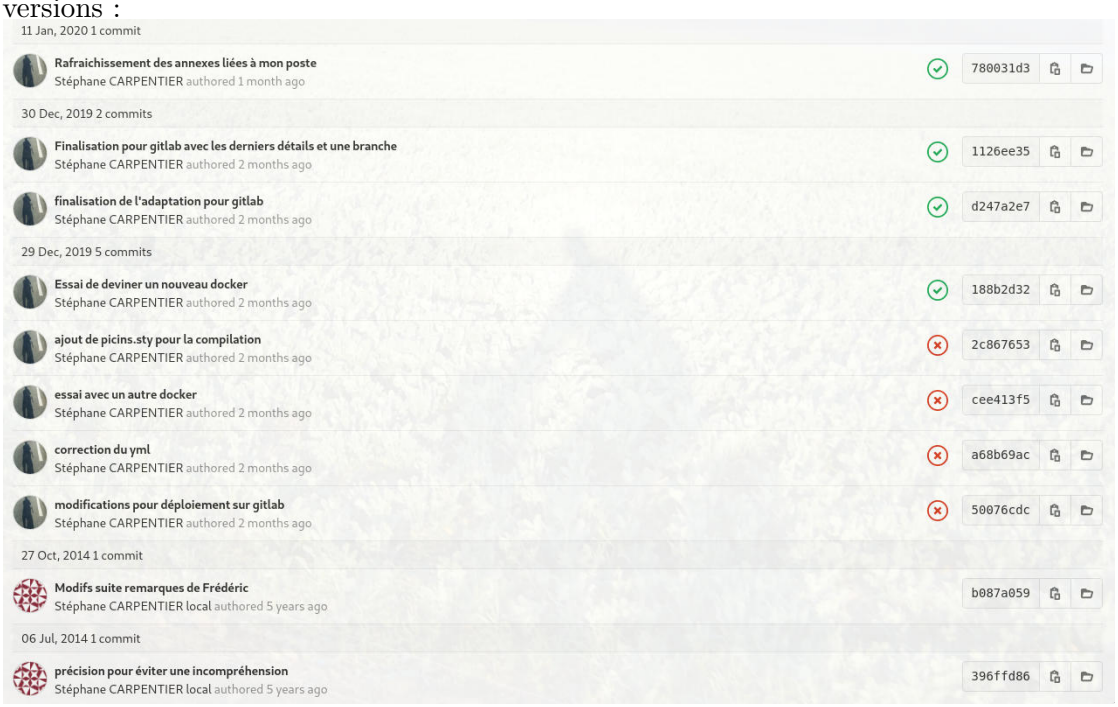


Je peux aussi regarder ce que j'ai modifié et quand je l'ai modifié :

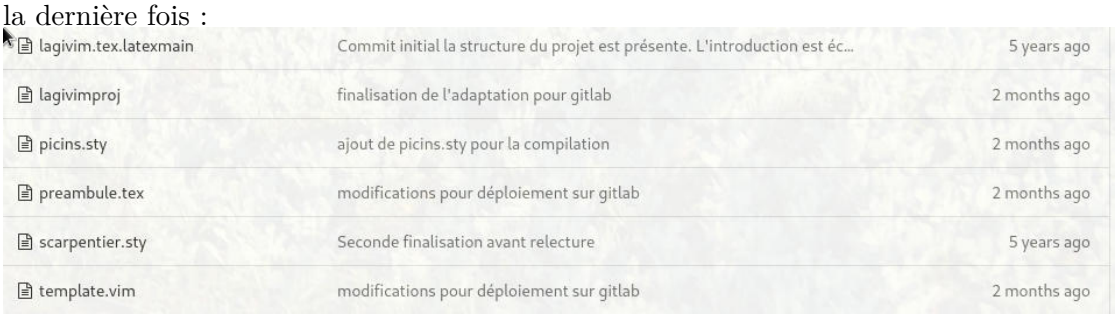


Suivi des versions

Quand je veux suivre les versions de mon document, Git m'apporte une aide bien supérieure à ce que m'apporte l'exploreur de fichiers. D'abord, je regarde les différentes versions :



Ensuite, je regarde la liste des fichiers et avec quelle version ils ont été modifiés pour la dernière fois :



De là, je peux voir quels fichiers sont nouveaux, quelles modifications ont été apportées à chaque fichier. Bref, j'ai un suivi assez précis de l'évolution de mon document. Vous pouvez voir que ce n'est pas la ligne de commande que j'utilise. La raison est simple, c'est pour faire une belle transition vers la dernière chose dont je dois parler.

À propos des sources

Comme je n'utilise que des logiciels libre, je trouve normal de mettre ce document sous licence libre. Donc, les sources sont disponibles et accessibles sur mon espace gitlab : https://gitlab.com/scarpet42/lagivim. C'est de cet espace que sont tirées les deux dernières captures d'écran. Vous avez plusieurs possibilités. La première, et la plus simple, est de ne pas vous intéresser aux sources et donc, de ne pas y aller. La seconde est d'aller voir plus en détail ce que Git apporte sans chercher à comprendre les documents LaTeX et à naviguer dessus. La troisième, de ne vous intéresser qu'aux sources LaTeX sans vous occuper de l'historique. Pour ça, c'est simple, sur la page d'accueil, vous cliquez sur le bouton de téléchargement de la ligne master. Vous avez un fichier compressé qui comporte toutes mes sources. Et enfin, vous pouvez vous intéresser à la fois aux sources et à l'historique. Pour ça, vous avez juste à faire un `git clone https://gitlab.com/scarpet42/lagivim.git` (vous devez avoir installé Git sur votre ordinateur).