

# Groff, L<sup>A</sup>T<sub>E</sub>X, Haiku OS et la recherche d'une informatique plus sobre

*Emmanuel Wald*

07/12/2024 | Catégories : [Informatique](#) & [Écologie](#)

Que faites-vous au quotidien sur votre ordinateur ? Si vous ne travaillez pas dans la production assistée par ordinateur et n'êtes pas informaticien-ne, l'essentiel de votre temps est probablement passé à lire, écrire (des courriels ou des messages sur les réseaux sociaux), écouter de la musique ou des podcasts et regarder des vidéos. Et, peut-être, jouer à des jeux-vidéos.

En somme, ce que vous faites sur votre ordi en 2024 ressemble quand même beaucoup à ce que vous faisiez (ou vos parents faisaient) en 2000, et en grande partie même en 1990. Ce qui a radicalement changé en 25 ans, ce n'est pas ce pour quoi nous utilisons des ordinateurs, ce sont les technologies que nous utilisons pour le faire.

Et pour utiliser ces technologies plus récentes, même si nous faisons globalement la même chose avec, nous avons *besoin* d'utiliser des ordinateurs beaucoup plus puissants qu'à l'époque, et de plus en plus puissants. Essayez d'écouter de la musique tout en tapant un gros texte sur un traitement de texte avec un navigateur ouvert pour vérifier vos courriels et des informations pour votre texte avec un ordinateur d'il y a 20 ans. Je suis à peu près sûr que vous allez jeter votre vieille tour par la fenêtre, poussé-e par la rage.

Alors, qu'est-ce qui a changé ? Étions-nous plus patient-es en 2000 qu'en 2025 ? Je ne le pense pas. Je pense que la cause de cette augmentation des besoins en puissance de calcul de nos logiciels du quotidien est le fait que le prix des technologies s'est effondré. Pour donner un exemple, ramené à sa valeur de 2023, le prix du GFLOPS<sup>1</sup> était d'un peu plus 1 000 \$ en 2000 ; il n'est plus que de 0,001 25 \$ en 2023. À quoi bon économiser une ressource presque gratuite ?

Presque gratuite, oui, mais pour qui ? Les ordinateurs sont moins chers aujourd'hui qu'il y a 20 ans, mais ils restent une grosse dépense. Et là est tout l'enjeu : continuer à vendre. Le capitalisme a besoin de ça, il ne peut jamais répondre à un besoin car un-e consommateur-riche sans besoin ne consomme plus. D'où la *fast fashion*, l'obsolescence programmée, les changements marketing, la gadgétisation... il ne faut jamais que la frénésie d'achat s'arrête. Et en informatique, cela s'est traduit par la vente d'ordinateur toujours plus puissants, qui ont entraîné la sortie de logiciels toujours plus gourmands, qui ont rendu les anciens ordinateurs obsolètes, qui ont poussé à l'achat d'ordinateurs toujours plus puissants, ... un cycle sans fin et délétère pour notre environnement et nos

---

<sup>1</sup>Le FLOPS est le nombre d'opérations en virgule flottante par seconde (de l'anglais *floating-point operations per second*), et donc, selon [Wikipédia](#) « une unité de mesure de la rapidité de calcul d'un système informatique ». Exprimé en GFLOPS, il donne le nombre de *milliards* d'opérations en virgule flottante par seconde.

portemonnaies. Et tout ça, rappelons-le, pour faire la *même chose*<sup>2</sup>. Quel gâchis !

À quoi sert toute cette puissance de calcul supplémentaire alors ? Pas à vous et moi. Prenons une page web normale. La page en elle-même demande très peu de puissance de calcul pour être affichée, mais au lieu d'être rédigée et hébergée avec des technologies simples qui font simplement leur travail, HTML, CSS et à la limite PHP, vous aurez des kilo- et des kilo-octets de scripts divers et variés, d'animations publicitaires et de trackers. Non seulement la page est plus lourde qu'il y a 20 ans, mais ce qui a été rajouté est essentiellement là pour nous emmerder. À la place d'être des utilisateurs-rices, nous sommes devenus-es les utilisés-es. Et en réaction, beaucoup d'entre nous installent des protections, la plus connue (et sans doute la plus utile) étant [uBlock Origin](#), qui certes allègent les pages (et au-delà, rendent le web supportable, merci aux programmeurs-euses qui créent tout cela !), mais alourdissent les navigateurs.

Et pour résister à mon petit niveau à tout ça, depuis quelques années, je m'amuse<sup>3</sup> à tester les solutions informatiques les plus économes possibles. J'ai passé quelques jours presque uniquement dans la console, principalement sur un ordi où j'avais installé une Debian « headless » (c'est-à-dire sans interface graphique). Je tape ce site web depuis un simple éditeur de texte (généralement [Mousepad](#), mais celui-ci en particulier est écrit sur [Pe](#), et toujours [Nano](#) pour la relecture), je le mets en ligne simplement avec la commande `scp`. J'ai essayé, sans grand succès je l'avoue, d'utiliser un gestionnaire de fenêtres par pavage (« tiling window manager »). Et je joue, dans des machines virtuelles, à limiter la puissance de mon ordi et faire tourner des systèmes d'exploitation alternatifs. Celui qui me plaît le plus est [Haiku OS](#).

Haiku OS n'est pas simplement un système d'exploitation léger, un des rares qui ne soient pas un UNIX (je n'ai rien contre UNIX, mais la différence est en soi attrayante). Haiku est à l'origine un clone de [BeOS](#), donc il devrait logiquement tourner sur des machines de 2001 à condition de se limiter aux logiciels de l'époque. Mais c'est aussi un système en version bêta. S'il est possible de faire beaucoup de choses, tout n'est pas possible et, surtout, tout n'est pas optimisé. Néanmoins, sa différence lui donne un grand charme. Ce n'est pas le genre de système qui reste en dehors de votre chemin, comme l'est, pour moi, [Manjaro](#) (mon principal système d'exploitation), mais c'est agréable de se prendre les pieds dans ce tapis en particulier (tant qu'on a un système plus fiable à côté, évidemment).

---

<sup>2</sup>Les jeux-vidéos sont peut-être une exception, car l'expérience de jeu est réellement différente si on joue avec un ordinateur puissant, capable de faire tourner des jeux visuellement et ludiquement riches. Mais il y a de très bons jeux qui tournaient en 2000 (pour ne donner qu'un exemple, 2000 c'est l'année où [Deus Ex](#) est sorti), donc il y a fort à parier qu'on pourrait créer des jeux incroyables même en limitant drastiquement la puissance de calcul nécessaire, et les ordis « gamer » capables de faire tourner les derniers AAA sont de toute façon *déjà* un marché à part.

<sup>3</sup>Je m'amuse, car je n'en ai plus besoin, et je n'ai pas la prétention de croire que les « petits gestes » sont réellement efficaces. J'ai aussi bien conscience de mon privilège pour ne pas l'avoir toujours eu ; j'en profite pour préciser que les critiques que je peux faire de l'informatique actuelle ne visent pas les utilisateurs-rices finaux-les, ni les professionnel-les de l'informatique, qui bien souvent n'ont pas d'autre choix.

Il faut dire que je n'ai pas besoin d'une grande puissance de calcul. Je joue peu voire pas, et généralement quand je joue c'est aux jeux de mon adolescence, donc peu intéressants. Je n'éditionne pas de vidéos, je n'utilise pas de logiciels particulièrement gourmands. Les deux logiciels qui prennent le plus de RAM sont mon navigateur web et mon traitement de texte. Pas de chance, ce sont aussi les deux plus utilisés (avec Thunderbird, qui est léger).

J'ai essayé de naviger surtout sur le [smolweb](#) ([Gemini](#) et [Gopher](#)), et d'utiliser des navigateurs plein-texte pour le WWW, comme [Lynx](#), ou simplement des navigateurs sans Javascript. Mais je finis toujours par revenir à Firefox ou par désactiver NoScript ; le smolweb est quand même assez limité niveau thématiques traitées, et le web « normal » est cassé, inutilisable sans script.

Niveau traitement de texte, j'ai eu plus de chance : j'ai de l'expérience en HTML et avec lequel j'ai rédigé mon plus gros et plus complexe document, ma thèse de doctorat. L<sup>A</sup>T<sub>E</sub>X est un langage de composition de documents ; plutôt que d'avoir un traitement de texte qui affiche à l'écran directement ce que le texte imprimé ou le PDF donnera, nous avons un éditeur de texte qui affiche le texte et des balises de code qui permettent ensuite de créer un PDF mis en forme. Par exemple, si sur LibreOffice pour mettre le mot « fédération » en italique il faut prendre sa souris, mettre le mot en subbrillance, et cliquer sur l'icône « italique » ; avec L<sup>A</sup>T<sub>E</sub>X il suffit de taper `\emph{fédération}` (pour « **emphasis** »). Les doigts ne quittent pas le clavier, et l'ordinateur ne fait quasiment aucun calcul à la volée, et calcule tout d'un coup lorsqu'il rend le PDF. Je préfère de loin cette deuxième solution : on écrit en pensant au sens plutôt qu'au rendu esthétique, et sur une machine peu puissante on peut rédiger de très gros documents sans souci. L<sup>A</sup>T<sub>E</sub>X tournait ainsi sans souci sur ma machine moyenne gamme, j'ai pu passer de Windows à Linux pendant ma thèse sans perdre du temps à remettre ma thèse en page, et la qualité de la mise en page a été soulignée par le jury, alors que j'y avais passé moins de temps qu'un collègue qui utiliserait MS Word.

Sur Haiku, les choses se vivent différemment. Comme je l'ai dit, les limitations de ce système d'exploitation ne sont pas dues uniquement à la faible puissance que j'ai allouée à la machine virtuelle qui l'accueille, mais aussi au fait qu'il n'est pas terminé. Firefox n'y est tout simplement pas encore disponible, et les alternatives ([WebPositive](#) et [Falkon](#) essentiellement) ne sont pas tout à fait au niveau<sup>4</sup>. Ils m'obligent donc à naviguer autrement, un tout petit peu. Le smolweb y est par contre tout à fait accessible, notamment *via* le très beau navigateur [Lagrange](#).

En revanche, L<sup>A</sup>T<sub>E</sub>X et LibreOffice sont tous deux disponibles. J'avoue ne pas avoir eu trop besoin de tester LibreOffice, puisque je ne l'utilise que dans un cadre pro, et que pour ça je n'ai pas le loisir de tester des logiciels en bêta. Je voulais cependant voir si L<sup>A</sup>T<sub>E</sub>X fonctionnait ; et la réponse est « oui, mais mal ».

L<sup>A</sup>T<sub>E</sub>X a en effet un défaut : il est vraiment lourd. Une installation raisonnable prend

---

<sup>4</sup>Pour être honnête, je n'ai jamais essayé Falkon en dehors de Haiku, je ne sais donc pas si les problèmes que j'y rencontre sont dus à Falkon ou à Haiku.

au moins 3 GB d'espace, plus si vous avez des besoins spécifiques<sup>5</sup>. Si l'on veut se protéger des aléas, une installation complète prendra plutôt 5 GB. Au prix du GB aujourd'hui cela peut sembler dérisoire, mais ce que je pense de l'utilisation des GFLOPS « gratuits » vaut aussi pour les GB « gratuits ». Et *a fortiori* dans une machine pensée pour être de petite taille. Et un document  $\text{\LaTeX}$  avec des renvois, une table des matières, ... exigera plusieurs compilations de suite, ce qui est assez lourd aussi en termes de puissance de calcul demandée.

En circulant sur les sites dédiés à Haiku pour voir ce que les gens faisaient pour utiliser confortablement  $\text{\LaTeX}$ , j'ai découvert un tout autre langage pour créer des jolis PDF : [Roff](#).

Roff est exactement le genre de logiciels que j'ai appris à aimer : un truc qui date des années 70, capable de très grandes choses déjà à l'époque mais qui a été oublié pour d'autres solutions plus « modernes » et qui n'a survécu que grâce à un usage de niche (ici, la mise en page des manuels UNIX). Roff, dans son implémentation GNU appelée [Groff](#), existe dans les dépôts de Haiku et, lisai-je, fonctionne bien. J'ai donc décidé de l'essayer.

Mais la chose peut être démotivante. Groff, comme  $\text{\LaTeX}$ , est tout un univers avec ses codes, son vocabulaire et son mode de fonctionnement. Comme  $\text{\LaTeX}$ , pour un novice, la somme de choses à comprendre simplement pour savoir si c'est une solution praticable pour soi est énorme, surtout si l'on n'a pas une formation dans l'informatique, comme moi. Mais contrairement à  $\text{\LaTeX}$ , il n'y a pas pour Groff de manuel pour apprendre à l'utiliser, juste une documentation, certes complète et bien faite, mais assez complexe de par sa complétude même.

Mais je me suis quand même lancé. En utilisant [la série de macros appelée « mom »](#). Et après quelques heures à tester ce système, je l'ai trouvé accessible, léger<sup>6</sup> et incroyablement puissant. Il permet de maîtriser la mise en page, la mise en forme, de collationner différents documents, de gérer les en-têtes et pieds-de-page, y compris recto verso, de mettre des titres, des liens, des notes de bas de page, marginales et de fin <sup>7</sup>, des tables (des matières, des figures, ...), des équations, ... Pour vous donner une idée du rendu, vous pouvez télécharger cet article mis en PDF *via* Groff et les macros Mom, en cliquant [ici](#) (et télécharger le code source et sa relative transparence en cliquant [ici](#); n'hésitez pas à ouvrir ce code source avec un éditeur de texte plutôt que votre navigateur qui risque de mal l'afficher).

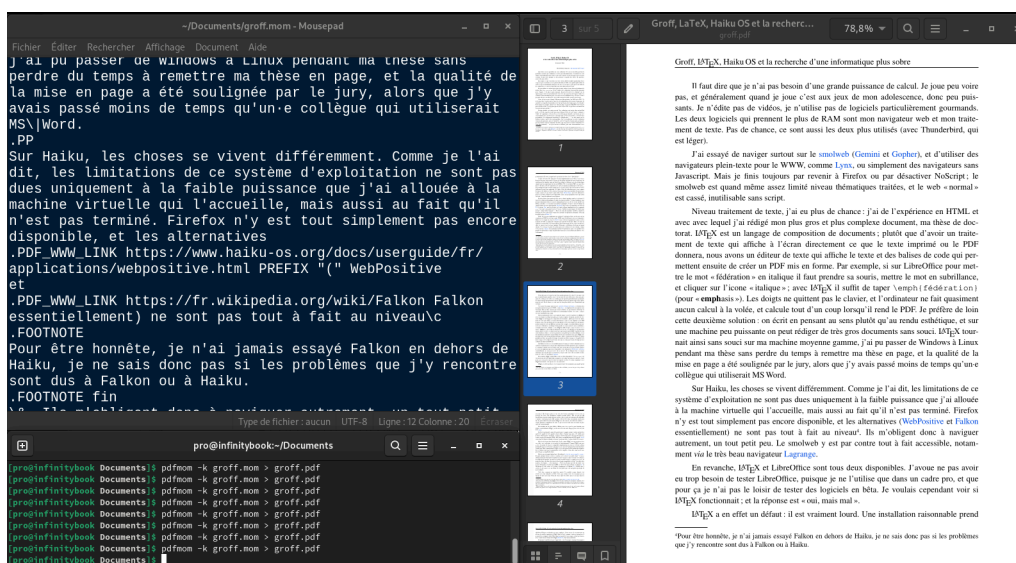
Voilà donc comment un logiciel des années 70, considéré comme dépassé s'est révélé tout à fait utilisable en 2024. Certes, par certains côtés il est marqué par son époque (il gère assez mal l'Unicode, donc taper des lettres qui ne sont pas dans les

---

<sup>5</sup>Ça peut paraître beaucoup, mais c'est équivalent au dernier [Microsoft Office qui demande 4 GB](#).

<sup>6</sup>Une des grosses différences entre  $\text{\LaTeX}$  et Groff est que le premier est un langage compilé là où le deuxième est un langage interprété. N'étant pas programmeur moi-même, je ne suis pas sûr de l'importance de la différence, mais je crois comprendre que ça a une grosse incidence sur la légèreté du second par rapport au premier.

<sup>7</sup>Même si utiliser les notes de fin dans un document fonctionnant par page devrait, sauf exception dûment justifiée, être puni de 5 ans d'emprisonnement et 75 000 € d'amende, m'est avis. Enfin, bref.



## Un exemple d'organisation d'un écran seul pour taper du Groff

alphabets d'Europe occidentale est assez complexe). Certes encore, il ne permet pas un niveau de contrôle équivalent à L<sup>A</sup>T<sub>E</sub>X. Mais si Groff continue à tenir ses promesses, il pourrait bien remplacer LibreOffice dans la majorité de mes usages, même professionnels, comme un mi-chemin entre L<sup>A</sup>T<sub>E</sub>X [Markdown](#), léger mais insuffisant.

Et puisque la meilleure façon d'apprendre, c'est d'enseigner, et puisqu'il manquait à mon avis une ressource pour faire ses premiers pas avec Groff, j'ai décidé d'écrire un tutoriel pour (faire) découvrir Groff. Un truc assez basique, mais qui permettrait ensuite en naviguant dans la documentation, de maîtriser cet outil. Ce n'est pas encore tout à fait terminé, mais attendez-vous à voir une page « Groff » dans le [dictionnaire de ce site](#) très bientôt !

Source : <https://wald.ovh/~Emmanuel/2024-12-07-groff.html>