

Journal : Licht 0.1

Posté par [chimrod](#) ([page perso](#)) le 13/02/18 à 15:07. [Licence CC by-sa](#)

Tags : aucun

Sommaire

- [Licht](#)
 - [Installation](#)
 - [Utilisation](#)
- [Réflexions sur les tableurs](#)
 - [SC l'ancêtre](#)
 - [SC-IM la continuité](#)
 - [LibreOffice](#)
 - [Les manques](#)
- [Le futur](#)

Bonjour à tous, j'ouvre aujourd'hui un journal pour vous présenter licht, un tableur en console.

Licht

C'est la première version que je diffuse, et après plus d'un an passé sur le programme, je me rends compte que je n'ai pas grand-chose d'aguideur à vendre :

- ça utilise l'interface sexy de ncurses,
- la prise en charge des fichiers openOffice est basique,
- le nombre de fonctions disponibles restent limitées.

En gros, ça suffit pour faire des petits calculs sans charger sa mémoire, mais il ne faut pas s'attendre à quelque chose de révolutionnaire. C'est d'ailleurs pour cela qu'il s'agit d'une version 0.1 : c'est une version minimale propose un minimum de fonction, mais qui est censé fonctionner.

J'avais commencé à écrire licht suite à un article dans un vieux GLMF, sur un tableur écrit en C en 200 lignes de codes. Ça avait l'air assez simple, et j'ai essayé de m'y mettre. Résultat des courses, après plus d'un an de temps libre passer à coder, je n'ai pas encore la totalité des fonctionnalités présentées dans ces deux pages (pas de camemberts, pas de graphiques) : je crois que j'ai dû me loucher quelque part...

À la place, j'ai un système de typage, un catalogue dynamique de fonctions (que je compte utiliser plus tard pour mettre en place un système de plugins), une gestion (partielle) des fichiers open document en lecture et écriture et surtout j'ai appris plein de trucs !

- Calculer des [jour julien](#)
- Comment me servir des GADTs en OCaml pour écrire un système de typage
- Écrire des grammaires (pour lire les saisies des utilisateurs, ou les formules stockées dans libreOffice)
- Des [arbres splay](#) (qui me fascinaient depuis ma lecture d'Okasaki),
- Et j'ai également eu l'occasion de réfléchir à ce qu'est un tableur (mais j'y reviendrai un peu plus loin).

Je suis content du code (qui n'est pas parfait, mais qui est assez organisé pour que j'ose le montrer), et j'ai pris plaisir à l'écrire, ce qui est l'essentiel (-: Maintenant si d'autres personnes trouve un intérêt, que ça soit dans l'utilisation ou dans le code, n'hésitez pas à m'en faire part, ça me fera plaisir !

Installation

Le plus simple pour l'installation est de passer par `opam` :

```
# sudo aptitude install opam libncurses-dev libiconv-dev libgmp-dev zlib1g-dev
$ opam pin add -k git licht http://git.chimrod.com/cgit.cgi/licht.git
```

Vous pourrez ensuite lancer `licht` accompagné ou non d'un fichier ods en paramètre.

Utilisation

Vous trouverez sur la page du projet [une petite description](#) de ce que l'on peut faire avec.

Attention ! J'ai dit que le support des fichiers ods était basique, aussi, je vous déconseille de réécrire un fichier ouvert : le fichier est complètement réécrit, et toutes les fonctions non gérées seront perdues (formatage, etc)

Réflexions sur les tableurs

Alors comme je n'ai pas grand-chose à dire sur ce que j'ai produit, je vais enrichir mon journal en présentant des choses que je n'ai pas écrites. Pour commencer, je vais vous présenter mes voisins.

SC l'ancêtre

[sc](#) est un tableur en console qui fourni les fonctions de base. Il offre même le luxe de pouvoir aligner le texte.

Il présente l'avantage de pouvoir sauvegarder ses fichiers dans un format texte, qui présente chaque formule comme une affectation.

Le programme trouverait ses origines en 1987, ce qui en fait une antiquité. Malheureusement, je n'ai pas réussi à retrouver les sources datant de cette époque. En termes de fonctionnalités, on a l'impression que ça n'a pas beaucoup évolué depuis, l'interface sobre en noir et blanc résume ce que l'on peut faire avec l'application.

On ne lui jettera toutefois pas la pierre, à part le fait de mélanger le formatage avec le contenu de la cellule, il a surtout le défaut de ne pas être connu !

SC-IM la continuité

scim est à sc ce que vim est à vim : une continuité et une avancée. Il propose de la couleur, une interaction avec gnuplot, et le prise en charge des formats excels. Regardez la page du [projet](#) qui est assez alléchante ! Une belle attention a été portée aux fonctionnalités de l'hm ce qui le rend assez intéressant.

J'avais testé un peu avant de commencer à coder licht, et le projet était encore un peu jeune. Il semble avoir bien muri si l'on en croit les changements qui ont eu lieu depuis.

Dans les points négatifs, je citerai une gestion incompréhensible des copier/coller (en tout cas pour moi), et s'être concentré sur le format xlsx plutôt que ods.

LibreOffice

Le mastodonte des tableurs, je le cite ici car il est communément utilisé par tout le monde. Je m'en suis servi de référence quand je me posais des questions existentielles en écrivant licht.

C'est la référence pour la liste des fonctions disponibles, et d'outils pour analyser et mettre en forme les données (y compris les fameux *tableaux croisés dynamiques* que ma femme aimerait me voir implémenter dans licht !).

Il s'offre également le luxe d'internationaliser le nom des fonctions disponibles (même si cela m'a toujours plus perturbé que simplifié la vie). Et je l'ai utilisé pour tester le comportement du tableur sur quelques cas limites, ce qui m'a permit de tomber sur des incohérences...

Allez ? Bon, je vous explique en deux mots : [covariance et contravariance](#). C'est plus clair ? Là je vois les experts du typage qui lèvent les yeux en disant : « mais qu'est ce que ça vient faire dans un tableur ? » Ok, allons y pour la version longue : quand on construit une liste d'éléments, il faut que toutes les valeurs soient du même type. Le problème se pose avec les date, car elles sont gérées en interne comme des nombres. Quel est donc le résultat d'une fonction qui mélange des dates et des nombres, par exemple SUM(A1:A2) ? Et bien avec LO, selon le type de la dernière valeur, nous pourrons avoir un résultat sous forme de date ou de nombre... Fichtre ! (Pour ma part, fait le choix que la date étant un format plus spécialisé que le nombre, il est avait une priorité plus importante, si vous avez une solution plus élégante, je suis preneur !)

Les manques

Bon, je ne vais pas lister tous les tableurs existants, Wikipédia fera ça mieux que moi. Je voudrai revenir sur ce que devrait être un tableur. Vraiment. Pas sûr ce qu'il faudrait ajouter pour avoir le truc à la mode, mais ce qu'il faut pour qu'un tableur redevienne ce qu'il est censé être : un éditeur de formules. J'ai cité ci-dessus le problème de la covariance et la contravariance qui est un sujet que l'on retrouve dans la programmation, essayons d'aller plus loin et voir ce qui manque vraiment pour qu'une feuille de calcul soit capable de faire vraiment des calculs (en mode Oulipo, parce qu'on peut écrire ses calculs dans tous les sens)

Pour vous guider dans mon raisonnement, je vais partir de la formule ci-dessous :

$$P(x) \times f(y) + (1 - P(x)) \times g(z)$$

À première vue, il s'agit d'une formule toute simple composée d'additions et de multiplications. En fait, si la fonction `P` (pour Prédicat) renvoie 1 ou 0, alors cette formule et la version mathématique du code ci-dessous :

```
if (P(x)) {
    f(y);
} else {
    g(z);
}
```

C'est d'ailleurs pour cela que la fonction IF (ou SI en version francisée) est disponible dans les tableurs évolués : il s'agit bien d'une formule déguisée.

En travaillant simplement sur des formules mathématiques, il est possible d'exprimer des concepts que l'on retrouve dans les langages informatiques (la réciproque n'est cependant pas vrai) Malheureusement, aucun tableur ne permet aujourd'hui de saisir de telles formules. Des fonctions sont disponibles pour pallier ce manque, mais certaines fonctions sortent du cadre de ce qu'un tableur devrait proposer (même si cela permet de transformer un tableur en une [machine de turing](#))

J'avoue ne pas trop maîtriser le domaine de la calculabilité, et j'ouvre juste le sujet en laissant la porte ouverte.

Le futur

J'ai envie de continuer à travailler le côté *formule*. Je ne cherche pas la parfaite compatibilité avec les fichiers LibreOffice (mais je suis preneur de toute contribution qui irait dans ce sens, que ça soit pour enrichir le catalogue des fonctions existantes, ou pour une meilleur intégration du format), et proposer un format un format incompatible ne me gêne pas.

Sans changer la syntaxe en profondeur, je pense qu'il est possible d'écrire des formules telles que :

```
A1(x; y; z) = (x + y) / z
A2 = A1(1; 2; 3)
```

c'est à dire laisser l'utilisateur composer lui même ses propres formules et pouvoir enrichir le catalogue de fonctions présenté par défaut. Dans l'idéal, toute [fonction élémentaire](#) devrait ainsi pouvoir être saisie directement. Le tableur pourrait proposer la fonction SOMME et PRODUIT qui permet de calculer, non pas la somme d'une liste d'élément (SUM(A1:D6)), mais la fonction telle qu'on la retrouve en mathématique :

$$\sum_{i=1}^{99} A1(i)$$

que l'on pourrait coder comme la fonction : B2=SUM(1; 99; A1(_))

Avec cette simple évolution, on devrait être capable d'écrire une formule capable de calculer la nième décimale de pi, ou le nième nombre premier, si vous voulez prendre un peu d'avance sur la prochaine version, vous pouvez en discuter dans les commentaires :-)

J'ai également parlé de plugins, et c'est un point que j'ai envie d'ajouter rapidement, pour pouvoir ajouter des fonctions spécifiques qui ne rentrent pas dans le module standard. Le code pour écrire une nouvelle fonction est [assez simple](#), et donc peut être facilement enrichi. Je pense que je vais commencer par coder les jets de dés.

Et après ? Bah je ne sais pas. Je suis en train de quitter mon poste actuel vers un autre dans lequel je lâche complètement la technique, et ne sais pas trop quel sera mon emploi du temps dans les mois à venir. Cette première version est donc une manière de mettre au propre les côtés sur lesquels je travaille (et une manière aussi de répondre à ma femme qui n'arrête pas de me dire « tu fais toujours plein de choses sur l'ordinateur, mais ça serait bien qu'on voit un peu le résultat ») !

Je me suis fait plaisir en codant, et envie d'explorer d'autres horizons, j'espère avoir réussi un peu à transmettre tout ça dans ce journal. J'apprends, comme nous tous, tout les jours, et cette annonce est aussi l'occasion d'avoir des retours, ou d'autres pistes qui peuvent venir s'intégrer dans cette réflexion sur les tableurs. La discussion est ouverte, c'est maintenant à vous !

Non, l'ancêtre est...

Posté par [lolop](#) ([page perso](#)) le 13/02/18 à 16:47. Évalué à 4 (+2/-0).

[Visicalc](#) (y'a aussi eu ensuite Lotus 123, Multiplan le précurseur d'Excel).

...

Python 3 - Apprendre à programmer en Python avec PyZo et Jupyter Notebook → <https://www.dunod.com/sciences-techniques/python-3>

Re: Non, l'ancêtre est...

Posté par [chimrod](#) ([page perso](#)) le 13/02/18 à 19:18. Évalué à 2 (+0/-0).

Exact, mais mon tour d'horizon n'était pas de présenter tous les tableurs existants (comme dit plus haut, Wikipédia fera ça mieux que moi), mais d'en présenter quelques-uns, dont j'ai pu m'inspirer ou qui peuvent valoir le détour. `sc` est encore maintenu, et disponible dans les dépôts de debian par exemple.

Intérêt ?

Posté par [mattell](#) le 13/02/18 à 19:09. Évalué à 1 (+0/-0).

Une question qui a le goût du troll mais qui est une vraie question :

De nos jours, quel est l'intérêt d'un tableur en console ?

Re: Intérêt ?

Posté par [chimrod](#) ([page perso](#)) le 13/02/18 à 19:25. Évalué à 2 (+0/-0).

Pour l'utilisateur, aucune idée. À part vouloir travailler dans une session ssh, ou alors être un fan du terminal je ne sais pas trop. C'est comme les roguelikes, certains continuent d'y jouer en console, voire de faire des interfaces qui imitent un terminal... tous les goûts sont dans la nature...

De mon côté, je l'ai plus pris comme un défi, dont la mayonnaise à pris. Et aujourd'hui, je sais bien que les tableurs font mieux que moi la mise en forme de calculs.

Pour un fana du scripting, on pourrait imaginer appliquer un script sur chaque cellule, pour réinjecter ensuite le résultat dans le tableau ou ce genre de chose, mais pour l'instant c'est juste une idée que je n'ai pas encore complètement développée.

Re: Intérêt ?

Posté par [Graveen](#) le 13/02/18 à 19:33. Évalué à 2 (+0/-0).

| Oui je n'imaginai même pas que cela puisse encore exister. Mais c'est bien d'avoir le choix. sur SC-IM, le plotting m'a impressionné :)

Note : les commentaires appartiennent à ceux qui les ont postés. Nous n'en sommes pas responsables.