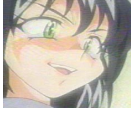


Journal : Vibe Coding lisp amateur

Posté par Phibrizo (site web personnel) le 15 mars 2026 à 23:50.
Licence CC By-SA.

Étiquettes : claude, common_lisp, vibe_coding



Cher journal,

Je ne suis pas un programmeur. Parfois je le regrette un peu, j'aurais peut-être mieux gagné ma vie, mais ma vie a pris un autre chemin. J'ai quand même quelques notions, j'ai commencé à écrire des programmes en Basic sur mon TO7, j'ai déjà écrit quelques lignes de code, notamment durant ma thèse mais bon, dire que je "sais" coder serait un peu exagéré.

Il se trouve aussi que j'aime le lisp, ne serait-ce que conceptuellement. J'écris aussi parfois quelques lignes d'assembleur pour m'amuser avec mon C64 et mon mega65. Mais c'est difficile de dépasser le stade du hello world.

Aussi, en ce début de XXIème siècle déjà bien entamé, j'ai découvert depuis peu la puissance de l'IA. Et, après avoir réfléchi, j'ai décidé de combiner mes hobby.

Je me suis dit que ce serait sympa d'écrire un assembleur 6502 en lisp.

Je me suis donc tourné vers l'oracle numérique des temps modernes (Claude, pour ne pas la nommer) et elle m'a répondu "Bonne idée!" (c'est toujours une bonne idée avec elle).

En moins de 5 minutes, elle m'a proposé une architecture qu'elle a ajusté après quelques questions et m'a demandé "on y va ?"

Avec toute la grâce dont je suis capable je lui ai répondu un truc comme "bah euh ouais..."

Et là, les choses se sont enchaînées. J'ai appris des termes comme "lexer", "parser", "IR", "frontend" et d'autres encore que j'ai sûrement oublié. Et je l'ai laissée coder.

Elle m'a sorti un premier fichier. Que j'ai immédiatement testé. Et qui a immédiatement planté. Mais en plantant le fichier a généré des erreurs et des Warnings. Je lui ai renvoyé. Elle a corrigé et en a généré un autre. Qui a a aussi planté. Et on a continué. Jusqu'à ce que le fichier passe sans erreurs ni warnings.

"Bien" me dit-elle, et on a continué.

Le cycle a continué. Après peu de temps j'ai proposé de lancer une batterie de tests automatisés. "Bonne idée!" m'a t-elle répondu à nouveau. Les tests ont continué, mais on avait cette fois une vue plus globale. Cette fois, plusieurs dizaines de fonctions se lançaient à la suite. Erreurs/warnings/retour à l'envoyeur. Jusqu'à ce que après un temps ridiculement bref, on atteigne le chiffre incroyable de 0 erreurs et 0 warnings. "On continue ?" me demanda t-elle

"Bah euh ouais..." ai-je répondu comme toujours.

Le code a continué à croître. Le nombre de fonctions augmentait, mais les tests impitoyables ont continué ramenant chaque fois le nombre d'erreurs et de warnings à zéro.

A un moment, on a eu un problème. Un des fichiers plantait en affichant un "end of file" et Claude était persuadée que le souci venait du téléchargement. Que mon fichier était tronqué .J'avais beau la contredire, elle n'en démordait pas.

J'ai alors utilisé mon IH (Intelligence Humaine) et j'ai testé le code avec une autre implémentation. J'étais sous SBCL, j'ai testé sous CLISP. Le code plantait toujours, mais avec un autre message d'erreur. Claude a alors identifié le bug, un des commentaires finissait par être interprété comme une fin de fichier. Le bug était corrigé, le code avait incroyablement grossi, et on était toujours à 0 erreurs et 0 warnings.

J'ai demandé à Claude de déployer une batterie de tests avec les 2 implémentations. ça lui a permis d'identifier d'autres bugs. A l'issue de cette chasse impitoyable, on était toujours à 0/0 mais cette fois sous DEUX implémentations.

On a continué. Claude m'a mis en place asdf. Maintenant, je pouvais lancer les tests sous slime. A nouveau quelques bugs. Nouvelles corrections. Résultat 0/0 sous 2 implémentations et sous slime. Et pour plusieurs centaines de tests.

On a continué. Et finalement est arrivé le grand moment. J'ai lancé l'assembleur sur un fichier et il l'a assemblé.

Le fichier n'était pas bon. Mais après quelques nouvelles itérations... c'était bon.

Jusque là, je n'étais pas trop mécontent. Mais on a continué.

L'étape d'après, c'était un frontend qui permet de lancer l'assembleur lisp *depuis* un repl. Je n'ai aucune idée de si c'est une première mondiale ou si dix millions de geeks l'ont déjà fait avant moi mais peu importe. ça a marché. Je peux lancer un repl et programmer directement en assembleur dedans.

Je commençais à ce stade à connaître une certaine satisfaction.

On a déployé d'autres backend: Mega65 et X16. Les fichiers s'assemblent sur les 3 architectures et tournent sous émulateur. En tout cas pour les fichiers tests.

Pas mal, pas mal. "Ma petite Claude" ai-je demandé, pourrais-tu tant qu'à faire traduire la doc, l'architecture et rajouter un changelog ? "Bonne idée!" me répondit-elle.

Ne sachant plus quoi faire pour la soirée je lui a ensuite demandé "Pourrais-tu faire des scripts pour mettre au point une série de tests automatisés afin de comparer les fichiers générés par ceux assemblés par d'autres programmes ?", question à laquelle elle a répondu "Bonne idée, les tests de régression!". L'idée était si bonne qu'elle existait déjà, j'ai appris un nouveau mot.

Peut-être 30 minutes plus tard, la suite de scripts et de programmes de tests était prête

quelques minutes, plus tard, le verdict était sans appel: tous les fichiers générés par mon assembleur sont identiques à ceux générés par acme et par ca65. Il n'y a que les fichiers de macros qui ne passent pas car le programme de tests ne peut pas convertir leurs macros dans mon système de macros car ils sont trop différents et les scripts de conversion qu'elle a mis en place vers mon format pour acme et ca65 (je ne vous ai pas parlé des scripts de conversion ?) ne peuvent pas les convertir. Enfin, peut-être le pourrait-elle si j'insistais, mais mon objectif n'est pas de réécrire acme et ca65, mon propre assembleur me suffit.

parce que oui, entre la poire et le fromage Claude m'avait aussi implémenté un système de macros. Mais bon il est presque secondaire car je peux directement utiliser les macros Lisp s'il le faut.

Bon, pour la suite il y aura d'autres choses à voir. Davantage de fichiers de tests, sur davantage d'architectures. D'autres backend. Z80 et M68k pour commencer, ensuite qui sait X86 et ARM32/64 mais ce sera plus long.

Mais pour ce week-end... ça ira. Parce que oui, tout ça, un assembleur modulaire en Common Lisp qui assemble des fichiers (au moins de test) sans erreur

Proposer un contenu

Identifiant

Identifiant

Mot de passe

Mot de passe

☐ Connexion automatique

Se connecter

Pas de compte ? S'inscrire...

sous trois architectures (assez proches, d'accord mais quad même), extensible à d'autres, avec des tests automatisés pour tester l'assembleur lui même d'une part, et les fichiers générés d'autre part face à d'autres compilateurs, ça ne m'aura pris qu'un week-end.

(5 commentaires).

Markdown

EPUB

Bonne idée !

Posté par Enzo Bricolo 🌸🌸🌸 le 16 mars 2026 à 06:17.
Évalué à 2 (+1/-1).

Je vais demander à Claude de faire un bot de tribune en golang qui ne tombe pas en panne, qui soit capable de poster un truc pertinent à minuit pile et qui évite de dire des choses "politiquement incorrectes".



Répondre

questions

Posté par steph1978 le 16 mars 2026 à 08:36. Évalué à 8 (+6/-0).

Quelques questions me brûlent les lèvres doigts



1. combien de tokens pour cette merveille ?

2. où est le code ?

Répondre

show me...

Posté par Gil Cot ✓ (site web personnel, Mastodon) le 16 mars 2026 à 09:04. Évalué à 7 (+5/-0).

Ça semble bien beau tout ça ; on peut regarder le dépôt ?



--

"It is seldom that liberty of any kind is lost all at once." — David Hume

Répondre

[^] # Re: show me...

Posté par Nicolas (site web personnel) le 16 mars 2026 à 09:55. Évalué à 3 (+2/-0).

Et l'historique des interactions avec l'IA. Elle m'a l'air bien flatteuse cette Claude (les LLMs sont connus pour être paramétrés ainsi, et ça explique largement une part du délire autour de ces outils).



Répondre

Et moi ce week-end ...

Posté par Fernando le 16 mars 2026 à 11:48. Évalué à 6 (+5/-0).

j'ai aéré mon composte, bêché, planté des patates (jeudi à la saint Joseph, je peux pas), semé des petits pois, préparé mes semis de tomates et nettoyé mes haies.



Et tout cela à la main (donc sans cramer la planète pour rien) !

Et en plus je suis allé voter.

Pas mal pour un weekend, non ?

Répondre

Envoyer un commentaire

Suivre le flux des commentaires

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

Revenir en haut de page

Derniers commentaires

- Re: Peut-être pas di...
- Et moi ce week-end ...
- Re: C'est bien mais...
- Re: C'est bien mais...
- Classification
- Re: Svp arrêter de di...
- Re: Hmm...
- Re: Le sucre est une ...
- Re: C'est bien mais...
- Re: Hmm...
- Re: J'ai eu un problè...
- Re: réponse

Étiquettes (tags) populaires

- intelligence_artificielle
- merdification
- grands_modèles_de...
- états-unis
- bigtech
- sortie_version
- politique
- capitalisme
- capitalisme_de_surv...
- linux
- jeu_vidéo
- bulle

Sites amis

- April
- Agenda du Libre
- Framasoft
- Éditions D-BookeR
- Éditions Eyrolles
- Éditions Diamond
- Éditions ENI
- La Quadrature du Net
- Lea-Linux
- En Vente Libre
- Grafik Plus
- Open Source Initiative

À propos de LinuxFr.org

- Mentions légales
- Faire un don
- L'équipe de LinuxFr....
- Informations sur le s...
- Aide / Foire aux que...
- Suivi des suggestion...
- Wiki du site
- Règles de modération
- Statistiques
- API pour le développ...
- Code source du site
- Plan du site