

Journal : gb3: mincir sa tribune pour la plage

Posté par devnewton (site web personnel) le 11 juin 2021 à 14:36. Licence CC By-SA.
Étiquettes : coincoin, tribune, golang, gb3

Bonjour Nal,



Si tu suis mes aventures, tu sais que je développe régulièrement des tribunes.

Que ce soit pour tester un [concept](#), enrichir un [produit existant](#) ou tester des technos.

C'était le cas avec [jb3](#), la tribune la plus friturée du marché mais aussi la plus bloated: Java, Spring Boot, Mongodb et maintenant Node-Red parce que c'est rigolo.

J'avais parfois des envies de [minimalisme](#), mais quand c'est trop rikiki, ça manque de sensation.

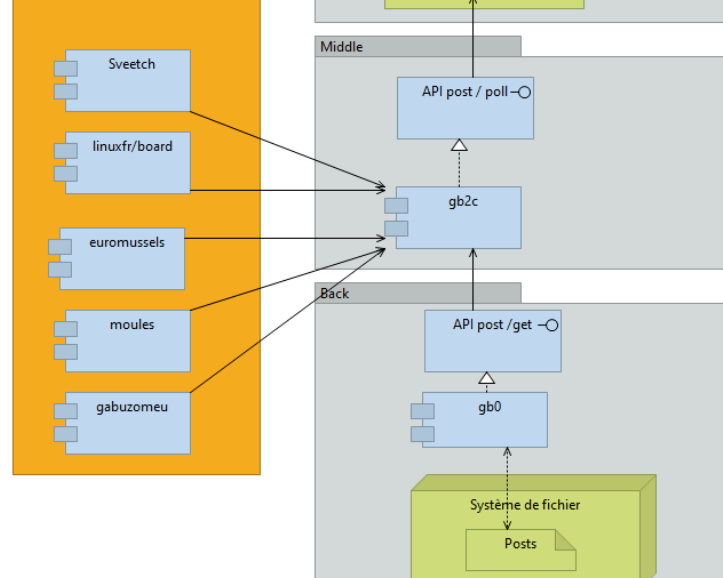
Aujourd'hui je t'écris, car j'ai eu un nouveau prétexte (tester Golang) pour écrire une nouvelle suite logicielle de moulage: [gb3](#).

- La [dém](#).
- [Les sources](#) (licence MIT)

Architecture: TOGAF ready

gb3 se compose de trois composants:

- gc2, un coincoin en vanilla.js ;
- gb0, un bouchot sans IHM en Golang qui utilise le système de fichier comme base de données;
- gb2c, un agrégateur des meilleures tribunes du web en Golang. Il sert de dauphin à gc2.



Cette architecture d'applications à douze facteurs respecte peut être la philosophie Unix, le modèle C4, le design en microservices, ainsi que toutes les bonnes pratiques connues ou inconnues de devoups à ce jour, notamment la couverture de 100% de tests unitaires sur la prod grâce la méthode Production Unit Testing Enhancement (PUTE).

Avec cette méthode, on teste les fonctionnalités une par une en production en attendant que les utilisateurs se plaignent pour corriger les bugs. Le logiciel en production est donc toujours au top sauf quand il ne marche pas.

Fritures: less is less

Ma précédente solution moulesque avait beaucoup de fritures, au prix d'une certaine complexitude. Cette fois j'ai décidé de n'implémenter que des fonctionnalités que j'utilise au quotidien (moulage multi tribune, totoz, emoji, images jointes) et d'abandonner les rigolotes mais peu utiles (bots, trollomètre) et les problématiques pour la privacité des moules (présence en ligne, archivage de longue durée).

J'ai aussi réduit le nombre de paramétrage (on ne peut plus créer de tribune perso, la liste des tribunes est fixe). Bref un processus de gnomisation qui permet d'avoir un logiciel maintenable et sain pour les utilisateurs puisque le développeur sait toujours mieux qu'eux ce dont ils n'ont pas besoin.

Go goal?

Le but de ce projet était d'évaluer Go. Pendant tout le développement je me suis répété *c'est nul à chier ce langage en plus les libs sont pas top* et puis à la fin le résultat est simple à maintenir et fiable.

Et puis ça consomme un poil moins de RAM:

Processus	PID(p)	Name(n)	CPU%(c)	Mem(m)
	10252	gb0	0.0%	15MiB
	10298	gb2c	0.0%	26MiB
	18812	java	0.0%	576MiB
	13012	node-red	0.0%	109MiB

(Ndm: récupérée via webarchive)

Bref je suis mitigé<.

Et si je recommençais en Rust cette fois?

(12 commentaires). [Markdown](#) [EPUB](#)

Go with Rust

Posté par Glandos le 11 juin 2021 à 14:53. Évalué à 5.

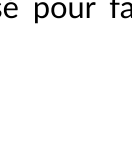
Je plussoie. Attention, il mettre de l'inox A4 en métallique, parce que le milieu marin, ça fait vraiment rouiller.



Pourquoi pas Ada ?

Posté par Ysabeau (site web personnel, Mastodon) le 11 juin 2021 à 17:40. Évalué à 4.

J'ai cru comprendre que la taille des strings dans Ada n'était pas super élastique ce qui t'oblige à garder la ligne. Parfait pour la plage non ? Et là, je sens que tu vas me dire que c'est un truc utilisé pour faire des gros avions et que ça va pas le faire.



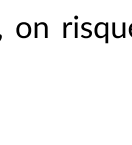
(en fait, tu fais bien ce que tu veux).

Je n'ai aucun avis sur systemd

[^] # Re: Pourquoi pas Ada ?

Posté par devnewton (site web personnel) le 11 juin 2021 à 17:53. Évalué à 10. Dernière modification le 11 juin 2021 à 17:53.

Le problème d'Ada (ou Object Pascal ou Eiffel et plein d'autres), c'est que c'est un bon langage mais pas très vivant sur le marché de l'emploi :(



Alors même avec de bons strings, on risque de se retrouver à poil.

Ce post est offensant ? Prévenez moi sur <https://linuxfr.org/board>

[^] # Re: Pourquoi pas Ada ?

Posté par Ysabeau (site web personnel, Mastodon) le 12 juin 2021 à 13:54. Évalué à 3.

En fait, si j'en crois LinuxFr (en admettant que ça soit représentatif de quelque chose), les langages les plus populaires actuellement seraient C++ et Python avec le petit nouveau Rust qui fait tout pour monter.



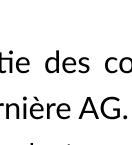
C'est l'impression que cela me donne en tout cas. Elle est peut-être erronée (désolée pour les Adaïstes).

Je n'ai aucun avis sur systemd

[^] # Re: Pourquoi pas Ada ?

Posté par Blackknight (site web personnel, Mastodon) le 12 juin 2021 à 14:39. Évalué à 4.

si j'en crois LinuxFr (en admettant que ça soit représentatif de quelque chose)



C'est un indice comme un autre :)

C'est l'impression que cela me donne en tout cas. Elle est peut-être erronée (désolée pour les Adaïstes).

Même pas grave, ça fait partie des constatations chez Ada-France à la dernière AG.

Le développement en Ada n'est pas aussi rapide et aussi simple que peut l'être un développement en Python notamment. Du coup, c'est moins fun.

En fait, de mon point de vue, c'est surtout après qu'on y gagne, en temps de maintenance, car le refactoring est très guidé par le compil.

Ceci dit, [le dernier index PYPL](#) montre une grosse progression, jamais connue pour le langage sur cet index :)

Peut-être est-ce dû à la meilleure intégration de [Spark Ada](#) dans le langage ou au fait que les plateformes embarquées ont plus de cote en ce moment.

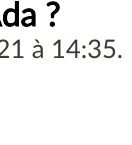
En aparté, s'il y en a que ça intéresse, un développeur a enregistré, à l'occasion de l'Advent of code, deux séances de programmation Ada ([là](#) et [là](#)) où il "découvrirait" le langage. Bon, faut voir 8h à tuer... :D

Je vais mettre ça dans la rubrique *liens*

[^] # Re: Pourquoi pas Ada ?

Posté par Ysabeau (site web personnel, Mastodon) le 14 juin 2021 à 14:27. Évalué à 3.

Ceci dit, le dernier index PYPL montre une grosse progression, jamais connue pour le langage sur cet index :)



Vu qu'il est basé sur le nombre de recherches google de tutoriels sur les langages; on peut peut-être corriger ça avec l'activité informatique des industries qui utilisent ce langage (notamment aéronautique si j'ai bien compris) ? Du genre un gros développement qui pousse les gens à chercher des tutoriels pour l'ancien et le nouveau ?

Je n'ai aucun avis sur systemd

[^] # Re: Pourquoi pas Ada ?

Posté par claudex le 14 juin 2021 à 14:35. Évalué à 4.

Ou une vague de départ à la retraite et du coup, poser personne à qui poser les questions en interne.

[^] # Re: Pourquoi pas Ada ?

Oui, c'est tout à fait possible mais on peut aussi rêver quand on voit arriver quelques nouveaux sur StackOverflow.



Proposer un contenu

Identifiant

Identifiant

Mot de passe

Mot de passe

Connexion automatique

Se connecter

Pas de compte ? S'inscrire...

on peut peut-être corrélér ça avec l'activité informatique des industries qui utilisent ce langage (notamment aéronautique si j'ai bien compris) ?



Pour avoir bossé quelques années dans le secteur, je n'y ai vu que trois ou quatre projets en Ada. En même temps, c'est chiant, les trucs fonctionnent pendant des lustres, difficile de vendre de la maintenance :D

Du genre un gros développement qui pousse les gens à chercher des tutoriels pour l'ancien et le nouveau ?

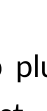
Alors dans ce cas, on peut se trouver dans le cas de portage vers d'autres langages mais le bon indicateur, c'est le nombre d'offres d'emploi et de ce côté-là, c'est pas forcément le feu d'artifice.

Ceci dit, apprendre Ada est largement à la portée du développeur C/C++ ou Java donc au final, c'est pas bien compliqué de recycler un dev... Mais ça, je ne crois pas que grand monde le fasse.

[^] # Re: Pourquoi pas Ada ?

Posté par devnewton (site web personnel) le 12 juin 2021 à 14:59. Évalué à 8.

Il faut distinguer la popularité (les devs aiment bien) et la popularité (les projets sont codés avec).



En entreprise, on voit beaucoup plus de Java et de Go que du Python et du Rust.

Il y aussi des langages impopulaires ou peu à la mode mais très très très utilisés (PHP, Cobol).

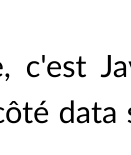
--

Ce post est offensant ? Prévenez moi sur <https://linuxfr.org/board>

[^] # Re: Pourquoi pas Ada ?

Posté par Blackknight (site web personnel, Mastodon) le 12 juin 2021 à 19:03. Évalué à 2. Dernière modification le 12 juin 2021 à 19:03.

En entreprise, on voit beaucoup plus de Java et de Go que du Python et du Rust.



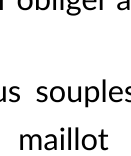
Ca dépend, dans ma boîte, c'est Java, C#, PHP et du Python surtout côté data science .

On a un seul projet en Go mais parce que c'est un ajout de fonctionnalités à Mattermost et effectivement, pas de Rust.

[^] # Re: Pourquoi pas Ada ?

Posté par Blackknight (site web personnel, Mastodon) le 12 juin 2021 à 11:50. Évalué à 5.

Ca peut le faire, en Ada, y en a pour tous les goûts.



La preuve, les *Strings* sont au nombre de trois:

- les *Strings* de taille fixe. Idéal pour obliger à garder la ligne pour la page

- les *bounded strings*, un peu plus souples qu'on pourrait assimiler au surfer (le maillot pas le contenu)

- les *unbounded strings* totalement libres, une sorte de naturisme de la chaîne de caractères

En plus, ça existe aussi en [grande taille](#) et [très grande taille](#) :)

Finalement, c'est de la string qui ne te lâche jamais

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: Lapin compris
- Flutter
- Re: Et puisqu'il y a u...
- Re: BTTF
- Re: Lapin compris
- Re: ... pour préserve...
- Re: ... pour préserve...
- Re: Unified Attestati...
- Re: BTTF
- Re: Papier, tempora...
- Re: Haiku pour Rasp...
- Re: Et puisqu'il y a u...

Étiquettes (tags) populaires

- intelligence_artificielle
- le_monde_diplomati...
- grands_modèles_de...
- merdification
- états-unis
- capitalisme
- datacenter
- administration_fran...
- souveraineté_numer...
- cybersécurité
- logiciel_libre
- google

Sites amis

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

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 dévelop...
- Code source du site
- Plan du site