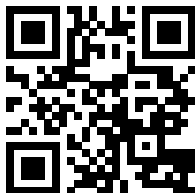


# Pourquoi et comment se lancer dans le libre ?

Matthieu Moy

26 octobre 2022

<https://matthieu-moy.fr/spip/IMG/pdf/talk-libre.pdf>



# Plan

- 1 C'est quoi le « libre » ?
- 2 Pourquoi contribuer ?
- 3 Comment s'y mettre ?
- 4 Se mettre le pied à l'étrier
- 5 Exemples personnels
- 6 Conclusion

# Plan

- 1 C'est quoi le « libre » ?
- 2 Pourquoi contribuer ?
- 3 Comment s'y mettre ?
- 4 Se mettre le pied à l'étrier
- 5 Exemples personnels
- 6 Conclusion

# Libre Vs Open Source



- Liberté de l'utilisateur
- Éthique



open source  
initiative

- Excellence technique
- Business model

# Moi et le libre

- Utilisateur de GNU/Linux depuis 20 ans
- Contributeur occasionnel de plusieurs logiciels
- Mainteneur de quelques petits logiciels
- Google Summer of Code mentor
- Projets étudiants « contribuer à un logiciel libre » (LIFPROJET et POM à Lyon1)

# Plan

- 1 C'est quoi le « libre » ?
- 2 Pourquoi contribuer ?**
- 3 Comment s'y mettre ?
- 4 Se mettre le pied à l'étrier
- 5 Exemples personnels
- 6 Conclusion

# Parce que le monde se porte mieux avec le logiciel libre

- Partage, communauté, non-commercial
- Écosystème d'entreprises  $\neq$  monopole  
*According to The Linux Foundation, since 2005, "some 14,000 individual developers from over 1,300 different companies have contributed to the kernel."*<sup>1</sup>
- Écosystème de logiciels sur lequel construire  
Exemple : internet et le web reposent largement sur le libre.

↪ C'est sympa de poser une pierre à l'édifice !

---

1. <https://thenewstack.io/contributes-linux-kernel/>

# Parce que c'est fun !

- Projet perso privé : je code, je suis content (= moi avant internet)
- Projet libre : je code, je rencontre des gens, on argumente, des gens sont contents, d'autres pas, ...



## Pour flatter son égo et briller en société

From: Linus Torvalds <torvalds@linux-foundation.org>

To: Matthieu Moy <Matthieu.Moy@grenoble-inp.fr>

[...]

Subject: Re: Fwd: [Survey] Signed push

Date: Wed, 14 Sep 2011 08:27:35 -0700

On Wed, Sep 14, 2011 at 4:03 AM, Matthieu Moy

<Matthieu.Moy@grenoble-inp.fr> wrote:

>

> Would the notes mechanism be able to annotate ref names i  
> of commit sha1?

That would be a horrible, horrible notion.

[...]

So no, don't confuse branch *\*contents\** with branch *\*descrip*

Linus

## Pour flatter son égo et briller en société

```
From: Linus Torvalds <torvalds@linux-foundation.org>  
Date: Fri, 17 Feb 2017 09:38:17 -0800  
To: Matthieu Moy <Matthieu.Moy@grenoble-inp.fr>
```

[...]

```
So this notion that the bottom of the commit message is some  
header crap is WRONG.
```

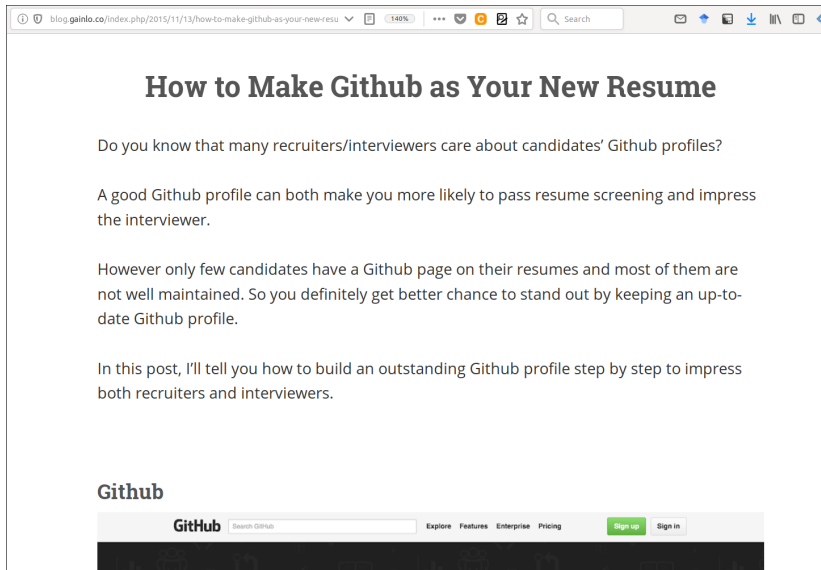
```
Stop it. It caused bugs. It's wrong. Don't do it.
```

Linus

# Pour flatter son égo et briller en société



# Pour trouver du boulot



The screenshot shows a web browser window with the address bar containing the URL `blog.gainlo.co/index.php/2015/11/13/how-to-make-github-as-your-new-resu`. The page title is "How to Make Github as Your New Resume". The main content includes the following text:

Do you know that many recruiters/interviewers care about candidates' Github profiles?

A good Github profile can both make you more likely to pass resume screening and impress the interviewer.

However only few candidates have a Github page on their resumes and most of them are not well maintained. So you definitely get better chance to stand out by keeping an up-to-date Github profile.

In this post, I'll tell you how to build an outstanding Github profile step by step to impress both recruiters and interviewers.

**GitHub**

Below the text, there is a partial view of the GitHub website header, showing the GitHub logo, a search bar, and navigation links for "Explore", "Features", "Enterprise", and "Pricing". There are also "Sign up" and "Sign in" buttons.

## Pour trouver du boulot

- Exemple : moi, contributeur modeste (286 commits dans Git par exemple), pas à la recherche d'un emploi :
  - ▶ Démarché par Facebook pour travailler sur leur infrastructure de gestion de versions,
  - ▶ Démarché par Google, plusieurs fois<sup>2</sup>.

---

2. Dont une fois qui était un copain qui me faisait une blague, mais pas toutes

# Pour apprendre des choses !

- Écrire du code utilisé par des vrais gens  $\Rightarrow$  se raccrocher à la vraie vie ( $\neq$  TP scolaire,  $\neq$  prototype de recherche jamais utilisé)
- Participer à une équipe de développement avec des gens *très* compétents.
- Revue de code, discussion sur le code  $\Rightarrow$  on apprend de ses erreurs et des compétences des autres.

# Plan

- 1 C'est quoi le « libre » ?
- 2 Pourquoi contribuer ?
- 3 **Comment s'y mettre ?**
- 4 Se mettre le pied à l'étrier
- 5 Exemples personnels
- 6 Conclusion

# Scratch your own itch

« grattez là où ça vous démange »



# Commencer petit

# Comprendre la communauté

- Communication : mailing-list ? messagerie instantanée ? issues GitHub, Lab, ... ?
- Contributions de code : demander avant ? Envoyer un patch direct ?
- Revue de code : comment ça marche ? Quel niveau d'exigence ?
- Regarder le fichier `CONTRIBUTING.md` dans le code s'il existe.

# Il n'y a pas que les développeurs qui sont importants

- Écrire du nouveau code = augmenter la dette technique (« les autres » devront relire le code, maintenir et corriger les bugs, ...)
- Relire du code, corriger des bugs, tester = réduire la dette technique.
- Sans compétences en informatique : traductions, documentation, animations de forums et mailing lists utilisateurs.
- Feedback constructif en temps qu'utilisateur (rapports de bugs)

# Faire un bon rapport de bug

- Pas glop :
  - ▶ J'ai essayé d'utiliser le logiciel et ça bugge, merci de corriger.

# Faire un bon rapport de bug

- Pas glop :
  - ▶ J'ai essayé d'utiliser le logiciel et ça bugge, merci de corriger.
- Glop :
  - ▶ Steps to reproduce :
    - 1 ...
    - 2 ...
    - 3 ...
  - ▶ Expected behavior :  
...
  - ▶ Actual behavior :  
...

# Faire un bon rapport de bug

- Pas glop :
  - ▶ J'ai essayé d'utiliser le logiciel et ça bugge, merci de corriger.
- Glop :
  - ▶ Steps to reproduce :
    - 1 ...
    - 2 ...
    - 3 ...
  - ▶ Expected behavior :  
...
  - ▶ Actual behavior :  
...
- Glop glop :
  - ▶ GDB backtrace attached
  - ▶ It bisects down to commit 37f61b7 (blablabla, 2011-11-07), CC-ing the author.
  - ▶ Tested on master branch, the bug is still there
  - ▶ Patch attached

# Faire un bon rapport de bug

## ● Conseils :

- ▶ Rester factuel ( $\neq$  c'est un scandale que ça ne marche pas)
- ▶ Éviter les comparaisons si elles ne sont pas constructives (« le logiciel alternatif XYZ fait ça, on pourrait s'en inspirer »  $\neq$  « même le logiciel XYZ le fait, et pas vous !?! »)
- ▶ Remercier les contributeurs
- ▶ Être prêt moralement à envoyer un message dans un trou noir (vous n'avez pas payé, la communauté ne vous doit rien).
- ▶ Autant que possible, compilez la dernière version vous-mêmes pour vérifier si le bug est toujours présent (et plus tard confirmer si la correction marche).

# Participer à la vie du logiciel libre

- Adhérer/s'investir dans des associations

- ▶ La Quadrature du Net | Internet & Libertés.

<https://www.laquadrature.net/>

- ▶ April : Promouvoir et défendre le logiciel libre.

<https://www.april.org/>

- ▶ Framasoft - Changer le monde, un octet à la fois.

<https://framasoftware.org/>

Basée à Lyon !



# Plan

- 1 C'est quoi le « libre » ?
- 2 Pourquoi contribuer ?
- 3 Comment s'y mettre ?
- 4 Se mettre le pied à l'étrier**
- 5 Exemples personnels
- 6 Conclusion

# Un cadre pour contribuer

- Google Summer of Code (GSoC)  
(<https://summerofcode.withgoogle.com/>)
  - ▶ 2 mois en télétravail payé par Google
  - ▶ 1 ou 2 mentor identifié dans la communauté
  - ▶ Validable comme un stage ? (pas partout)
- Outreachy (<https://www.outreachy.org/>)
  - ▶ ≈ GSoC (mais c'est pas Google qui paye)
  - ▶ Réservé aux « minorités » (femmes, trans, noirs américains, ...)
- Projets scolaires
  - ▶ Si vous m'avez comme prof au moins ... Vous faites LIFPROJET au printemps ?
- Stage
  - ▶ exemple : <https://careers.mozilla.org/university/>

# Trouver comment commencer petit

- Idéal : trouver un truc qui vous embête et qui semble facile à corriger (exemple : typo dans un message visible par l'utilisateur)
- Beaucoup de projets maintiennent une liste de bugs pour débutants :

- ▶ « Good first bugs » chez Mozilla (Firefox) :

`https://developer.mozilla.org/en-US/docs/Mozilla/Developer\_guide/Introduction#Find\_a\_bug\_we've\_identified\_as\_a\_good\_fit\_for\_new\_contributors.`

- ▶ `Git #leftoverbits`

`https://git.github.io/SoC-2019-Microprojects/`

- ▶ ...

# Plan

- 1 C'est quoi le « libre » ?
- 2 Pourquoi contribuer ?
- 3 Comment s'y mettre ?
- 4 Se mettre le pied à l'étrier
- 5 Exemples personnels**
- 6 Conclusion

# SchemeWeb2

- SchemeWeb utilisé pour un projet d'école, mais 1) ne marchait pas sur ma machine, 2) manquait de quelques fonctionnalités faciles à coder.
- Modification, discussion avec l'auteur, publication sous le nouveau nom SchemeWeb2
- « Wow » quand mon code est apparu sur ctan.org :-)

# Emacs / Gnus

- Fonctionnalité triviale dans le mailer inclu dans Emacs ( $\approx$  30 LOC de lisp)
- Signature des papiers officiels pour contribuer au projet GNU
- « Wow, la free software foundation m'envoie un courrier papier, la classe. La prochaine étape c'est d'inviter Stallman au restau ? »

# Emacs / Gnus

- Fonctionnalité triviale dans le mailer inclu dans Emacs ( $\approx$  30 LOC de lisp)
- Signature des papiers officiels pour contribuer au projet GNU
- « Wow, la free software foundation m'envoie un courrier papier, la classe. La prochaine étape c'est d'inviter Stallman au restau ? » (c'est arrivé 4 ans plus tard en vrai !)

# GNU Arch

- Ancêtre/oncle éloigné de Git
- Contribution à l'outil
- Mode Emacs (Xtla) :
  - ▶ Contributeur puis mainteneur.
  - ▶ Bons retours d'utilisateurs ( $\approx$  « GNU Arch est incompréhensible, mais avec Xtla c'est agréable à utiliser »)
  - ▶ Package Debian



# Git

- GNU Arch est mort, je contribue à quoi maintenant ?
- Recette du succès :
  - 1 Faire faire un TP de Git à des étudiants
  - 2 En avoir marre de répondre 5 fois à la même question en 10 minutes parce qu'un message d'erreur n'est pas clair
  - 3 Envoyer un patch
  - 4 `goto 1`
- Coaching d'étudiants (projets scolaires, mentor Google summer of code)

# git-multimail

- Contribution pour mon utilisation perso
- mainteneur → moi : « tu veux pas me remplacer, j'ai plus la motiv', j'utilise plus l'outil de toutes façons ? »

## git-multimail

- Contribution pour mon utilisation perso
- mainteneur → moi : « tu veux pas me remplacer, j'ai plus la motiv', j'utilise plus l'outil de toutes façons ? »
- moi → mainteneur : « euh ... non, j'ai pas le temps non plus moi?!? »

## git-multimail

- Contribution pour mon utilisation perso
- mainteneur → moi : « tu veux pas me remplacer, j'ai plus la motiv', j'utilise plus l'outil de toutes façons ? »
- moi → mainteneur : « euh ... non, j'ai pas le temps non plus moi?!? »
- mainteneur → moi : « OK, je prends ça pour un 'oui' »

# git-multimail

- Contribution pour mon utilisation perso
- mainteneur → moi : « tu veux pas me remplacer, j'ai plus la motiv', j'utilise plus l'outil de toutes façons ? »
- moi → mainteneur : « euh ... non, j'ai pas le temps non plus moi?!? »
- mainteneur → moi : « OK, je prends ça pour un 'oui' »
- Quelques mois plus tard :

```
~/dev/git-multimail$ git shortlog -s | sort -nr | head -5
 480  Matthieu Moy
 304  Michael Haggerty
  41  Richard Hansen
  16  Elijah Newren
   7  Edward d'Auvergne
```

## chamilotools et mechanicalsoup

- J'aime pas cliquer dans une interface web
- Mon université utilisait Chamilo (= clicodrome web)
- Automatisation des actions sur Chamilo avec « chamilotools »
- Besoin d'une lib Python pour simuler un navigateur web
- MechanicalSoup intéressant mais plus vraiment maintenu, manque des fonctionnalités
- ⇒ reprise en main du projet, rejoint par un autre contributeur
- Super terrain d'expérimentation (460 Lignes de Python + 770 lignes de tests), projet le plus « irréprochable » possible, utilisation d'outils modernes (GitHub, Travis, ReadTheDocs, codecov.io, lgtm.com, ...).

# darktable

- Article chaque année sur les nouveautés de la dernière version, très appréciés (c'est la saison si vous voulez aider ...)
- Participation sur la mailing-list utilisateur et le forum francophone.
- 29 bug reports (18 « fixed »)
- Beaucoup de corrections de bugs mineurs
- Améliorations d'ergonomie mineures
- 1 fonctionnalité triviale (curseur « biais d'exposition » dans le module « courbe de base »)

# Plan

- 1 C'est quoi le « libre » ?
- 2 Pourquoi contribuer ?
- 3 Comment s'y mettre ?
- 4 Se mettre le pied à l'étrier
- 5 Exemples personnels
- 6 Conclusion



## Contribuez à des logiciels libres ...

- Vous deviendrez (encore) plus intelligents
- Vous aurez l'impression de vous rendre utiles
- Vous rencontrerez des gens incroyables
- Vous deviendrez riches et célèbres (peut-être)

<https://matthieu-moy.fr/spip/IMG/pdf/talk-libre.pdf>

