Le logiciel libre à l'université

Violaine Louvet

Journée OPUS, 19 septembre 2023

Algorithme, code source, logiciel, de quoi parle-t-on ?

Petit historique

- *Software* utilisé dès 1953 pour désigner la partie immatérielle de l'ordinateur par rapport au *hardware*, partie matérielle.
- Le mot *logiciel* apparaît en 1969 (à partir des mots *logique* et *matériel*).

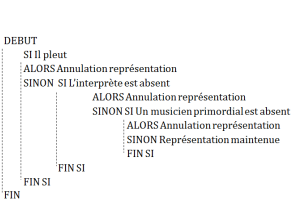
- Algorithme : décrit le déroulé pour la résolution d'un problème posé.
- Code source : mise en oeuvre et formalisation de l'algorithme dans un langage informatique (par exemple python, C++, java ...). C'est un (ou plusieurs) fichier(s) texte.
- Exécutable : traduction du code source (en général via un compilateur ou un interpréteur) en code binaire compréhensible par l'ordinateur.
- Logiciel : en général, l'ensemble global comprenant le code source et/ou l'exécutable, et le plus souvent la documentation, des exemples d'utilisation, éventuellement les dépendances ... et évidemment la licence associée.

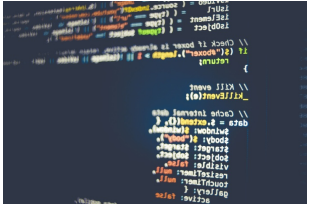
Ce qu'il faut retenir

- « De manière simplifiée, l'algorithme est une recette de cuisine, et le code source sa réalisation concrète »(cf rapport Bothorel)

- L'« âme »d'un logiciel est dans son code source, c'est-à-dire dans les instructions telles qu'elles sont rédigées pour être lisibles par l'humain.

- Seul le code source permet d'accéder aux informations techniques et scientifiques





Petite histoire de l'informatique et du logiciel

- Jusque dans les années 1970, seules certaines entreprises avaient les moyens financiers de s'acheter des ordinateurs et donc les logiciels associés.
  - $\longrightarrow$ Ces entreprises avaient un intérêt particulier à laisser les chercheurs et développeurs disposer de ces logiciels notamment pour les étudier et les modifier.
    - A cette période des débuts de l'informatique, le partage de code source entre chercheurs était une pratique universellement admise
    - les programmes informatiques sont alors traités, dans les universités concernées, de la même manière que n'importe quelle information scientifique : mis à la disposition de tout un chacun pour étude, exploitation et amélioration
- Progressivement, avec le développement des ordinateurs personnels, l'importance économique des logiciels apparaît.
  - Bill Gates (création de Microsoft en 1975) publie en 1976 An Open Letter to Hobbiyst dans laquelle il dénonce la faible protection des créations des développeurs.
  - En Octobre 1976, le Copyright Act est adopté aux Etats-Unis, protégeant les logiciels par le droit d'auteur américain.
  - $\longrightarrow$ C'est la naissance des logiciels propriétaires.

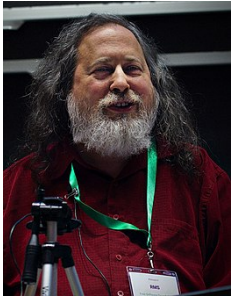
Philosophie libre

- L'échange libre de logiciels a donc existé depuis les débuts de l'informatique, sans qu'on ait défini les principes de ces partages.
- En particulier dans la communauté des hackers (« bidouilleurs »), désignant originellement ces jeunes étudiants en informatique du MIT, génies de la programmation

  $\longrightarrow$ Un des principes de l'« éthique hacker » est que toute information est par nature libre

Richard Stallman

Informaticien au MIT, membre de la communauté des hackers, frustré de ne pas pouvoir modifier le logiciel qui contrôlait l'imprimante de son labo, il est alors convaincu de la nécessité pour les gens de pouvoir librement modifier le logiciel qu'ils utilisent.



Naissance du logiciel libre

- Richard Stallman fonde la Free Software Foundation (FSF) en 1985.
- Cet organisme formalise les 4 grands principes du libre dans des licences, textes juridiques qui fixent les conditions pour exploiter un logiciel ou une oeuvre intellectuelle.
- La FSF soutient le développement du projet GNU depuis le début.
  - $\longrightarrow$ Le projet GNU concerne le développement d'un système d'exploitation complètement libre.
  - $\longrightarrow$ Sa version fonctionnelle s'appuie sur le noyau linux, développé indépendamment par Linus Torvalds

Les 4 libertés fondamentales du logiciel libre

- possibilité d'utiliser l'oeuvre, pour tous les usages ;
- possibilité d'étudier l'oeuvre ;
- possibilité de redistribuer des copies de l'oeuvre ;
- possibilité de modifier l'oeuvre et de publier ses modifications.

Logiciel libre $\neq$ logiciel gratuit

Et maintenant ?

- Le logiciel libre est né dans la pratique universitaire, le partage des connaissances est au coeur de la recherche. Une grande majorité des logiciels développés dans la recherche le sont sous licence libre (cf rapport à venir MESR)
- De grandes entreprises du numérique (comme IBM, Cisco, Intel, Samsung, etc.), mais aussi les GAFAM (Google, Amazon, Facebook, Apple, Microsoft) comptent parmi les plus importants contributeurs au logiciel libre.
  - $\longrightarrow$ Même Netflix, qui consomme à lui seul près d'un quart de la bande passante d'Internet en France, utilise des logiciels libres et contribue à leur développement.
- Les logiciels libres et open source se retrouvent aujourd'hui partout autour de nous, et cela bien souvent de manière non visible : les serveurs, les sites web, les objets intelligents, les services dans le cloud, reposent en grande majorité sur des logiciels et des systèmes d'exploitation libres.

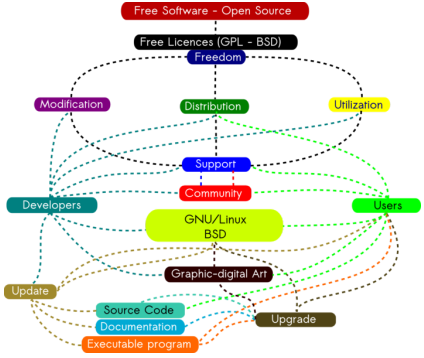
Logiciel libre vs open source

- Principale différence se situe dans les valeurs que chacun porte
  - Le mouvement du *libre* est un mouvement social qui soutient des valeurs philosophiques et politiques.
  - L'*open source* est plus récent et met en avant la méthodologie de développement et de diffusion du logiciel.

$\longrightarrow$ Le logiciel libre est forcément open source. Un logiciel peut être open source sans être libre au sens de la FSF.

Rôle des communautés du logiciel libre

- La communauté d'un logiciel libre désigne le groupe de personnes qui collaborent pour développer, maintenir et promouvoir un logiciel libre.
- Cette communauté est généralement composées de développeurs, de contributeurs et d'utilisateurs.
- Cette action collective assure le partage des connaissances, la transparence des développements, et la flexibilité des contributions.
- Ces contributions peuvent prendre des formes multiples.



Contribution informatique au logiciel libre

- Elles peuvent prendre différentes formes :
  - ⟶ Ajout d'une fonctionnalité dont vous aviez besoin, que vous avez développé pour votre usage et que vous proposez à la communauté.
  - ⟶ Identification et correction d'un bug.
  - ⟶ Vérification de sécurité (audit de code) ...

- Mais aussi d'une façon plus indirecte, apports scientifiques sur les algorithmes implémentés par exemple.

L'exemple de Scikit Learn

`https://scikit-learn.org/stable/developers/contributing.html#`

Contribution à la documentation du logiciel libre

Documenter

- La documentation est essentielle pour tout logiciel.
- C'est une activité totalement différente du développement et qui nécessite plutôt des compétences métier.

Traduire

- Pour favoriser une appropriation par le plus grand nombre, il est souvent important de proposer la documentation ou l'interface dans différentes langues.
- A nouveau, les compétences en jeu ne sont pas techniques.

Aider à améliorer l'ergonomie

- Echanger autour de propositions sur une utilisation plus facile, des modifications au niveau des interfaces utilisateurs ... qui aideront à faire évoluer le logiciel.

Rapporter les bugs

- Le rapport de bug est l'une des contributions les plus courantes.
- Certains dysfonctionnements n'arrivent que dans des cas de figure précis, et l'aide des utilisateurs est essentielle.
- Repérer, communiquer et documenter les bugs rencontrés de la manière la plus précise possible avec le maximum d'informations.

Forge logicielle

La forge logicielle est un outil indispensable pour tout développement de logiciel. En particulier, elle intègre pas mal de ressources permettant de favoriser les diverses contributions de la communauté :

- Listes de diffusion ;
- Outil de suivi de bugs ;
- Gestionnaire de documentation ...

Contribuer au financement

- Le soutien financier aux logiciels libres est essentiel à leur survie.

- Il peut se faire selon différents modèles :
  - ⟶ financement direct par des subventions / adhésions selon la structure juridique (fondation, entreprise ...) qui coordonne les développements
  - ⟶ achat de services associés (formation, installation, configuration ...)
  - ⟶ souscription à un service de type SaaS Software as a Service) ...

L'exemple d'ElabFTW

- Logiciel de cahier de laboratoire électronique
- Développé à l'Institut Curie
- Sous licence AGPL (équivalent GPL pour les services Web)
- Modèle économique basé sur les services : support, hébergement, formation, développement personnalisé, ...
- https://doc.elabftw.net/contributing.html

Pourquoi utiliser du logiciel libre : indépendance

L'utilisation d'un logiciel non libre à l'école

Pourquoi certains éditeurs de logiciels privateurs offrent-ils des exemplaires gratuits de leurs programmes non libres aux écoles ?
Pour implanter la dépendance à leurs produits : ces éditeurs ne fourniront évidemment pas d'exemplaire gratuit à ces élèves et étudiants une fois qu'ils auront leur diplôme, ni aux entreprises pour lesquelles ils travailleront.
Une fois dépendant, il vous faudra payer et les mises à jour futures pourront se révéler onéreuses.

- L'ouverture du code source garantit une indépendance, un contrôle sur le logiciel et la pérennisation des accès au code.
- L'utilisation d'un logiciel propriétaire rend captif : comment récupérer ses données ? Comment migrer vers un autre logiciel Que se passe-t-il si l'éditeur disparaît ?

Souveraineté et interopérabilité

- « Logiciel privateur » == « logiciel propriétaire », est plus parlant (privation de libertés) :
  - opacité quasi totale du fonctionnement interne,
  - impossibilité d'adapter aux besoins ou de corriger les erreurs,
  - perte possible des données dû à l'utilisation de formats propriétaire fermés.
- Les logiciels libres et open source présentent des garanties de réversibilité bien plus importantes
  - Souvent conçus avec une architecture ouverte et standardisée
  - Respect des standards ouverts (fichiers de données) et donc interopérabilité

Le problème des données fermées

« Les analyses par spectrométrie de masse produisent de grandes quantités de données, directement en sortie de l'instrument. Chaque fabricant a son propre format de données, qui est le plus souvent propriétaire. Ordinairement, les données aux formats propriétaires sont traitées par les logiciels fournis avec l'instrument, eux aussi propriétaires. Comme souvent dans le domaine des formats de fichiers, ces formats ne sont pas pérennes et les licences d'utilisation des logiciels propriétaires sont très coûteuses. Il y a donc ici un problème majeur d'interopérabilité. »

« Logiciels libres : à la recherche du bien commun », Edlira Nano, Olivier Langella, Filippo Rusconi, JRES 2021

La question de la qualité

- L'accès au code source permet l'étude, la critique ou la correction des programmes.
- Plus la communauté est importante, plus la qualité et la stabilité du logiciel sont assurées :
  - Par le nombre d'utilisateurs qui testent, vérifient dans de nombreuses situations et font remonter les problèmes.
  - Par le nombre de développeurs ou de contributeurs qui font bénéficier d'un large panel de connaissances techniques et de savoir-faire.
  - Un nombre de développeurs important nécessite la mise en place de bonnes pratiques de développement, de règles de programmation, d'audit automatique ... qui sont autant de garantie de qualité des programmes.

L'exemple des contributions à Omeka S

Logiciel libre de plateforme de publication web pour relier les collections du patrimoine culturel, les corpus scientifiques, etc. https://omeka.org/s/docs/developer/contributing/

- Utilisation du guide de codage PSR-2 (standards de programmation pour le langage PHP)
- Utilisation du modèle gitflow (pour le versionnage)
- Auto-documentation, commentaires ....

La sécurité dans les logiciels libres

- La montée en puissance des logiciels libres en fait des cibles pour les pirates.
  - Ceci est facilité par l'accès au code.
- Le dynamisme des communautés open source, leur mode de fonctionnement permet :
  - de détecter les failles rapidement,
  - de diffuser les informations très vite,
  - d'apporter les correctifs.
- L'ouverture du code permet également de s'assurer que le logiciel ne collecte pas ou ne partage pas de données personnelles sans autorisation.
  - Si on a les compétences pour le vérifier !!
  - A nouveau, le dynamisme et l'ouverture de la communauté permet de détecter très vite toutes les anomalies.

L'utilisation d'un logiciel libre est-elle rentable ?

- Il n'y a pas de coût de licence.
  - Les clients de logiciels propriétaires doivent renouveler leurs licences, racheter de nouvelles versions de leurs applications.
  - Il peut également y avoir des problématiques d'obsolescence programmée.

- Les modèles économiques libres reposent la plupart du temps sur les services qui peuvent être associés.

Libre n'est pas gratuit !

Il ne faut pas négliger les coûts liés à l'installation, la configuration, le déploiement, la prise en main, la migration, la formation ...
qui existent aussi pour les logiciels propriétaires

Logiciel libre et intégrité de la recherche

- Le logiciel libre est avant tout une philosophie du partage des communs, de la connaissance.

- Cette philosophie est naturellement née dans le milieu universitaire.
  - Le mouvement de la science ouverte, en particulier autour des données, ne fait finalement que reproduire ce qui existe dans le monde du logiciel depuis le début.

- Il est difficile d'imaginer produire un résultat scientifique à partir d'une boîte noire dont on ne sait pas ce qu'elle fait exactement.
  - C'est pourtant régulièrement le cas des articles scientifiques basés sur l'utilisation de logiciels propriétaires.

- La question de la reproductibilité est au coeur des problématiques actuelles.
  - Elle n'est possible que si les logiciels utilisés dans les publications sont libres.

Un logiciel reste un logiciel !

Pas de situation idyllique !

Tout logiciel reste :

- un programme avec ses bugs,
- qu'il faut prendre en main,
- pas toujours intuitif / facile d'utilisation,
- qui est très dépendant du contexte : système d'exploitation, matériel, ..., contexte très évolutif

- Un logiciel libre non mature, avec peu de développeurs et / ou une petite communauté, n'est pas forcément à mettre entre toutes les mains
- Si on veut vraiment contribuer, pour l'adapter à ses besoins, il faut avoir des compétences techniques (mais au moins on a la liberté de le faire !)
- Quoi qu'il en soit, les évolutions technologiques font qu'il est difficile de se projeter au-delà de 5 à 10 ans !
  $\longrightarrow$ Il faut anticiper dès le départ les changements importants (montée de version majeure, disparition ou redéveloppement complet ...)

Logiciel de recherche et science ouverte

- Au même titre que les données et les publications, les logiciels développés dans la recherche s'inscrivent pleinement dans la science ouverte

⟶ Des enjeux importants autour du référencement et de la description des logiciels (et des bonnes pratiques de développement) :

- Archivage sur Software Heritage pour pérenniser les codes sources
- Notice sur HAL pour description, métadonnées, et citation
- Un lien existant entre les deux depuis quelques mois

- Des évolutions à envisager dans les services d'accompagnement technique et documentaire pour prendre en compte cet objet scientifique.

Conclusions

- Le logiciel libre est depuis toujours associé au monde de la recherche académique
- La majorité des logiciels développés dans les laboratoires sont des logiciels libres
- Utiliser du logiciel libre, c'est s'inscrire dans un mode de pensée autour des communs numériques co-construits auxquels chacun peut apporter quelque chose
  $\longrightarrow$ contribuer, partager et échanger à travers une communauté
- Cela reste du logiciel
  $\longrightarrow$ avec des problématiques techniques, d'utilisation, ... qu'il soit libre ou propriétaire !