

# Conception d'un système de "tagging" des fichiers avec Rust

Steven Liatti

Projet de bachelor - Prof. Florent Glück

Hepia ITI 3<sup>ème</sup> année

11 mai 2018

**Résumé** Le but de ce projet est de concevoir et développer un "moteur de gestion de tags" pouvant gérer des dizaines de milliers de fichiers et tags associés de manière efficace, en Rust. Le stockage des tags utilisera le mécanisme des "extended attributes" disponibles dans la plupart des systèmes de fichiers modernes. Le moteur d'indexation devra surveiller les fichiers modifiés, créés, ou supprimés afin d'indexer les tags avec un minimum de latence (temps réel). Si le temps le permet, le système développé sera intégré à un environnement desktop choisi (Gnome, KDE, etc.).

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Motivations . . . . .	3
1.2	Buts . . . . .	3
<b>2</b>	<b>Analyse de l'existant</b>	<b>3</b>
2.1	Fonctionnalités incluses dans le système d'exploitation . . . . .	3
2.1.1	macOS . . . . .	3
2.1.2	Windows . . . . .	5
2.2	Applications utilisateur . . . . .	5
2.2.1	TagSpaces . . . . .	5
2.2.2	Tmsu . . . . .	5
2.2.3	Tagsistant . . . . .	5
2.2.4	TaggedFrog . . . . .	5
2.2.5	Tabbles . . . . .	5
2.2.6	Dropbox . . . . .	5
<b>3</b>	<b>Analyse des besoins</b>	<b>5</b>
<b>4</b>	<b>Technologies</b>	<b>5</b>
4.1	Rust . . . . .	5
4.2	Les extended attributes . . . . .	5
4.2.1	Théorie . . . . .	5
4.2.2	Petites manipulations . . . . .	6
4.3	Notifications . . . . .	6
<b>5</b>	<b>Architecture</b>	<b>6</b>
<b>6</b>	<b>Réalisation</b>	<b>6</b>
<b>7</b>	<b>Discussion</b>	<b>6</b>
<b>8</b>	<b>Conclusion</b>	<b>6</b>
8.1	Remerciements . . . . .	6
<b>9</b>	<b>Références</b>	<b>7</b>

# Table des figures

1	Vue et gestion d'un tag dans le Finder macOS ([1]) . . . . .	4
---	--	---

# Table des listings de code source

1	mdls listant les tags d'un fichier sous macOS ([2]) . . . . .	4
---	---	---

# 1 Introduction

[3] [4] [5] [6] [7] [1] [8] [2] [9] [10] [11] [12]

## 1.1 Motivations

## 1.2 Buts

# 2 Analyse de l'existant

Dans cette section, nous allons analyser les principales solutions existantes, qu'elles soient intégrées directement dans un système d'exploitation ou sous la forme d'applications utilisateur.

## 2.1 Fonctionnalités incluses dans le système d'exploitation

### 2.1.1 macOS

macOS possède son propre système pour étiqueter des fichiers. Il est intégré depuis la version OS X 10.9 Mavericks. Depuis l'explorateur de fichiers, l'utilisateur a la possibilité d'ajouter, modifier, supprimer, rechercher des tags. Les fichiers peuvent avoir plusieurs tags associés. Un code couleur permet de plus facilement se souvenir et visualiser les tags attribués. Dans l'explorateur de fichiers, les tags se retrouvent sur le bas côté, pour y accéder plus rapidement.

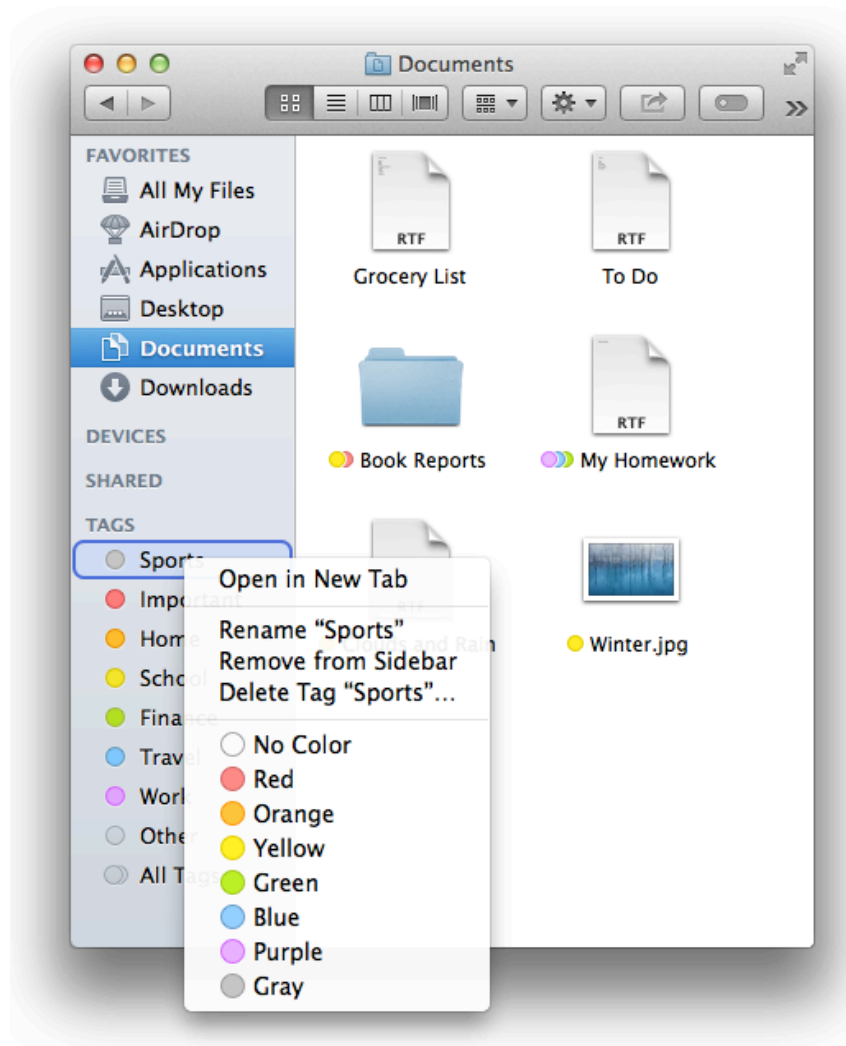


FIGURE 1 – Vue et gestion d'un tag dans le Finder macOS ([1])

Lorsque l'on clique sur un tag, une recherche Spotlight est effectuée. Spotlight est le moteur de recherche interne à macOS. Spotlight garde un index des tags, fournissant un accès rapide aux fichiers correspondants. Tous ces tags peuvent se synchroniser sur les différents "iDevices" via iCloud. Finalement, un menu de réglages permet la gestion des tags (affichage, suppression, etc.) ([1], [8]). L'implémentation de ce système utilise les extended attributes (voir section 4.2) pour stocker les tags. Les différents tags se trouvent dans l'attribut `kMDItemUserTags`, listés les uns à la suite des autres. Via le Terminal, à l'aide de la commande `mdls`, nous pouvons afficher la liste des tags associés à un fichier, nommé "Hello" pour l'exemple :

```

1 % mdls -name kMDItemUserTags Hello
2 kMDItemUserTags = (
3     Green,
4     Red,
5     Essential
6 )

```

Listing 1 – `mdls` listant les tags d'un fichier sous macOS ([2])

Ici, ce fichier "Hello" est étiqueté avec trois tags, "Green", "Red" et "Essential". Le fait que l'indexation est réalisée avec Spotlight implique une réindexation des fichiers dans le cas d'un changement de nom pour un tag donné sous macOS. Le framework système FSEvents donne une solution partielle : c'est une API (utilisée également par Spotlight) qui offre aux applications la possibilité d'être notifiées si un changement a eu lieu sur un dossier (un événement toutes les 30 secondes). FSEvents maintient des logs de ces changements dans des fichiers, les applications peuvent ainsi retrouver l'historique des changements quand elles le veulent ([11]).

### 2.1.2 Windows

À partir de Windows Vista, Microsoft a ajouté la possibilité aux utilisateurs d'ajouter des méta-données aux fichiers ; parmi ces méta-données se trouvent les tags.

## 2.2 Applications utilisateur

### 2.2.1 TagSpaces

### 2.2.2 Tmsu

### 2.2.3 Tagsistant

### 2.2.4 TaggedFrog

### 2.2.5 Tabbles

### 2.2.6 Dropbox

## 3 Analyse des besoins

surveillance des fichiers indexation

## 4 Technologies

### 4.1 Rust

### 4.2 Les extended attributes

#### 4.2.1 Théorie

Les extended attributes, ou "attributs étendus" en français, sont un moyen de stocker des méta-données des fichiers sous forme de paire `nom: valeur`. Ils sont attachés aux fichiers. De nombreux systèmes d'exploitation gèrent leur usage : ext2-3-4, XFS, Btrfs, UFS1-2, NTFS, HFS+, ZFS.

4.2.2 Petites manipulations

4.3 Notifications

5 Architecture

6 Réalisation

7 Discussion

cahier des charges rempli ? réalisation OK ? performances Rust VS C Bugs Améliorations futures

8 Conclusion

8.1 Remerciements

## 9 Références

- [1] Apple team. Os x : Tags help you organize your files. <https://support.apple.com/en-us/HT202754>, février 2015. Consulté le 08.05.2018.
- [2] John Siracusa. Os x 10.9 mavericks : The ars technica review - tags implementation. <https://arstechnica.com/gadgets/2013/10/os-x-10-9/9/>, octobre 2013. Consulté le 08.05.2018.
- [3] Rust Team. The rust programming language, 2nd edition. <https://doc.rust-lang.org/stable/book/second-edition/>. Consulté le 25.04.2018.
- [4] Manuel Hoffmann. Are we (i)de yet? <https://areweideyet.com/>. Consulté le 25.04.2018.
- [5] Computational Geometry Lab. Learning rust with entirely too many linked lists. <http://cglab.ca/~abeinges/blah/too-many-lists/book/>. Consulté le 28.04.2018.
- [6] Jean-Francois Dockes. Extended attributes and tag file systems. <https://www.lesbonscomptes.com/pages/tagfs.html>, juillet 2015. Consulté le 04.05.2018.
- [7] Jean-Francois Dockes. Extended attributes : the good, the not so good, the bad. <https://www.lesbonscomptes.com/pages/extattrrs.html>, juillet 2014. Consulté le 04.05.2018.
- [8] John Siracusa. Os x 10.9 mavericks : The ars technica review - tags. <https://arstechnica.com/gadgets/2013/10/os-x-10-9/8/>, octobre 2013. Consulté le 08.05.2018.
- [9] John Siracusa. Mac os x 10.4 tiger - extended attributes. <https://arstechnica.com/gadgets/2005/04/macosx-10-4/7/>, avril 2005. Consulté le 08.05.2018.
- [10] John Siracusa. Mac os x 10.4 tiger - spotlight. <https://arstechnica.com/gadgets/2005/04/macosx-10-4/9/>, avril 2005. Consulté le 08.05.2018.
- [11] John Siracusa. Mac os x 10.5 leopard : the ars technica review - fsevents. <https://arstechnica.com/gadgets/2007/10/mac-os-x-10-5/7/>, octobre 2007. Consulté le 08.05.2018.
- [12] Jean-Pierre André. Using extended attributes. <http://jp-andre.pagesperso-orange.fr/extend-attr.html>, janvier 2017. Consulté le 09.05.2018.