

Ce projet a pour objectif de créer un synchroniseur de systèmes de fichiers.

## 1 DESCRIPTION DU PROJET

### 1.1 Introduction

Beaucoup d'utilisateurs d'ordinateurs veulent accéder à leurs fichiers à partir de machines différentes. La méthode la plus pratique consiste à utiliser un seul système de fichiers qui est monté par plusieurs machines à travers le réseau. Les systèmes de fichiers partagés ne sont pas facilement utilisables lorsque les machines ne sont pas connectées à un même réseau. Par exemple, un utilisateur a les mêmes fichiers sur sa machine de bureau et son portable. Lorsqu'un fichier est modifié sur le portable alors que celui-ci est déconnecté du réseau, il lui faut manuellement copier la nouvelle version vers le système de fichiers qui ne contient que l'ancienne ; lorsque des fichiers ont été modifiés sur chacun des deux systèmes de fichiers, cette opération devient complexe et est la source de nombreuses erreurs.

Un logiciel de synchronisation de fichiers est un logiciel qui rend identiques deux arbres de fichiers de façon automatique en produisant autant que possible un résultat conforme aux attentes de l'utilisateur. Le but de ce projet est d'écrire un tel logiciel.

Il existe sur le marché des synchroniseurs de fichiers très complets.

### 1.2 Fonctionnement

Le but du synchroniseur est d'arriver aussi près que possible de l'état où les arbres A et B sont identiques, i.e. où si un fichier p de A (noté p/A) existe alors p de B (noté p/B) existe aussi et est identique (données et métadonnées), et vice versa. On entend par métadonnées le type, les permissions du fichier p, la taille de p ainsi que la date de dernière modification de p.

En général, si un fichier a été modifié des deux côtés, il n'est pas possible d'arriver à une synchronisation complète ; on dit alors qu'il y a conflit entre les deux versions du fichier.

Le synchroniseur de fichiers manipule deux arbres de fichiers, que nous appellerons A et B, ainsi qu'un fichier contenant le journal de la dernière synchronisation réussie. Le fichier de journal, stocké par exemple dans \$HOME/.synchro, contiendra les chemins A et B ; en outre, pour chaque fichier p/A = p/B qui a été synchronisé sans conflit, le fichier de journal contient le chemin p, le type et les permissions du fichier p, la taille de p ainsi que la date de dernière modification de p.

On dit que le fichier p/A (resp. p/B) est conforme au journal lorsqu'il existe une entrée pour p dans le fichier de journal, et le fichier p/A (resp. p/B) a les mêmes mode, taille, et date de dernière modification que ceux qui sont stockés dans le journal.

### 1.2 Synchroniseur simple

Le synchroniseur parcourt les deux arbres A et B en parallèle. Pour tout fichier p, il effectue les actions suivantes :

- Si p/A est un répertoire et p/B est un fichier ordinaire ou l'inverse, il y a conflit.
- sinon, si p/A et p/B sont tous deux des répertoires, il descend récursivement.
- sinon, si p/A et p/B sont deux fichiers ordinaires qui ont les mêmes modes, taille et date de dernière modification, la synchronisation est réussie, et il n'y rien à faire.
- sinon, si p/A est conforme au fichier de journal et p/B ne l'est pas, ce dernier a changé ; il faut alors copier le contenu, le mode et la date de dernière modification de p/B vers p/A.
- inversement, si p/B est conforme au fichier de journal et p/A ne l'est pas, c'est de p/A vers p/B qu'il faut faire la copie.
- enfin, si p/A et p/B sont tous deux des fichiers ordinaires, et aucun des deux n'est conforme au fichier de journal (soit parce qu'il n'y a pas d'entrée pour p, soit parce que celle-ci ne correspond pas aux métadonnées des deux fichiers), il y a conflit.

Le synchroniseur réécrit ensuite le fichier de journal avec les données de tous les fichiers ordinaires dont la synchronisation a réussi.

Il faudra prendre garde au fait que les deux arbres n'apparaissent pas forcément dans le même ordre dans le système de fichiers : il se peut que A contienne les fichiers p et q, dans cet ordre, tandis que B contient q et p.

A vous de décider aussi comment gérer les conflits. On pourra simplement afficher la liste des conflits une fois la synchronisation faite ; on pourra demander à l'utilisateur d'effectuer un choix. Si les fichiers sont des fichiers texte, on pourra afficher les différences entre les deux fichiers (par exemple en exécutant la commande diff).

### 1.3 Synchroniseur avec comparaison du contenu

Si un fichier a été modifié de façon identique des deux côtés, le synchroniseur ci-dessus indique un conflit fallacieux. En présence d'un conflit entre deux fichiers ordinaires, le synchroniseur avec comparaison du contenu compare le contenu des deux fichiers ; s'ils sont identiques,

- si les métadonnées des deux fichiers sont aussi identiques, il n'y a rien à faire, la synchronisation a réussi ;
- si les métadonnées d'un des deux fichiers sont identiques à celles stockées dans le journal, c'est les métadonnées de l'autre qui ont changé ; il suffit alors de changer les métadonnées du premier, et la synchronisation a alors réussi ;
- si les métadonnées diffèrent, les deux fichiers sont en conflit, mais uniquement sur les métadonnées ; il peut être utile de le faire savoir à l'utilisateur.

Bien-sûr, dans le cas où la synchronisation réussit dans l'un des cas ci-dessus, il faudra en stocker le résultat dans le fichier de journal.

## **2 EXTENSIONS**

Toutes les extensions au sujet seront les bienvenues.

L'interface utilisateur est laissée à votre discrétion. En particulier, on pourra penser à implémenter une interface utilisateur conviviale en cas de conflit, par exemple en proposant de lancer un outil de calcul de différences.

On pourra aussi penser comment gérer les situations critiques qui peuvent arriver lorsque l'un des arbres A et B est modifié durant la synchronisation.

## **3 PRECISIONS**

Pour simplifier la mise en œuvre de ce projet, le synchroniseur sera développé seulement pour effectuer la synchronisation de 2 systèmes de fichiers présents sur la même machine.

## **4 RAPPORT ET PRESENTATION**

Un rapport de quelques pages doit être envoyé à l'adresse mail [florent.retraint@utt.fr](mailto:florent.retraint@utt.fr). Une présentation et un test du shell seront effectués la semaine précédant les finaux sur Zoom. Bon courage !!!