A black and blue logo with white text

Description automatically generated

RAPPORT DE PROJET

LO14 2023/2024

Synchronisateur de fichiers

**Version 1.0**

**Membres de l'équipe**: Cristina Meda Titu, Julia Stępień

Contenu

[Introduction 4](#_Toc155786167)

[Architecture 4](#_Toc155786168)

[1. Composants and Modules 4](#_Toc155786169)

[2. Communication 4](#_Toc155786170)

[3. Gestion des données 5](#_Toc155786171)

[4. Concurrence et parallélisme 5](#_Toc155786172)

[5. Interface Utilisateur (UI) 5](#_Toc155786173)

[Application flow 5](#_Toc155786174)

[6. Performance 6](#_Toc155786175)

[Critères de synchronisation des fichiers 6](#_Toc155786176)

[Résolution des conflits pour différents types de fichiers 6](#_Toc155786177)

[Gestion des fichiers inexistants 6](#_Toc155786178)

[Gestion des fichiers non conformes 6](#_Toc155786179)

[Résolution des conflits pour les fichiers modifiés (non conformes) avec un contenu identique 7](#_Toc155786180)

[Résolution des conflits pour les fichiers avec un contenu différent, mais des métadonnées identiques 7](#_Toc155786181)

[Description de l'algorithme 7](#_Toc155786182)

[sync <path-to-tree1> <path-to-tree2> 7](#_Toc155786183)

[Étape n° 1 : Validation 7](#_Toc155786184)

[Étape n° 2 : Préparation du fichier journal 7](#_Toc155786185)

[Étape n° 3 : Préparation du fichier de sortie 8](#_Toc155786186)

[Étape n° 4 : Synchronisation - traversée simultanée des arbres 8](#_Toc155786187)

[Étape n° 5 : Afficher des informations de journalisation 8](#_Toc155786188)

[resolve <option> <path-file1> <path-file2> 9](#_Toc155786189)

[Étape n° 1 : Validation 9](#_Toc155786190)

[Étape n° 2 : Résolution 9](#_Toc155786191)

[Étape n° 3 : Gestion du journal 9](#_Toc155786192)

[Étape n° 4 : Feedback 9](#_Toc155786193)

[resolveAll <option> 10](#_Toc155786194)

[Étape n° 1 : Validation 10](#_Toc155786195)

[Étape n° 2 : Obtention des entrées de conflit 10](#_Toc155786196)

[Étape n° 3 : Resolution 10](#_Toc155786197)

[Étape n° 4 : Gestion du journal 10](#_Toc155786198)

[Étape n° 5 : Feedback 10](#_Toc155786199)

[Conclusion 11](#_Toc155786200)

# Introduction

L'objectif principal de ce projet est de développer notre propre outil de synchronisation de fichiers qui permet des synchronisations automatiques de deux systèmes de fichiers. Le programme devrait être capable de gérer les conflits, de gérer les liens symboliques et de fournir une interface simple et conviviale avec des informations claires sur le processus de synchronisation et la résolution des conflits.

Notre projet se concentre sur le développement d'un synchroniseur de fichiers pour deux systèmes de fichiers situés sur la même machine. La tâche du synchroniseur est de comparer en parallèle et de synchroniser des fichiers et des répertoires, en tenant compte du contenu et des métadonnées tout en considérant les conflits potentiels. Nous fournissons des mécanismes pour résoudre efficacement les conflits en notifiant l'utilisateur des différences. Une partie intégrante de l'outil est un fichier journal qui est mis à jour à chaque synchronisation réussie.

# Architecture

## Composants and Modules

1. Script de synchronisation nommé sync
2. Script de résolution des conflits nommé resolve
3. Script de résolution de tous les conflits nommé resolveAll
4. Le fichier journal qui stocke les deux chemins d'arborescence à synchroniser et les entrées avec les détails des fichiers qui ont été mis à jour avec succès jusqu'à présent.
5. Le fichier de conflits contenant tous les chemins en conflit trouvés après l'exécution du script sync.

## Communication

Le programme sync manipule les données contenues dans le *journal*, ajoutant ou modifiant les métadonnées des fichiers qui ont été synchronisés pour les deux arbres. Il génère également le contenu du fichier de conflits avec les chemins problématiques trouvés.

Les scripts resolve et resolveAll utilisent les données stockées dans le fichier de conflits à des fins de validation (programme resolve) ou en tant que fichier d'entrée (programme resolveAll).

## Gestion des données

Mécanismes de stockage:

1. Le fichier journal qui enregistre les détails des sources de modifications (chemin du fichier, type, permissions, taille, date de la dernière modification).
2. Le fichier de conflits qui enregistre tous les chemins ayant des problèmes de synchronisation et les raisons de ces problèmes.
3. Les fichiers auxiliaires paths1, paths2, paths\_tree1, paths\_tree2, only\_tree1, only\_tree2, utilisés couramment dans la résolution de la tâche, sont supprimés à la fin du programme.

## Concurrence et parallélisme

Les deux chemins d'arborescence sont parcourus en parallèle lors de la vérification et de la synchronisation.

## Interface Utilisateur (UI)

L'interface utilisateur est configurée sous la forme d'une interface en ligne de commande, offrant les fonctionnalités suivantes:

Synchroniser deux chemins d'arborescence, afficher et stocker les conflits (fichier *conflicts*) et suggérer les méthodes de résolution : **sync <path-to-tree1> <path-to-tree2>**

Résoudre un conflit entre deux chemins à partir du fichier généré par la commande sync (conflits) avec la méthode sélectionnée (option) : **resolve <option> <path-file1> <path-file2>**

Résoudre tous les conflits à partir du fichier généré par la commande sync (conflits) avec la méthode sélectionnée (option) : **resolveAll <option>**

### Application flow

L'utilisateur saisit la commande associée, sync, en fournissant les deux chemins d'arborescence à synchroniser en tant qu'arguments. Après validation, si les répertoires sont trouvés, l'application lance le processus de synchronisation. Si les chemins ne correspondent à aucun répertoire existant, la première étape est répétée jusqu'à ce que l'utilisateur saisisse deux chemins valides. Après finalisation, l'application affiche la liste des conflits et les options de résolution parmi lesquelles l'utilisateur peut choisir. Le programme enregistre également les chemins des conflits ainsi que leur raison dans un nouveau fichier. Pour résoudre un conflit, l'utilisateur saisit la commande resolve, mentionnant l'option de résolution et les deux chemins en conflit. Un message est affiché à la fin. Une autre possibilité est de résoudre tous les conflits en une fois, en utilisant la commande resolveAll qui prend tous les chemins de fichiers de conflit à partir du fichier de sortie généré par la commande sync et les résout en utilisant la méthode (option) choisie.

## Performance

Utilisation d'algorithmes efficaces pour la comparaison et la synchronisation.

Cas couverts

Dans le contexte de la synchronisation de fichiers entre deux systèmes de fichiers A et B, les règles suivantes sont appliquées:

## Critères de synchronisation des fichiers

Si le fichier est ordinaire avec des données et des métadonnées identiques dans A et B, considérez-le comme synchronisé sans autre action nécessaire.

Vérification récursive des repertoires

Si le fichier est présent à la fois dans A et B et de type répertoire, initiez un examen récursif des fichiers contenus dans les répertoires respectifs.

### Résolution des conflits pour différents types de fichiers

Si le fichier est présent dans les deux systèmes de fichiers mais qu'une occurrence est un fichier ordinaire tandis que l'autre est un répertoire, catégorisez-le comme un conflit.

### Gestion des fichiers inexistants

Si le fichier existe dans A mais ne peut pas être trouvé dans B, exécutez l'opération de copie du fichier de A vers B. Si le fichier existe dans B mais est absent dans A, agissez en conséquence.

### Gestion des fichiers non conformes

Si le fichier dans A est conforme au fichier journal, mais le fichier dans B ne l'est pas, le fichier dans B a été mis à jour. Par conséquent, le contenu et les métadonnées du fichier dans B doivent être copiés vers le fichier dans A, et vice versa.

Si les deux fichiers dans A et B ne correspondent pas au fichier journal (ont été modifiés) et ont des contenus différents, marquez-les comme conflit. En cas de contenu identique, la section suivante décrit la solution appropriée.

### Résolution des conflits pour les fichiers modifiés (non conformes) avec un contenu identique

Si les deux fichiers ont été modifiés mais que leurs contenus sont identiques :

a. si les métadonnées sont identiques, aucune autre action n'est nécessaire.

b. si les métadonnées du premier fichier correspondent aux métadonnées stockées dans le journal, ce qui signifie que le deuxième fichier a été modifié, mettez à jour ses métadonnées pour correspondre au fichier modifié.

c. si les métadonnées des deux fichiers diffèrent de celles du journal ou si l'entrée n'est pas trouvée, catégorisez-le comme un conflit et avertissez l'utilisateur en conséquence.

### Résolution des conflits pour les fichiers avec un contenu différent, mais des métadonnées identiques

Si les deux fichiers ont les mêmes métadonnées, enregistrées ou non dans le *journal* (coïncidence de taille, de date et d'heure), mais que le contenu est différent, marquez-le comme un conflit.

Ces règles établissent une approche efficace et systématique de la synchronisation de fichiers afin de travailler correctement à travers le processus de synchronisation et de gestion des conflits entre les systèmes de fichiers A et B.

À chaque synchronisation réussie, le *journal* est mis à jour pour l'entrée associée.

# Description de l'algorithme

## sync <path-to-tree1> <path-to-tree2>

Le script de synchronisation prend en argument deux chemins absolus de fichiers représentant deux arbres de fichiers et les synchronise tout en marquant les conflits entre eux.

### Étape n° 1 : Validation

Les chemins sont vérifiés pour s'assurer qu'il s'agit de répertoires existants. En cas d'arguments invalides, un message d'erreur sera affiché.

### Étape n° 2 : Préparation du fichier journal

Le fichier de suivi de la synchronisation des deux arbres, nommé "*journal*", est en cours de préparation. S'il n'existe pas, créez-en un nouveau vide. Les chemins des arbres sont écrits sur la première ligne, séparés par un espace, afin d'indiquer les sources de synchronisation. S'il existe déjà, et si les chemins ne correspondent pas à ceux marqués dans le journal, supprimez tout le contenu et créez un nouveau fichier journal pour les nouveaux chemins d'arbre.

### Étape n° 3 : Préparation du fichier de sortie

Le fichier de sortie, nommé "*conflicts*", est en cours de préparation pour une utilisation ultérieure. S'il n'existe pas, créez-en un nouveau vide. Sinon, effacez son contenu.

### Étape n° 4 : Synchronisation - traversée simultanée des arbres

L'algorithme commence par rechercher les fichiers exclusivement présents dans le premier arbre et les copie vers l'autre. Ensuite, le même processus est répété pour le deuxième arbre. Cela se fait en triant les deux fichiers contenant l'ensemble des chemins des deux arbres et en les comparant avec la commande *comm*. L'option "-23" signifie de supprimer les lignes uniques aux deux fichiers et d'imprimer uniquement les lignes uniques au premier fichier. L'option "-13" ne prendra que les lignes uniques du deuxième fichier. Enfin, nous appelons la même commande mais avec l'option "-12" pour récupérer uniquement les chemins communs des deux fichiers.

La fonction *copy\_content* prend en arguments le fichier contenant les chemins à copier, le chemin absolu de l'arbre source et le chemin absolu de l'arbre de destination. Son rôle est de copier les fichiers et répertoires mentionnés dans le fichier d'arguments depuis l'arbre source vers l'arbre de destination.

Par conséquent, *copy\_content* est appliquée aux deux fichiers générés avec la commande *comm*, s'assurant que le contenu trouvé uniquement dans l'un des arbres est copié vers l'autre.

La prochaine étape consiste à lire chaque chemin depuis le fichier des chemins communs, construire les chemins absolus dans les deux arbres, puis vérifier chaque cas mentionné ci-dessus, dans la section *Cas couverts*, et les gérer efficacement comme décrit.

En cas de conflit, les chemins problématiques et la raison du conflit sont ajoutés en une seule ligne au fichier de sortie (*conflicts*).

En cas de synchronisation, le chemin du fichier, son type, sa taille, ses autorisations et la date de la dernière modification sont ajoutés au fichier *journal*, ou l'entrée correspondante, si elle existe, est mise à jour.

### Étape n° 5 : Afficher des informations de journalisation

Les informations de synchronisation sont affichées pendant le processus: le fichier actuel en cours de vérification, les métadonnées dans les deux arbres, l'entrée éventuelle si elle existe, et les raisons de synchronisation/conflit.

Étape n° 6 : Afficher des suggestions pour résoudre les conflits

Un message d'information est affiché à la fin, contenant les commandes à utiliser pour résoudre les conflits (resolve et resolveAll) avec les options disponibles (keep-first, keep-second, keep-initial, keep-latest).

## resolve <option> <path-file1> <path-file2>

Le script de résolution prend en arguments une variable spécifiant la façon de gérer le conflit et les deux chemins absolus des fichiers nécessitant cette résolution. Le script résout un conflit spécifique entre deux fichiers donnés en utilisant l'option sélectionnée par l'utilisateur et l'historique de résolution des métadonnées est mis à jour dans le fichier *journal*.

### Étape n° 1 : Validation

L'algorithme gérant les options vérifie si l'option spécifiée par l'utilisateur est l'une de celles prises en charge par le script. Si l'option est invalide, alors le programme renvoi un message d'erreur est affiche que l'option sélectionnée est invalide.

Le fichier *conflicts* vérifie si le conflit provient bien des fichiers fournis. Si aucun conflit n'est trouvé au sein des fichiers fournis, alors un message d'erreur est affiché indiquant qu'aucun conflit n'a été trouvé entre les fichiers fournis.

### Étape n° 2 : Résolution

Le script lit chaque ligne à partir d'un fichier nommé "*conflicts*", où chaque ligne contient deux chemins en conflit.

Pour chaque entrée en conflit : il détermine les types des chemins en conflit (répertoires ou fichiers réguliers), en fonction de l'option de résolution, effectue des actions telles que la copie, la suppression ou la mise à jour des fichiers.

### Étape n° 3 : Gestion du journal

Le fichier *journal* est mis à jour avec les métadonnées résolue.

### Étape n° 4 : Feedback

Pendant la résolution le script affiche des messages indiquant l'état de résolution et la synchronisation de chaque fichier ou répertoire. Si toutes les étapes du script se sont bien exécutées et que le journal a bien été mis à jour, un message de réussite est affiché. À la fin, vérifie s'il reste des conflits dans le fichier "*conflicts*" et affiche un message récapitulatif.

#### Options disponibles:

***keep-first*** *-* sélectionne et garde la partie en conflit du premier fichier, l'autre est mis à jour avec les données sélectionnées ainsi que ses métadonnées

***keep-second*** *-* sélectionne et garde la partie en conflit du second fichier, l'autre est mis à jour avec les données sélectionnées ainsi que ses métadonnées

***keep-latest***- sélectionne et garde la partie en conflit du fichier modifié le plus récemment à l'aide du timestamp, l'autre est mis à jour avec les données sélectionnées ainsi que ses métadonnées

***keep-initial*** *-* sélectionne et garde la partie en conflit du fichier modifié il y a le plus longtemps à l'aide du timestamp, l'autre est mis à jour avec les données sélectionnées ainsi que ses métadonnées

## resolveAll <option>

Le script de résolution prend un seul argument en entrée, un potion spécifiant la façon de régler le conflit. Le script résout tous les problèmes venant du fichier conflicts en utilisant l'option sélectionnée par l'utilisateur et les métadonnées résolues sont mises à jour dans le fichier journal.

### Étape n° 1 : Validation

Le script vérifie si le paramètre d'option est bien supporté par le programme. Si l'option n'est pas valide, alors un message d'erreur est affiché indiquant une option invalide.

### Étape n° 2 : Obtention des entrées de conflit

Les lignes du fichier conflicts sont lues dans l'ordre pour obtenir les chemins des fichiers pour chaque conflit.

### Étape n° 3 : Resolution

Pendant la lecture du fichier, la méthode de résolution utilisée est celle spécifiée par l'utilisateur en options sur toutes les lignes de conflits.

### Étape n° 4 : Gestion du journal

Pour chaque conflit résolu, le fichier journal est mis à jour avec les métadonnées de résolution.

### Étape n° 5 : Feedback

Si toutes les étapes se sont bien passées et que le journal a bien été mis à jour, un message de réussite est affiché.

# Conclusion

Le projet de synchronisateur de fichier présente une solution simple pour synchroniser deux systèmes de fichiers, gérant les conflits avec une grande possibilité d'options de résolution disponibles. L'interface en lignes de commande, combiné avec le gestionnaire de données et le mécanisme de résolution de conflits, permet une gestion user-friendly et efficace de l'expérience de synchronisation de fichiers. Le design modulaire du projet et les méthodes pour être conforme avec ce que l'utilisateur a demandé permettent une versatilité et une utilisation du projet dans diverses scénarios de synchronisation.