



Projet LP25 A2023

Frédéric Lassabe



UNIVERSITÉ
FRANCHE-COMTÉ



Le projet dans les grandes lignes

- ▶ Reproduire un comportement simplifié de la commande *rsync*
 - ▶ Fait une sauvegarde depuis une source
 - ▶ Le dossier à sauvegarder
 - ▶ Qui peut être sauvegardé plusieurs fois
 - ▶ Vers une destination
 - ▶ Contiendra la sauvegarde
 - ▶ Peut être mise à jour par des appels successifs
 - ▶ Sauvegarde récursive
 - ▶ on descend dans les répertoires jusqu'au bout
 - ▶ uniquement des fichiers réguliers et des dossiers
 - ▶ copie avec date de modification identique
 - ▶ copie avec droits d'accès identiques
 - ▶ si taille et date identiques, la somme MD5 est utilisée
- ▶ Développement en C
- ▶ suivant les notions de LP25 (cours, TD, TP)
- ▶ poussées plus loin sur un cas concret et complexe

Principe de l'application



- ▶ Deux implémentations différentes
 - ▶ Séquentielle
 - ▶ Parallèle (processus)
 - ▶ Code commun important
- ▶ Séquentiellement :
 - ▶ Créer 2 listes des fichiers/dossiers et leurs propriétés
 - ▶ Une pour la source
 - ▶ Une pour la destination
 - ▶ En déduire à partir de la source une liste des différences, i.e. :
 - ▶ Entrées de la source absentes de la destination
 - ▶ Entrées de la source différentes (fonction *mismatch*) de la destination
 - ▶ Copier toutes les entrées sources de la liste de différences vers la destination



Méthode de travail



- ▶ Sujet assez long et détaillé
- ▶ Squelette de code fourni (signatures des fonctions)
- ▶ Travail :
 - ▶ Implémenter les fonctions
 - ▶ En se conformant aux instructions
 - ▶ Et commentaires doxygen
- ▶ Prendre connaissance de tout le sujet avant de commencer
 - ▶ Des fonctions en appellent d'autres
 - ▶ Bien comprendre les fonctions pour en déduire les dépendances



Méthode de travail – suite



- ▶ Travailler en groupe
 - ▶ Se répartir les tâches
 - ▶ Se donner des jalons intermédiaires
 - ▶ Présenter les réalisations au groupe
 - ▶ Échanger les rôles
- ▶ Deux dates pour le code
 - ▶ Le 10 décembre pour une version préliminaire (au moins le séquentiel fonctionne)
 - ▶ Rendu important : il sera relu et annoté
 - ▶ Permettre la progression
 - ▶ Le 7 janvier pour la version finale (tout est opérationnel)





- ▶ Le groupe
 - ▶ Faire un groupe (3 ou 4 membres) qui peut travailler ensemble
 - ▶ Notamment : emploi du temps
 - ▶ Donc : même groupe de TP, ou groupes de TP sur lesquels tous les membres peuvent être libres
 - ▶ Mixer les compétences :
 - ▶ Organisationnelles
 - ▶ Techniques (code, environnement)
 - ▶ Algorithmique



Gestion de projet (2)



- ▶ Les outils
 - ▶ L'IDE
 - ▶ Tout le groupe avec le même IDE
 - ▶ Toujours produire un Makefile compilable hors IDE
 - ▶ Le code
 - ▶ Fonctionne sous Linux natif ou VM, pas de WSL
 - ▶ Travail collaboratif avec git
 - ▶ Code rendu comme un lien vers un dépôt git
 - ▶ Avec historique des contributions des membres
 - ▶ Équilibrer le travail entre les membres



Le RETEX

- ▶ Demandé dans le rapport et la soutenance
- ▶ Outil essentiel de l'évolution des organisations et de leurs membres
- ▶ Le RETEX doit permettre une analyse sans concession
 - ▶ Ce qui a dysfonctionné (pourquoi ? comment éviter que ça se reproduise ?)
 - ▶ Les points de friction (signaux faibles de potentiels dysfonctionnements), à traiter comme les dysfonctionnements
 - ▶ Ce qui a bien fonctionné (et qui pourrait servir de bonne pratique)
- ▶ Le RETEX doit en particulier identifier les mauvaises pratiques pour les remplacer par des pratiques plus sûres et performantes.



Questions ?

