

Projet C – WildWater

Filière : préING2 – 2025/2026

MI-1 : Alexandre Gourdon, Yasmine El berraze, Thibault Joubert

1. Présentation générale du projet

Le projet C – WildWater consiste à développer une application permettant d'analyser des données issues d'un réseau de distribution d'eau potable. Les données sont fournies sous forme d'un fichier csv de grande taille représentant un graphe allant des sources de captage jusqu'aux usagers finaux.

L'application repose sur deux éléments principaux :

- Un script Shell : qui constitue le point d'entrée de l'application et gère l'interaction avec l'utilisateur,
- un programme en langage C : chargé d'effectuer les calculs complexes afin d'assurer de bonnes performances malgré le volume important de données.

Le projet a été réalisé sur une durée de trois semaines, en combinant le travail effectué lors des séances de cours d'informatique, des réunions de groupe, ainsi qu'un suivi régulier de l'avancement via GitHub.

2. Répartition des tâches au sein du groupe

Le travail a été réparti de manière claire afin de permettre une bonne organisation et une avancée cohérente du projet.

Développement du script Shell (Yasmine et Thibault)

- Conception et écriture du script Shell principal
- Gestion des arguments utilisateur (histo, leaks, max, src, real)
- Vérification de la validité des commandes et gestion des messages d'erreur
- Préparation et organisation des fichiers d'entrée et de sortie
- Lancement automatique de la compilation du programme C via make
- Génération des fichiers de résultats et des histogrammes (Gnuplot)
- Tests fonctionnels et correction des erreurs
- Rédaction partielle de la documentation utilisateur (Readme)

Développement du programme en langage C (Alexandre)

- Développement du programme C réalisant les traitements demandés
- Lecture et analyse du fichier csv
- Calcul des histogrammes des usines :
 - capacité maximale de traitement,
 - volume total capté par les sources,
 - volume réellement traité après pertes
- Calcul du volume total des fuites pour une usine donnée

- Mise en place de structures de données adaptées (AVL, listes chaînées)
- Gestion de la mémoire et des codes de retour
- Création et maintenance du Makefile

3. Organisation et planning de réalisation

Le projet a été organisé sur une période de trois semaines, avec une progression régulière.

Semaine 1 – Analyse et mise en place

- Lecture et compréhension du sujet
- Analyse du format des données CSV
- Répartition des tâches au sein du groupe
- Création du dépôt GitHub
- Premiers tests de lecture et de filtrage des données

Semaine 2 – Développement

- Développement du script Shell
- Développement du programme C
- Mise en place des structures de données principales
- Premiers tests fonctionnels
- Commits réguliers sur GitHub

Semaine 3 – Intégration et finalisation

- Liaison entre le script Shell et le programme C
- Tests des commandes histo et leaks
- Génération des fichiers de résultats finaux
- Correction des bugs identifiés
- Nettoyage du code et ajout de commentaires
- Finalisation de la documentation et des tests

4. Méthode de travail et collaboration

Le travail a été réalisé de manière collaborative :

- échanges réguliers pendant les séances de cours,
- réunions de groupe pour faire le point sur l'avancement,
- utilisation systématique de GitHub pour le suivi du projet et l'historique des modifications,
- commits fréquents permettant de conserver une trace claire du travail effectué.

Cette organisation a permis de coordonner efficacement les différentes parties du projet.

5. Limitations fonctionnelles de l'application

L'application c-wildwater présente plusieurs limitations fonctionnelles liées à ses choix de conception et à son périmètre d'utilisation.

Tout d'abord, le fonctionnement de l'application dépend strictement du format et de la cohérence des fichiers d'entrée. Toute anomalie dans la structure des données (champs manquants, séparateurs incorrects, valeurs non numériques) peut entraîner un rejet des lignes concernées ou des résultats incomplets.

Ensuite, la gestion des erreurs reste volontairement limitée. L'application ne fournit pas de messages détaillés en cas de données invalides ou de comportements inattendus, ce qui peut compliquer le diagnostic pour un utilisateur non technique.

Par ailleurs, les performances peuvent se dégrader lorsque l'application traite des fichiers de très grande taille. Bien que les structures de données utilisées soient efficaces, l'exécution reste séquentielle.

L'application est également restreinte à une utilisation en ligne de commande. Elle ne propose ni interface graphique ni interface web, ce qui limite son accessibilité à un public non familiarisé avec les environnements Unix.

Enfin, les fonctionnalités sont fixes et non extensibles dynamiquement. Toute évolution (nouveau mode de calcul, nouveau type de sortie ou format différent) nécessite une modification du code source et une recompilation de l'application.

6. Conclusion

Au début, le projet C – WildWater nous a semblé complexe, notamment à cause du volume des données et des contraintes techniques imposées. La compréhension du sujet et l'organisation du travail ont demandé un certain temps, mais au fur et à mesure de l'avancement, nous avons réussi à structurer le projet et à répartir efficacement les tâches. Grâce au travail en groupe, aux séances en cours et aux échanges réguliers, nous avons pu mettre en place les fonctionnalités principales attendues. Malgré certaines limites et améliorations possibles, nous sommes satisfaits du résultat obtenu au regard du temps imparti et des objectifs fixés.

