

Carnet de bord du groupe MEF2-D du projet “WildWater”

Table des matières :

- Répartition des tâches au sein du groupe : 3
- Planning de réalisation : 4-5
- Difficultés rencontrées : 6
- Les limites du projet : 6

Répartition des tâches au sein du groupe :

- **Siméon** s'est occupé du C (majeure partie du code en C est de lui),
- **Timothée** du Shell (Shell.sh),
- **Alexandre** a fait le Makefile, le readme et amélioré/fait fonctionner les différentes fonctions de Siméon et Timothée (travail en Shell et code C : correction du bug de Leaks par exemple).

Planning de réalisation :

Planning théorique :

1. PHASE 1 : Les Fondations

- Définir les structures de données (Structures C et AVL).
- Coder les fonctions de base de l'AVL (Insertion, Rotation, Équilibrage).
- *Test* : Créer un main qui insère 3 éléments "bidons" et affiche l'arbre.

2. PHASE 2 : Le Parsing (Lecture du CSV)

- Coder la lecture du fichier ligne par ligne.
- Remplir l'AVL avec les données réelles (filtrage pour ne garder que les usines au début).
- *Test* : Vérifier qu'on a bien toutes les usines en mémoire.

3. PHASE 3 : Traitement Histogramme (Histo)

- Ajouter la logique de sommation (Capacité, Volume Source, Volume Traité).
- Générer le fichier `.csv` temporaire de sortie.

4. PHASE 4 : Le Script Shell

- Coder le script qui compile le C, lance le traitement "Histo", et appelle Gnuplot.

5. PHASE 5 : Les Fuites

- Modifier des structures pour gérer la partie des fuites (listes d'adjacence pour les enfants)

L'objectif était de finir la phase 5 pour le lundi 15/12.

En pratique :

30/11 : création du dépôt github (vide) et lecture intensive du sujet

02/12 : Structure du dépôt + Début d'écriture du code

04/12 : Ecriture des fonctions avl + début du travail sur input.c et d'output.c

08/12 : création du Shell et création des parties pour activer les histogrammes et les fuites, fin de "afficherNoeud()" dans output.c

09/12 au 11/12 : *en Shell* : show help + gnuplot

en C : ajout de la lecture du fichier dans input + ajout de fonction dans le header

12/12->13/12 : création et fin du Makefile.

En C : fin des versions de base de output, input et main pour les histogrammes

14/12->18/12 : tout le travail pour faire marcher les fuites (leaks)

15/12 : *en Shell* : séparation des différents modes

16/12->18/12 : *en Shell* : travail sur le tri des 50 et 10 usines

17/12 : *en C* : adaptation de input et output pour le tri

19/12 -> 21/12 : écriture du Readme (début le 13/12), du pdf et nettoyage des commentaires

Difficultés rencontrées :

Tout d'abord, il y a l'utilisation de deux langages différents dont un qui était tout nouveau : nous avons commencé à l'utiliser deux semaines avant le projet.

Le fait de lier les deux ensemble représentait un obstacle.

Ensuite, il y a la longueur du projet car, même si cela ne se voyait pas à première vue, il a fallu faire plein de fonctions qui n'étaient pas décrites explicitement dans le sujet pour faire fonctionner le projet dans son ensemble.

De plus, notre propre gestion n'a pas été optimale, en effet notre plan et notre planning de base n'ont pas été totalement respectés.

Il y a aussi l'affichage de l'histogramme où il y a du mal à comprendre comment le faire (par exemple, pour le tri des 10 plus grandes usines et des 50 plus petites).

Enfin, pour la commande "leaks", nous avons eu beaucoup de difficultés à trouver la cause de l'affichage de valeurs fausses : la raison était que l'on prenait la capacité "max" de l'usine au lieu de la capacité réelle, ce qui donnait un résultat faux (trop grand).

limites du projet :

Toutes les fonctionnalités de base sont implémentées (les commandes histo et leaks).

Une erreur que l'on a trouvée est lorsque l'on écrit la commande pour lancer le projet pour leaks il ne faut pas mettre un espace de plus sinon les valeurs affichées ne sont pas bonnes.