

Rapport

Wildwater

Démarrage et mise en place

Nous sommes un groupe de trois dont les membres sont MACHEFERT Tom, SCHAFF Anton et ELIAS Charbel. Nous avons d'abord créé un fichier drive où nous avons enregistré toutes les consignes relatives au sujet (histogrammes, fuites et logique de lecture du format du fichier d'entrée...). Ceci nous a vraiment aidé à comprendre le sujet et nous a donné une idée globale de ce qui était attendu. Nous avons ensuite déposé l'URL de dépôt de notre projet. En revanche, par la suite, nous avons commencé à coder chacun de notre côté sur des environnements différents dont VS Code et Linux.

Avancement et organisation

Avant même de commencer à coder, nous avons discuté à l'oral des quelques tâches que chacun avait à faire: Tom s'est penché sur la partie Shell, Anton sur la partie lecture des fichiers d'entrée et Charbel sur la partie affichage des histogrammes. Chacun s'est concentré sur la partie fonctionnelle de ses tâches et par conséquent, on s'est éloigné des consignes à respecter. Par exemple, la partie shell s'occupait de filtrer et de trier les données du fichier d'entrée sans utiliser d'AVL dans un programme C. Aussi, on pensait que le programme C devait se charger de l'affichage des histogrammes car on ignorait l'existence de la bibliothèque Gnuplot dans Shell.

La prochaine étape était donc de créer un plan le plus détaillé possible de ce qui nous reste à faire pour la mise en place du programme (les structures, les AVLs, les fonctions et les fichiers). Nous avons donc défini les fichiers de notre projet. Afin de mieux organiser notre travail et d'éviter les conflits de code, nous avons décidé de travailler sur des branches séparées du dépôt Git. Chaque membre développait sa partie sur sa branche, puis nous effectuions des "merge" réguliers vers la branche principale. Cette méthode s'est avérée nécessaire car chacun avait besoin des fichiers développés par les autres pour pouvoir avancer sur sa propre partie. Même si cela a parfois entraîné des conflits à résoudre, cette organisation nous a permis de centraliser progressivement l'ensemble du projet dans la branche principale.

Nous avons aussi été confrontés à des problèmes liés à l'environnement de travail et à la taille des fichiers fournis. En effet, lors des premiers tests réalisés sous Linux, nous avons rencontré des difficultés avec le fichier de 500 MO, ce qui a mis en évidence des problèmes de performances et de temps d'exécution. C'est surtout Tom qui se chargeait de compiler et d'effectuer les tests de notre programme car c'est le seul d'entre nous qui travaille sur linux.

Les derniers jours avant le rendu du projet, on s'est tous penché sur les parties des autres pour corriger et essayer d'optimiser notre programme au mieux.

Résultats

Les résultats obtenus sont satisfaisants pour les fonctionnalités principales du projet. Les parties concernant les histogrammes fonctionnent correctement : les modes max, réel et src produisent bien les fichiers.dat attendus ainsi que les images.png correspondantes. Ces résultats sont exploitables et conformes aux consignes demandées.

En revanche, la partie concernant les fuites s'est révélée plus difficile. Le mode leaks nécessite un traitement plus lourd des données et entraîne un temps d'exécution important, pouvant atteindre 4 minutes. Cette partie a été difficile à corriger et à optimiser dans le temps imparti, notamment à cause de la complexité du rattachement des infrastructures et des calculs associés. Malheureusement, le temps nous aura manqué et le programme prend beaucoup de temps à générer un résultat.

Enfin, le script Bash est fonctionnel et permet de filtrer correctement les données en fonction des options choisies. Cependant, certaines étapes de filtrage peuvent également prendre du temps lorsque le fichier d'entrée est volumineux, ce qui contribue au temps total d'exécution.

Conclusion

En conclusion, durant les semaines consacrées à l'élaboration du projet, l'idée de base a évolué et d'autres idées ont fait surface, nous permettant de simplifier ou de rendre plus robuste le programme au fil du temps. Ce projet nous a surtout appris l'importance de bien respecter le cahier des charges dès le départ, car nous nous sommes parfois rendu compte que certaines consignes importantes avaient été mises de côté au profit de solutions qui semblaient plus simples sur le moment. Malgré ces difficultés, ce projet nous a permis de mieux comprendre la gestion de projets en groupe, l'organisation du travail et les contraintes liées au traitement de grandes quantités de données en langage C.