

Informatique III

P2 – MI 1 – Groupe I

2024/2025

Rapport de projet C-Wire

Kentaro LORET

Nourhene Merai

Mathis SINET

Sommaire :

- I. Répartition des tâches et planning de réalisation
- II. Limites fonctionnelles

Ceci est le rapport du projet d'informatique du semestre 3 pour le groupe P2-MI1-I. Nous y présenterons la répartition des tâches au sein de l'équipe, le planning de réalisation ainsi que les limitations de notre projet.

I. Répartition des tâches et planning de réalisation

Nous avons travaillé sur ce projet entre le mercredi 4 décembre et le dimanche 22 décembre. Dès le premier jour, la partie C a été attribuée à Kentaro et la partie Shell à Mathis. Nourhene n'était pas disponible au début, elle a notamment contribué aux commentaires et à l'information utilisateur.

Le premier jour, nous avons créé le script Shell et ajouté la vérification des paramètres. Le 5 décembre, Mathis a commencé à expérimenter le filtrage des données d'entrée avec « awk » tandis que Kentaro a commencé à travailler sur le C.

La semaine suivante, nous avons continué à travailler sur la partie C et la partie Shell pour les faire communiquer et répondre à l'objectif principal.

C'est également ici que l'on a décidé, sous les conseils des professeurs, de changer le filtrage de « awk » vers « grep » afin d'améliorer les performances. Nous avons testé plusieurs possibilités pour améliorer les performances de la partie filtrage, avec notamment l'utilisation de « pipes » pour limiter les accès disque. À la fin de la semaine, le cahier des charges du fonctionnel était terminé et la vitesse satisfaisante.

La dernière semaine a été consacrée à la correction de bugs, l'ajout des graphiques pour le traitement « lv_all », l'ajout de commentaires, du texte d'aide, des fichiers tests, du readme et de la rédaction du rapport.

II. Une limitation fonctionnelle : la vitesse d'exécution

La vitesse d'exécution a été notre principal obstacle lors de la réalisation de ce projet, notamment avec de gros fichiers d'entrée tels que « c-wire_v25.dat ». Si nous avons apporté plusieurs améliorations à la vitesse durant le développement, la vitesse peut ne pas encore être satisfaisante sur certaines configurations de machines avec des fichiers lourds.

Une piste d'amélioration envisagée aurait été d'utiliser des « pipe » avant l'exécution du programme C, néanmoins nous avons créé un fichier temporaire pour rendre le code plus clair et plus facilement modulable, mais aussi car nous avons choisi de récupérer le nombre de lignes du fichier temporaire afin de le transmettre à l'exécutable C, car nous avons eu un bug précédemment où le programme C lisait la dernière ligne deux fois, notre solution a été de sécuriser en faisant lire au programme C exactement le nombre de lignes du fichier.