



Rapport Projet OS partie 1

Realisé par: Kevin Issa et Arthur

matricule Kevin: 000514550

matricule Arthur:

email: kevin.issa@ulb.be et

Octobre 2022

Table des matières

1	Introduction:	3
2	Présentation du projet:	3
3	Problématique du projet:	3
3.1	Organisation du travail de groupe:	3
3.2	Choix du type de programmation:	3
3.3	Problème d'implémentation des queries:	4
4	Conclusion:	4

1. Introduction:

Après avoir appris le fonctionnement matériel d'un ordinateur, la prochaine étape est de plus en savoir sur le système d'exploitation et les méthodes plus avancées pour créer un programme.

C'est dans ce but que s'inscrit ce projet, sous couvert du thème de la base de donnée, nous appliquons un moyen d'exécuter en parallèle plusieurs parties du même programme grâce aux processus et de les faire communiquer entre eux avec des pipes. Dès lors, nous allons vous expliquer dans ce rapport notre manière de procéder ainsi que nos choix de conception et les difficultés rencontrées.

2. Présentation du projet:

Ce projet consiste en une base de donnée d'étudiants où l'utilisateur peut choisir des requêtes grâce à des commandes terminales telles qu'insérer un étudiant, si son id n'est pas déjà présent dans la base de donnée, de sélectionner une série d'étudiants correspondant à un filtre `champ=valeur` qui seront envoyés dans un fichier créé pour, d'en enlever suivant le même filtre ou de mettre à jour leurs informations grâce aux filtres `champ=valeur(appel à select)` `champ modifié = valeur modifiée`.

3. Problématique du projet:

3.1. Organisation du travail de groupe:

Etant donné qu'il s'agit de notre premier travail en groupe, un problème que l'on a eu était qu'étant donné que nous avions des méthodes de programmation différentes et des habitudes différentes lors du choix des noms de variables, lors de la mise en commun de fonctions liées, la relecture fut difficile et nous avons alors pris conscience du problème de compréhension du code qui s'est posé.

Dès lors, nous avons décidé de demander à des personnes du milieu professionnel, comment ils travaillent en équipe. Ainsi nous avons séparé notre github en branche pour chaque fonctionnalité du projet sur lequel nous travaillons qui seront à la fin merge sur le main. De plus, nous avons posé des règles de programmation lorsque nous codons, telles que nommer les variables de telle manière, laisser des commentaires pour expliquer la logique derrière les fonctions et prévenir lorsque l'on travaille sur une branche et lorsque l'on commit et push quelque chose afin d'éviter les cas d'overwrite lorsque nous travaillons sur la même branche.

3.2. Choix du type de programmation:

Le projet étant à la base en C puis changé en c++ suite à un changement de consignes, 2 choix s'offraient à nous. Etant donné que nous avions commencé en C nous pouvions continuer comme cela et ne pas prendre le risque de devoir modifier des fonctions

pour la partie 2 ou passer en c++ et profiter dès lors de la facilité qu'apporte le langage comparé au C (vecteur,string,référence,etc).

Dès lors, nous avons décidé de passer en c++ mais en gardant une forme de compatibilité avec C. En effet, nous avons décidé d'uniquement utilisé les strings et les vecteurs afin de nous faciliter un peu le codage tout en pouvant facilement repasser a du C si la partie 2 du projet l'exige. Par conséquent, nous n'avons pas utilisé les classes de c++ car cela demanderait de modifier profondément les fichiers fournis en prenant beaucoup de temps à faire cela sans avoir beaucoup de gains.

3.3. Problème d'implémentation des queries:

Lors de l'implémentation des queries, une des difficultés durant le parcours de la base de donnée est qu'il fallait trouver une méthode pour savoir quel champ l'utilisateur a utilisé comme filtre. La méthode simple étant de faire une série de if, else if

Cependant, cette méthode n'est pas très belle en terme de codage. Dès lors, nous avons cherché une méthode alternative qui permettrait de raccourcir le code sans le rendre plus dur à lire. Par conséquent, une des méthodes possibles serait d'utiliser un switch case mais étant donné que les switch cases de c++ sont limités aux int et que nous ayons trouvé aucune meilleur méthode que le if, else if, nous avons codé les queries, select, delete et update de cette manière, même si nous pouvions utiliser des opérateurs logiques, nous trouvions cela moins lisible.

4. Conclusion: