

Rapport d'OS

Andrius Ezerskis & Moïra Vanderslagmolen & Enes

October 29, 2022

Introduction

1 Stockage de la base de données

Afin de stocker les données de milliers d'étudiants, plusieurs choix se sont offerts à nous. Tout d'abord, nous avons utilisé `std::map`, comparable à un dictionnaire en python, dont les clés ont été l'identifiant de l'élève et les valeurs la struct student. Seulement, cela prenait beaucoup de place en mémoire et nous avons donc délaissé l'idée. Nous avons ensuite essayé la classe vector.

2 Processus vs Threads

Nous avons choisi des threads car ils ont une mémoire partagée, contrairement aux processus

3 Améliorations possibles

3.1 `std::map`

Nous avons comme idée de stocker, dans une structure de données `std::map` (équivalent d'un dictionnaire en Python), les identifiants des étudiants en clés ainsi que leur indice de position dans la base de données en valeurs de ces clés. Selon nous, cela aurait permis de faciliter et d'améliorer l'efficacité de la vérification de si un étudiant est déjà dans la base de données ou non.

En effet, lors de l'ajout d'un étudiant, nous aurions juste à vérifier si l'identifiant est présent dans le map pour éviter d'avoir des doublons dans la base de données.