Mesure de l'impact environemental du code d'OriieFlamme

L'écoconception d'un service numérique consiste à intégrer des contraintes environnementales dans tous les processus de développement, afin de réduire les impacts environnementaux du service numérique pendant son cycle de vie.

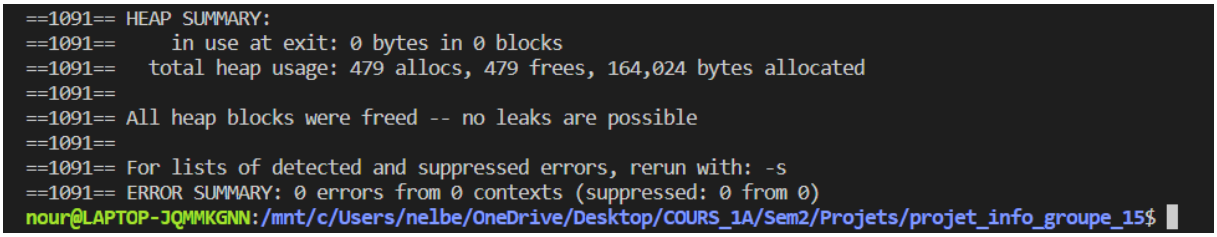
Les impacts peuvent avoir lieu à toute étape du cycle de vie du service : avant son développement (définition des besoins, analyse), pendant son développement (conception, développement logiciel, tests, mise en production) et après son développement (exploitation, maintenance, fin de vie).

Dans la partie qui suit nous nous intéressons à l'impact environemental réalisé lors du développement. Nous notons que ceci peut être fait en utilisant plusieurs outills et en analysant plusieurs méthodes.

Nous débuterons le travail par une analyse dynamique qui permettra de quantifier l'usage des ressources. Nous utiliserons pour ceci **Valgrind**.

1. Valgrind (L'une des bonnes pratiques classiques)

La première étape est de s'assurer qu'il n'y a aucune fuite mémoire et donc pas de gaspillage.
En exécutant notre code on remarque bien qu'il y a autant d'allocs que de free et donc aucune fuite mémoire.



2. Temps d'exécution

L'une des autres bonne pratique de l'éco-conception et l'éco-dévéloppement est de mesurer le temps CPU. Ceci permet de garantir que pas beaucoup de temps ni de ressource sont monopolisé lors de l'execusion. En effet, nous nous intéressons à ce que consomme le processeur et non pas le temps d'exécusion pour l'utilisateur.

J'ai utilisé la commande clock dans le **main** pour pouvoir calculer cette donnée.

3. Optimisation au tout début du développement et qui a déjà été faite :

- Création d'un dépôt git et des branche.
- Assurer l'intégration continue (Gitlab pipeline) sans excès inutiles.
- Création de la documentation