

# Projet Programmation 2023

Télécom Physique Strasbourg: 1A-IR

*Enseignant: Pierre Galmiche*

## 1 Idée générale du projet

Votre objectif lors de ce projet sera d'implémenter un algorithme de recommandation de film basé sur un jeu de données prévu pour une compétition appelée "Netflix Prize". Ce travail vous permettra d'améliorer vos compétences en programmation, mais aussi votre gestion de projet !

## 2 Attendus

Le projet doit s'effectuer **seul ou en binôme**. Les fichiers d'en-têtes doivent tous être documentés. Une documentation globale devra être générée à partir des fichiers source (génération à l'aide de make doc) et la page d'accueil de la documentation doit expliquer rapidement la structure de votre programme, comment il doit être utilisé, et ce qui ne fonctionne pas !! Le code source doit être déposé sur un projet "git" sur le "git de l'unistra". L'enseignant doit être ajouté dans les membres "maintainer" du projet pour y avoir accès. Les codes source doivent être déposés "sur le git" du projet avant le vendredi 12 Janvier 2024, 23h59.

Attention, si vous utilisez du code source trouvé sur internet, vous devez le citer clairement dans le code source et vous devez comprendre ce que vous avez copié (je me réserve le droit de vous poser des questions dessus). La difficulté du projet dépend de la quantité de choses que vous aurez implémentées vous-même et est prise en compte dans la notation. La grille de notation tient compte de la clarté du code, de la structure, des choix d'implémentation, des tests et de la quantité de choses réalisées. Notamment vous devez:

- organiser vos fichiers correctement (dans plusieurs dossiers).
- écrire une documentation (dans les fichiers d'entête).
- rendre un projet qui compile sans erreurs ni warning (en activant tous les warning), et sans erreur valgrind.
- écrire des tests (unitaires avec un programme tests.c et/ou d'exécution avec un test.sh) séparés qui permettent de vérifier que vos fonctions sont correctes.
- écrire un makefile avec des règles appropriées (all, doc, tests).
- prendre en charge le signal SIGINT (qui se produit lors d'un "Ctrl+C")

## 3 Détails sur le projet

### 3.1 Un parser à faire

Votre programme doit parser les fichiers de la base de donnée "Netflix Prize" disponible sur [ce lien](#). Les fichiers étant relativement volumineux, il est fortement conseillé de se "fabriquer" un fichier plus petit possédant les mêmes caractéristiques afin de tester le bon fonctionnement de votre programme. Attention, pensez à mettre un fichier ".gitignore" dans votre projet qui contient le chemin vers les fichiers avec les données, afin qu'ils ne soit pas ajoutés par git. Si un fichier ".txt" apparaît lorsque vous faites un "git status", c'est qu'il n'est pas correctement ignoré.

L'étape de passage du fichier doit permettre de stocker les informations pertinentes du fichier dans des structures (on peut ainsi ignorer beaucoup d'informations). Le passage peut être spécifique au fichier mais il serait préférable que cette étape puisse tolérer des petites variations dans le format, ou en tout cas, afficher des erreurs lisibles en cas de problème de format.

## 3.2 Un format binaire

Une fois le passage terminé, votre programme doit être capable de sauvegarder la structure dans un fichier binaire (indiqué avec une option -o OUTPUT\_FILE) ayant un format bien défini que vous devrez documenter. Cette étape permettra notamment de recharger la structure en mémoire beaucoup plus rapidement que de le faire depuis les fichiers ".txt".

## 3.3 Un algo de recommandation basé sur les statistiques observées

### Quelques statistiques

Une fois la structure en mémoire, vous devez implémenter des algorithmes pour obtenir des statistiques à partir des données sauvegardées. Chaque algorithme est exécuté en donnant une option de ligne de commande. Par exemple :

- l'option -f FOLDER précisera le chemin du dossier où seront sauvegardés les fichiers correspondants aux résultats demandés
- l'option -l LIMITE permettra d'interdire de prendre en compte des notes si leur date est supérieur à LIMITE
- l'option -s FILM\_ID donnera des statistiques sur le film d'identifiant FILM\_ID (nombre de notes, note moyenne ...).
- l'option -c X, Y pourrait permettre de ne prendre en compte que les notes des clients X et Y.
- l'option -b BAD\_REVIEWERS permettra de ne pas prendre en compte les notes de certains clients
- l'option -e MIN permettra de ne prendre en compte que les clients d'élite ayant vu un minimum de MIN films
- l'option -t précisera le temps d'exécution de votre algorithme

### Un algorithme de recommandation

Si la base de votre programme de statistiques fonctionne, vous pouvez le tester avec la partie recommandation. Par exemple, votre algo pourra donner une liste de films recommandés à partir d'une liste de films que l'utilisateur a aimé.

```
"/prog -r "F1,F2,F3,F4,F5,F6" -n 10" donnerais une liste de 10 films qui peuvent intéresser quelqu'un qui aime les films F1-F6. Il faudrait que ça puisse fonctionner si on donne un fichier en entrée: "/prog -r mes_likes.txt -n 10" Avec le fichier mes_likes.txt qui contient:
```

```
F1
F2
F3
...
```

On pourrait même imaginer -a ALGO.DE.RECOMMANDATION pour laisser l'utilisateur choisir son algorithme de recommandation si plusieurs sont implémentés.

Évidemment beaucoup d'exemple ci-dessus sont des extensions possibles, vous devez commencer par **résoudre le problème de base** avant de vous lancer dans des extensions. D'autres options peuvent être ajoutées. Vous pourrez aussi implémenter un affichage graphique afin de visualiser les statistiques faites avec des histogrammes par exemple. Ou encore chercher un film qui contiendrait un mot ou une lettre spécifique dans son titre.

## 3.4 Des tests pour vérifier que tout fonctionne bien

Le projet doit avoir un Makefile avec une règle clean et dont la première règle permet de compiler l'exécutable principal. Un dossier tests doit contenir un Makefile permettant de compiler un programme tests.c qui teste les différentes fonctions que vous avez implémentées. Il pourra aussi y avoir un script test.sh qui exécute le programme sur des fichiers de tests afin de vérifier que la sortie est correcte.