

Projet CY-Météo par Valentin FERNANDES et Erwann DURANEL

Principe de notre projet

Ce projet vise à créer une application qui traitera les données météorologiques d'un fichier pour afficher des graphiques.

Les données traitées peuvent être choisies par l'utilisateur et comprennent les précipitations, la température, l'humidité, la vitesse et la direction du vent, etc.

Un script Shell permettra à l'utilisateur de spécifier les données à analyser, de les filtrer et de créer des graphiques.

Traitement des données

L'utilisateur peut choisir d'afficher plusieurs types de données et sélectionner le lieu et la période pour l'analyse des données.

Le script Shell filtrera les données en fonction des spécifications de l'utilisateur et créera un fichier contenant les données filtrées.

Ce fichier sera ensuite trié à l'aide d'un programme C, qui triera les lignes du fichier grâce à un AVL.

Le programme C triera les informations et les écrira dans un fichier, puis retournera un code 0 en cas de succès ou une valeur positive en cas d'échec.

Création du graphique

Le script Shell va ensuite utiliser les données triées pour générer un diagramme de pression ou de température.

L'utilisateur spécifie les données à représenter sous forme de graphique en passant des arguments au script.

Arguments et options

Le script permet à l'utilisateur de choisir de traiter soit la température, soit la pression atmosphérique.

Au moins une de ces options doit être choisie pour que le script fonctionne, sinon un message d'erreur sera affiché.

Il n'y a pas de limite au nombre d'options qui peuvent être activées.

Les options -t et -p doivent être accompagnées d'un mode : -t<mode> : (t)emperatures, -p<mode> : (p)ressures.

Pour ces deux options, le mode doit être indiquée : 1 : pour avoir en sortie les températures (ou pressions) minimales, maximales et moyennes.

Conclusion

Cette application de traitement des données météorologiques utilise une combinaison de script Shell et de programme C pour traiter et trier les données, puis générer des graphiques en fonction des spécifications de l'utilisateur.

Le script dispose de plusieurs options et arguments pour permettre à l'utilisateur de choisir les types de données et de graphiques qu'il souhaite afficher.

Limitation

Par ailleurs, notre code connaît une limitation majeure. Sur nos machines personnelles, le programme n'aboutit pas et avorte son fonctionnement une 1 fois sur 10 environ.

Nous avons essayé sans succès de régler ce problème en isolant et testant chaque fonction.

Néanmoins, nous pensons qu'il s'agit d'un problème de réallocation de pointeur en cours de processus. En effet, la machine met à disposition une partie de sa mémoire à notre usage exclusif. Cependant, dans notre programme, une variable à la valeur supposée fixée (par notre algorithme) peut en réalité changer. Cela est sûrement le cas lors de déclaration de certains pointeurs.

Cette hypothèse permet d'expliquer pourquoi notre code fonctionne au bout de plusieurs tentatives d'exécution.

Pour observer le résultat de notre code, il suffit d'utiliser plusieurs fois la commande bash.

Répartition du travail

Nous nous sommes reparties le travail au sein du groupe de la manière suivante :

- Chainon Erwann décembre 2022
- Makefile Val décembre 2022
- AVL Valentin décembre 2022 - janvier 2023
- Gnuplot Valentin Erwann janvier 2023
- Gestion fichier Erwann Valentin janvier 2023
- readme Erwann février 2023
- PDF Erwann Valentin février 2023
- Shell Valentin décembre 2022 - février 2023