

Processeur de Données de Réseau Électrique

Cette application analyse les données de réseau électrique à partir d'un fichier CSV pour traiter la production et la consommation d'énergie aux niveaux des postes HVB, HVA, et basse tension. Elle utilise un arbre AVL pour gérer efficacement les données et fournit une sortie triée selon la capacité des stations.

Structure des Répertoires :

- **codeC/** : Contient le code source en C (`cwire.c`), le Makefile pour la compilation.
- **tests/** : Dossier pour les fichiers CSV générés après traitement.
- **tmp/** : Dossier temporaire pour les données intermédiaires, nettoyé après chaque exécution.
- **graphs/** : Dossier prévu pour les graphiques générés (non utilisé dans cette version).
- **.vscode/** : Contient des fichiers de configuration pour Visual Studio Code :
 - **launch.json** : Configure les options de débogage pour l'éditeur.
 - **settings.json** : Contient des réglages spécifiques pour l'environnement de développement.
 - **c_cpp_properties.json** : Configure l'IntelliSense pour aider à la complétion du code et d'autres fonctionnalités.
- **c-wire.sh** : Script shell pour exécuter l'application.
- **c-wire_v00.csv** : le fichier de données d'entrée utilisé par l'application
- **filtered.csv** : utilisé pour stocker une version filtrée des données originales contenues dans le fichier `c-wire_v00.csv`
- **README.md** : Ce fichier de documentation.

Compilation:

1. Naviguez vers le répertoire `codeC` : `cd codeC`
2. Compilez le code C en utilisant `make` : `make`

Ou :

Pour lancer l'application dans un environnement optimal, utilisez Visual Studio Code avec Git Bash. Ouvrez Git Bash via Visual Studio Code et exécutez la commande suivante dans le terminal :

```
./c-wire.sh c-wire_v00.csv hvb comp 1
```

Cette commande initie le traitement des données pour les postes HVB, ciblant les informations pour les entreprises (`comp`) et spécifiquement pour l'identifiant de la centrale 1.

Sortie :

Les résultats sont stockés sous `tests/`, formatés comme `<type_station>_<consumer_type>[_<plant_id>].csv`.

Gestion des Erreurs :

Le script inclut une gestion des erreurs pour les arguments invalides et les erreurs de fichier.

Nettoyage des Fichiers Temporaires :

Le répertoire `tmp/` est nettoyé avant chaque exécution pour garantir que les données précédentes n'influencent pas le processus actuel.

Dépendances :

- GCC pour la compilation du code C.
- Utilitaires Unix standards pour le script shell.
- Visual Studio Code et Git Bash pour une exécution optimale du script.

Ce README fournit toutes les instructions nécessaires pour comprendre et utiliser efficacement le processeur de données de réseau électrique.