

Processeur de données de réseau électrique

Cette application traite les données de réseau électrique à partir d'un fichier CSV pour analyser la production et la consommation d'énergie à différents niveaux (postes HVB, HVA, basse tension). Elle utilise un arbre AVL pour une gestion efficace des données et fournit une sortie triée en fonction de la capacité des stations.

Structure des répertoires :

- `codeC/`: Contient le code source en C (main.c, avl.h) et le Makefile..
- `tests/`: Répertoire où les fichiers CSV de sortie seront stockés..
- `tmp/`: Répertoire temporaire pour les données filtrées (créé pendant l'exécution).
- `graphs/`: Répertoire pour les graphiques (actuellement non utilisé dans cette version).
- `main_script.sh`: Le script principal en shell pour exécuter l'application..
- `README.txt`: Ce fichier.

Compilation:

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

Cela créera l'exécutable `process_data` dans le répertoire `codeC`.

Utilisation :

L'application est exécutée à l'aide du script shell `main_script.sh`.

`./main_script.sh <data file path> <plant type> <consumer type> [plant ID] [-h]`

Arguments:

`<data file path>`: Chemin vers le fichier CSV d'entrée.

`<plant type>`: Type de centrale à analyser (hvb, hva, lv).

`<consumer type>`: Type de consommateur à considérer (comp, indiv, all).

[plant ID]: (optionnel) Filtrez les résultats pour un identifiant de centrale spécifique (de 1 à 5).

-h: Afficher le message d'aide.

Exemple d'utilisation :

Analyser toutes les entreprises connectées aux postes HVB à partir de DATA_CWIRE.csv :

Bash

```
./main_script.sh DATA_CWIRE.csv hvb comp
```

Analysez tous les consommateurs (particuliers et entreprises) connectés aux postes de basse tension (LV) à partir de DATA_CWIRE.csv, filtrés pour plant ID 2:

Bash

```
./main_script.sh DATA_CWIRE.csv lv all 2
```

Output:

Les résultats seront enregistrés dans le répertoire tests/. Les noms de fichiers seront au format <plant_type>_<consumer_type>[_<plant_ID>].csv. Par exemple : hvb_comp.csv, lv_all_2.csv.

Les fichiers output contiennent trois colonnes séparées par des deux-points :

Station ID

Capacité (kWh)

Consommation (kWh)

Les données sont triées par capacité croissante.

Error Handling:

Le script inclut la gestion des erreurs pour les arguments invalides, les erreurs de fichier non trouvé, et les erreurs du programme C. Des messages d'erreur seront affichés dans la console en cas de problèmes.

Temporary Files:

Le script crée un répertoire tmp/ pour stocker les données filtrées temporaires. Ce répertoire est nettoyé avant chaque exécution, y compris l'argument d'identifiant de centrale.

Dependencies:

GCC compiler (for compiling the C code).

make utility.

awk, sort, bc, head, tail, mv, rm, mkdir (standard Unix utilities).