



Informatique
2024-2025
CY-Tech

Karin Attia
Yona Bulard
Ala Kallel

Rapport de Projet : C-Wire



Cette année, le choix du projet nous a été imposé. Le projet choisi par l'équipe pédagogique est un projet nommé C-Wire qui consiste à développer un programme capable de synthétiser et d'analyser les données d'un système de distribution d'électricité en France.

Dès lors, nous avons dû trouver un trinôme afin de pouvoir commencer le projet. Le groupe s'est formé assez naturellement, car nous étions les trois seules filles de la classe. Cela a permis une fluidité du travail, que ce soit en termes de réunions et d'appels pour avancer le projet. Nous avons créé un groupe WhatsApp pour communiquer et s'appeler, mais aussi un notre github partagé où chacun peut modifier le code sans pour autant effacer les traces de ce qu'on fait les autres.

Notre équipe n'a pas nécessairement délégué un rôle à chacune d'entre nous. L'avancement du code s'est fait spontanément. Lors de la première semaine, nous avons surtout tenté de comprendre toute ensemble en quoi consistait le code afin de bien-être sûr de ce que l'on attendait de nous et nous avons commencé à coder quelques structures en C.

La deuxième semaine, étant donné qu'une des trois membres était tombée malade, nous avons tenté de se répartir les tâches avec les deux autres membres du groupe. L'une s'était occupée de faire le programme en C en reprenant les fonctions déjà écrites avec les TD et en commençant à créer les nouvelles fonctions. L'autre s'est occupée de commencer le programme en Shell. La partie en Shell nous a paru être la plus difficile, car c'est un nouveau langage. Il a donc fallu préalablement s'approprier le cours en le lisant plusieurs fois et en refaisant les TD.

Lorsque nous n'étions plus en cours et que le troisième membre du groupe s'était rétabli, chacune a tenté d'avancer le code. Celle qui était malade a alors décidé de continuer le Shell en faisant toute la partie filtrage. Une autre à continuer le programme C et la dernière a fait ce qui est plus annexe, c'est-à-dire ce rapport de projet, le ReadMe et le Makefile.

La troisième semaine, nous avons dû accélérer davantage, car le projet était à rendre en fin de cette semaine. Dès lors, nous avons dû s'octroyer des moments pour pouvoir avancer le projet et tester si notre programme fonctionnait correctement sur les machines de l'école. Mais chacune de son côté continuait aussi le projet chez elle.

Cependant, nous avons fait face à quelques difficultés de réalisation du code. Tout d'abord, le filtrage des données dans le script Shell a été notre première difficulté. En effet, au début, nous ne savions pas quelle commande utiliser pour pouvoir réaliser ce filtrage, mais au final nous avons trouvé la commande **grep** qui est assez efficace. Ensuite, pour la partie en C, bien qu'une grande partie soit déjà

présente dans le cours, les fonctions qui demandent d'utiliser des fichiers nous ont demandé davantage de réflexions, car il nous a fallu replonger dans notre cours de l'année passée afin de nous assurer une bonne réalisation de ces fonctions. Finalement, les points que nous n'avions pas réussis à faire au début sont principalement liés à la gestion des erreurs dans le code. Par exemple, nous n'avions pas réussi à valider correctement les données filtrées avant de les utiliser, ce qui a causé quelques bugs difficiles à corriger. De plus, la gestion des erreurs dans le script Shell n'a pas toujours été optimale. Ensuite, on a eu du mal à récupérer les données filtrées du script Shell depuis le fichier dans le programme C, car nous avons rencontré des problèmes de lecture de fichiers et d'intégration des données, ce qui a ralenti l'avancement du projet. L'optimisation du traitement des fichiers dans le programme C n'avait pas été entièrement réalisée comme nous l'aurions souhaité, ce qui a également provoqué des lenteurs dans l'exécution du programme. Concernant le travail de groupe, le seul souci était de trouver des disponibilités communes entre nous afin de pouvoir effectuer le projet en harmonie.

Présentation des résultats :

Dans le dossier 'test' sur le répertoire github vous pourrez retrouver les résultats finaux pour tous les types de commandes que l'utilisateur peut entrer excepté lorsqu'il indique une centrale (prends trop de place sur github).

Dans le dossier 'test', vous pourrez retrouver un sous-dossier nommé exemple_hvb_comp_2. Il contient plusieurs fichiers avec les résultats et les fichiers intermédiaires si l'utilisateur tape la commande : `./C-Wire-test.sh input/c-wire_v25.dat hvb comp 2`. Un fichier n'a pas pu être mis sur github en raison de sa taille trop importante. Ce fichier se nomme `centrale_filtrees.csv`. Ce fichier contient uniquement des données filtrées qui concernent la centrale n°2, mais ce fichier n'est pas le fichier que l'on envoie au programme C. En effet, un autre fichier davantage filtré va être créé et cette fois-ci, il contient un filtrage plus précis en fonction de la station et du type de consommateur écrit par l'utilisateur. Les fichiers `centrale_filtrees.csv` et `donnees_filtrees.csv` seront mis dans un dossier 'tmp' lors de l'exécution du script. Un autre fichier va être mis dans le dossier 'tmp' nommée `fichier_tmp_result.csv` qui contient les données brutes renvoyées par le programme c, c'est-à-dire que ces données ne sont pas encore triées par ordre croissant. En sortant du dossier 'tmp' vous pourrez trouver un fichier qui se nomme `hvb_comp_2.csv` qui contient cette fois-ci les données triées par ordre croissant. Ce fichier est le fichier final. En fonction de ce que l'utilisateur écrit, le nom de ce fichier

final change comme spécifié dans le document de projet fourni. Le fichier final contient trois colonnes : la première pour les identifiants de station, la deuxième pour la capacité et la dernière pour la consommation.

Lorsque l'utilisateur choisit la station lv et tous les consommateurs (ainsi qu'une centrale s'il le souhaite), cinq autres fichiers temporaires (donc présent dans le dossier 'tmp') seront créés. Vous pourrez en retrouver un exemplaire dans le dossier 'test', dans le sous-dossier "exemple_lv_all" sur notre github . Ils se nomment lv_all_20_selectionnes.csv (qui réunit les 20 postes), lv_all_max(_identifiantdelacentrale)_consommation.csv (extrait les 10 postes avec la consommation la plus élevée, lv_all_min(_identifiantdelacentrale)_consommation.csv (extrait les 10 postes avec la consommation la plus faible), lv_all_trie_consommation.csv (tri par consommation) et lv_all_trie_diff_abs(_identifiantdelacentrale).csv (ajoute une colonne avec le calcul des quantité absolue d'énergie consommée en trop). Finalement, en ressortant du dossier 'tmp' vous trouverez aussi un fichier nommé lv_all_minmax.csv qui regroupe les 20 stations triées par quantité absolue d'énergie consommée en trop (du poste le plus chargé à celui le moins chargé) .

Enfin, dans le dossier 'test' puis dans le sous-dossier 'exemple_lv_all' vous trouverez le graphique bonus demander. Ce graphique est créé à partir du fichier lv_all_minmax.csv. En effet, en abscisse, on peut y retrouver les identifiants des 20 stations. En ordonnées, nous pouvons y voir les capacités en rouge et consommations en vert de ces stations. Lorsque vous voyez apparaître une barre rouge au-dessus d'une barre verte, cela veut dire que la capacité de la station est plus élevée que la consommation. Et lorsqu'il n'y a pas de barre verte, la capacité de la station est moins élevée que la consommation. Dès lors, lorsqu'il y a uniquement une barre verte, il y a surconsommation.

En conclusion, ce projet a été un vrai défi : l'entraide, la motivation et la persévérance ont été la clé de la bonne réalisation de ce projet. Ce travail nous a permis à la fois de consolider nos connaissances en C et en Shell, mais aussi d'apprendre à connaître de nouvelles personnes et d'approfondir certaines compétences en dehors du projet comme le fait de travailler en groupe ou encore de s'organiser.