UNIVERSITÉ TOULOUSE III PAUL SABATIER

L2 CUPGE 2019 /2020

Projet Informatique
Rapport

| EasyShare |

François Marcelon
Théo Arnauld Des Lions

Sommaire

1. Présentation du Projet

1.1. Objectif du projet, présentation.

L'objectif de notre projet EasyShare dans le cadre de l'enseignement Structures De Données, était de nous familiariser à la conception d'un programme, en travaillant à plusieurs.

Il nous a donc fallu réaliser un programme (i.e : "fichiers source et un exécutable testé et opérationnel avec la documentation"), dont le sujet et le cahier des charges sont spécifiés dans la suite du document. Le développement s'est fait - et cela faisait parti des contraintes - sous environnement Linux, et en langage C.

1.2. Cahier des charges.

- Réalisation d'un programme en langage C permettant de gérer un parc de ressources physique (ex : PC, livres, capteurs, ...) ainsi que les prêts de ces ressources.
- Doit contenir au moins deux structures différentes : **Personne** et **Ressource**. Sachant qu'une personne peut demander ou prêter plusieurs ressources et qu'une ressource a un propriétaire et ne peut être empruntée que par une personne à un moment donné.
- Une ressource est créée par une personne.
- Possibilité d'exporter les données prêtées sur une période choisie dans un fichier au format **JSON**.
- Les listes des personnes et des données prêtées doivent être chargées depuis un fichier, et sauvées à la fermeture du programme.
- Le programme doit proposer deux menus différents : **Utilisateur** et **Administrateur.**

1.3. Approche globale

Étant deux fervents utilisateurs de Windows, nous avons installé une **machine virtuelle Linux** sur nos ordinateurs respectifs pour pouvoir développer le programme sous cet environnement.

La centralisation des fichiers et la gestion des versions c'est faite grâce à **GitHub,** dont nous n'étions pas particulièrement familier, mais que nous avons appris à maîtriser au fur et à mesure.

Le confinement général ayant été déclaré au commencement du projet, nous n'avons pu nous entretenir *irl* que quelques fois, l'occasion de nous répartir les tâches et de construire la structure générale du programme. Pour le reste des communications, nous avons utilisé la plateforme **Discord**, pour son accessibilité, et la possibilité de faire des appel audios couplés à des partages d'écran en direct.

2. Mode d'emploi

Le code source du programme peut être récupéré sur le Repository GitHub du projet, où se trouvent la version graphique et la version textuelle.

La version graphique sera compilée avec
gcc -Wall src/EasyShare.c src/gestion_graphique.c src/user.c src/ressource.c -ljson-c $(pkg-config --cflags --libs gtk+-2.0) -o EasyShare.exe

Et exécutée avec :
./EasyShare.exe

Tout cela est récapitulé dans le **Readme.txt**.

3. Organisation du programme et de ses versions

3.1. Version textuelle, fonctions principales.

Dans cette partie nous aborderons les fonctions principales de la version textuelles du programme. Celles-ci servent aussi de base à la version graphique, surtout lorsque se sont des fonction essentielles à la gestion des ressources ou des utilisateurs.

3.1.1. Gestion des utilisateurs

3.1.1.1. Menu connexion, création de compte

Il s'agit de printf() et d'une saisie d'un numéro au clavier pour naviguer à travers les menus.

Création de compte demande un utilisateur vérifie s'il n'est pas existant et s'il n'y a pas de caractères non acceptés ensuite on passe au mot de passe où l'on vérifie s'il n'y a pas de caractères non autorisés. Ensuit cela lance la fonction écriture à la fin du json ressources.

Pour connection en 1 on vérifie si le login existe et ensuite on compare les mots de passe chiffré avec strcmp.

3.1.1.2. Menu utilisateur / administrateur.

Fonctions faisant appelle aux fonctions d'affichage pour ensuite emprunter, supprimer, rendre. Les fonctions du menu admins sont sensiblement les mêmes hormis le faite que ça demande un login quand on veut modifier pour un autre utilisateur quoi que ce soit.

3.1.1.3. Fonctions de recherche.

Les fonctions recherchées sont celles qui recherches dans les fichiers json (les données) car lire l'intégralité du fichier json et l'interpréter en json n'est pas

optimisé avec la librairie json-c d'où le choix de la syntaxe d'un objet json par ligne pour pouvoir interpréter seulement la ligne (tout en satisfaisant que la totalité du fichier reste un fichier json). Ensuite la ligne est transformé en la structure user.

3.1.2. Gestion des ressources.

Création d'une structure ressource pour pourvoir gérer les données plus simplement entre l'interface la saisie et l'écriture dans le json

3.1.2.1. Nouvelle ressource.

Créer la structure et récupère la saisie de l'utilisateur pour l'enregistrer dans la structure puis l'enregistre dans le json

3.1.2.2. Emprunt des ressources.

L'emprunt d'une ressource modifie la ressources en indiquant qu'elle est emprunté inscrit la personne qui effectue le prêt en plus du propriétaire. Et renseigne la date de prêt dans le fichier historique de chaque (du propriétaire et de l'emprunteur).
La fonction rendre et la même dans l'autre sens remet l'état de la ressource (empruntée=0) modifie l'historique pour inscrire quand la ressource a été rendu.

3.2 Version graphique, fonctions principales

Pour cette partie, nous nous intéresserons à la version graphique du programme. Alors que la version textuelle remplissait très bien le cahier des charges, nous avons décidé d'utiliser le temps supplémentaire à notre disposition pour améliorer sensiblement notre programme. Nous avons alors décidé de mettre en place un version graphique. Elle s'appuie grandement sur les fonctions de la version textuelle pour fonctionner, et met en place grâce à la librairie **GTK** des fenêtres, menus, entrées textes, boutons, etc ...

3.2.1. Menus

Dans tous les menus, l'utilisateur peut retourner en arrière ou quitter le programme via la petite croix de la fenêtre.

3.2.1.1. Menu Accueil

Le Menu Accueil est le premier menu à s'afficher lors du lancement du programme. Il affiche le logo de notre projet, son nom, et trois boutons qui permettent la navigation.

Le nom du projet s'affiche en gras et en police Sans. En effet, GTK permet l'utilisation de balises **span** dans ses *labels* (étiquettes).

Deux des boutons permettent respectivement de se connecter et de créer un compte, que je traiterais plus en détail dans la suite de ce rapport.

3.2.1.2. Menu utilisateur / administrateur

Les menus principaux utilisateur et administrateur sont les deux plus gros menu du programme. Ils permettent d'accéder aux fonctionnalités essentielles de celui ci, c'est à dire la création, gestion, et emprunt de différentes ressources par les utilisateurs. Les administrateurs, quant à eux, pourront consulter la totalité des ressources mises en place par les utilisateurs et éventuellement les supprimer, ainsi que parcourir la liste des

utilisateurs, voir l'historique global des échange et bannir un utilisateur si besoin en est.

Ces menus affichent aussi toujours le nom de l'utilisateur présentement connecté.

3.2.2. Fonction d'acquisition.

3.2.2.1. Connexion, Création de compte.

Les fonctions de connexion et de création de compte utilisent des entrées de textes pour fonctionner. L'utilisateur rentre alors son nom et son mot de passe qui sont comparés à ceux présent dans nos fichiers de sauvegarde. Lors de la connexion, ce sont le nom et le mot de passe (préalablement chiffré) qui sont comparés. Pour la création de compte, on vérifie seulement que le nom choisi par l'utilisateur n'est pas déjà pris. Un utilisateur qui créer un compte doit se connecter après, il ne l'est pas automatiquement.

Si les informations fournies lors de la connexion vérifient des identifiants que nous avons stockés, alors on affiche le menu utilisateur ou administrateur en fonction de l'identité de l'utilisateur.

3.2.2.2. Création, gestion des ressources.

Les menus de gestion des ressources sont tous très spécifique et assez limités. En effet, une seule action est possible par menu. Dans le menu de suppression de ressource par exemple, l'utilisateur affiche la liste de ses propres ressources sous la forme d'une succession de boutons *Radios*. C'est à dire qu'il ne peut sélectionner qu'une seule ressource à supprimer à la fois. De même pour l'administrateur, même si ce dernier a accès à la liste de toutes les ressources disponibles dans EasyShare.

Et il en est de même lors de l'emprunt d'une ressource. L'utilisateur fera d'abord le tri en choisissant le type de ressource qu'il veut emprunter (celles ci lui seront présentées via une fonction de recherche par type de ressources) et ensuite la ressource qui l'intéresse parmis la liste qui se présente à lui.

4. Bilan

En conclusion, ce projet nous a permis d'aborder la réalisation complète d'un programme, à plusieurs, et sous la contrainte d'une deadline.

Si nous ne nous connaissions pas vraiment au début, nous avons appris à travailler ensemble et à se compléter dans la manière de travailler et de coder.

Il nous aura aussi, et surtout, permis d'évaluer nos connaissances actuelle en c, nos limites, et nos capacités à apprendre de nouveau outils pendant même le développement.

Notre volontée de rendre un programme graphique nous à pris beaucoup de temps et au final nous ne rendons pas le projet ficelé et parfaitement abouti.

Nous sommes cependant fier de ce que nous avons pu tirer du travail (acharné) de ses dernières semaines.