

Compte Rendu

Langage C

Eliot Midavaine – Simon Defontaine
1^{ERE} ANNÉE LICENCE INFORMATIQUE

Table des matières

Introduction	2
Description du programme	2
Structure Parking	2
Fonction « Load Parking »	3
Fonction « Affiche Parkings »	4
Fonction « Affiche Parking »	5
Fonction « Choix Parking »	5
Fonction « Entrée Parking »	6
Mode Administrateur	7
Affichage	8
Expérience personnelle	8
Conclusion	8

Introduction

Le programme se nomme Where's my parking.

Ce projet est un gestionnaire de parking de la métropole lilloise. Il permet de gérer les parkings, d'améliorer l'expérience des usagers et d'optimiser l'utilisation des espaces de stationnement.

Le programme utilise un fichier au format csv qui contient la liste des parkings de la métropole lilloise et toutes leurs informations, les données sont lues et permettent grâce à l'appel de plusieurs fonctions de traiter les entrées et sorties des véhicules dans les parkings et de garantir une gestion fluide et précise des données.

Nous avons également rajouté un mode administrateur (Question Bonus) qui permet de superviser et d'éditer les informations des clients et des parkings.

Pour l'interface graphique nous avons utilisé Ncurses et avons opté pour un style graphique rappelant les BIOS des vieux ordinateurs d'époque ce qui donne à notre programme un aspect « Old School ».

Description du programme

Structure Parking

Les parkings sont traités à l'aide d'une structure préalablement définie

```
struct parking {  
    char id[8];  
    char nom[50];  
    char adresse[50];  
    char ville[25];  
    char etat[10];  
    int place_dispo;  
    int capacite_max;  
    char date_maj[26];  
    char affichage[10];  
};
```

Nous avons découpé la structure pour chaque élément du fichier CSV et avons donné à chaque élément une taille permettant de prendre en charge jusqu'à la plus grande valeur de cette catégorie.

Par exemple, l'ID est une chaîne de caractère de taille 8 fixe donc sa taille sera [8].

Autre exemple, le nom le plus long faisant environ 30 caractères, nous avons rajouté de la marge en cas d'ajout d'un nouvel élément dans le fichier.

Fonction « Load Parking »

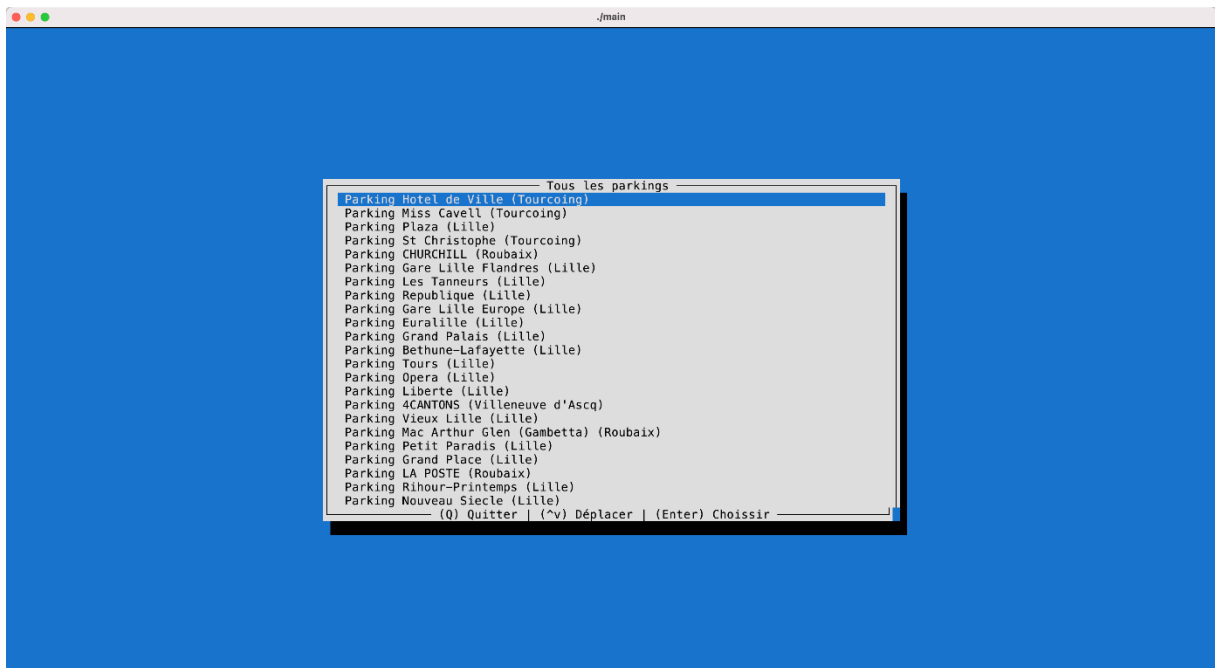
La fonction « Load Parking » lis le fichier CSV et de créer les structures correspondantes

```
void load_parking(FILE *fptr, struct parking *parkings, int nb_of_rows) {
    fseek(fptr, 0, SEEK_SET);
    while (fgetc(fptr) != '\n');
    for (int i = 0; i < nb_of_rows; ++i) {
        char ligne[250];
        fgets(ligne, 250, fptr);
        struct parking parking_temp;
        struct parking parking_traite;
        strcpy(parking_traite.id, strtok(ligne, ";"));
        strcpy(parking_traite.nom, strtok(NULL, ";"));
        strcpy(parking_traite.adresse, strtok(NULL, ";"));
        strcpy(parking_traite.ville, strtok(NULL, ";"));
        strcpy(parking_traite.etat, strtok(NULL, ";"));
        parking_traite.place_dispo = atoi(strtok(NULL, ";"));
        parking_traite.capacite_max = atoi(strtok(NULL, ";"));
        strcpy(parking_traite.date_maj, strtok(NULL, ";"));
        strcpy(parking_traite.affichage, strtok(NULL, ";"));
        parkings[i] = parking_traite;
    }
}
```

La fonction prend en entrée le pointeur vers un fichier CSV, une liste de structure parkings ainsi que le nombre de structure parking a remplir.

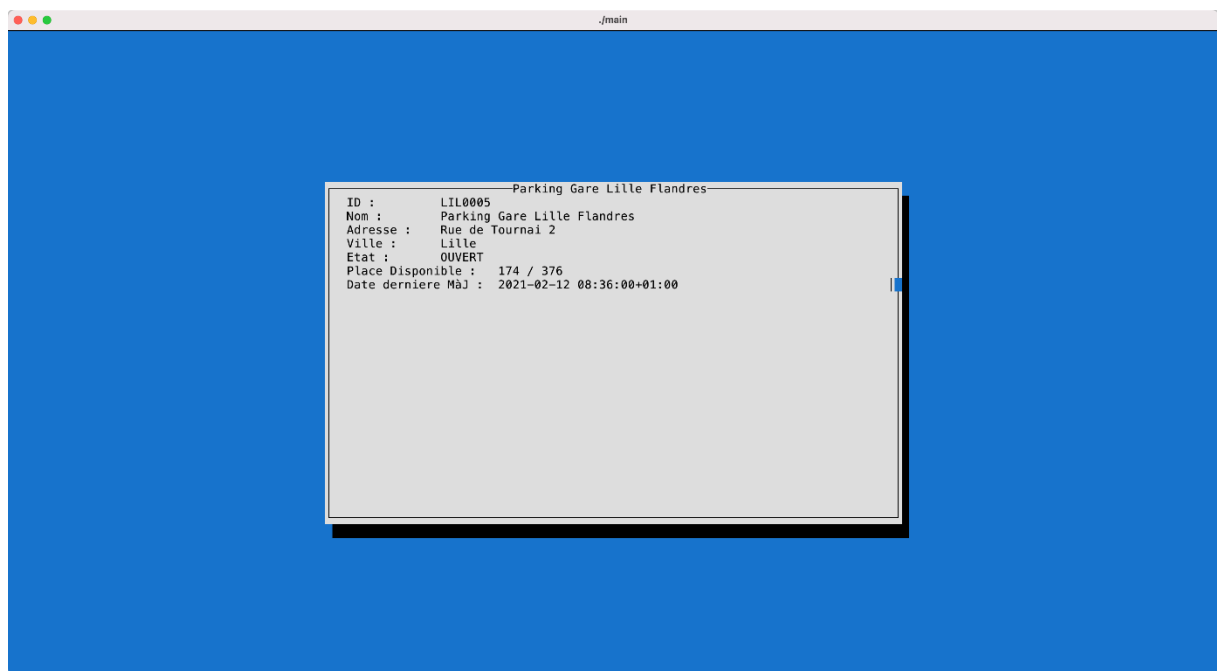
La fonction remplit ensuite la liste de parkings.

Fonction « Affiche Parkings »



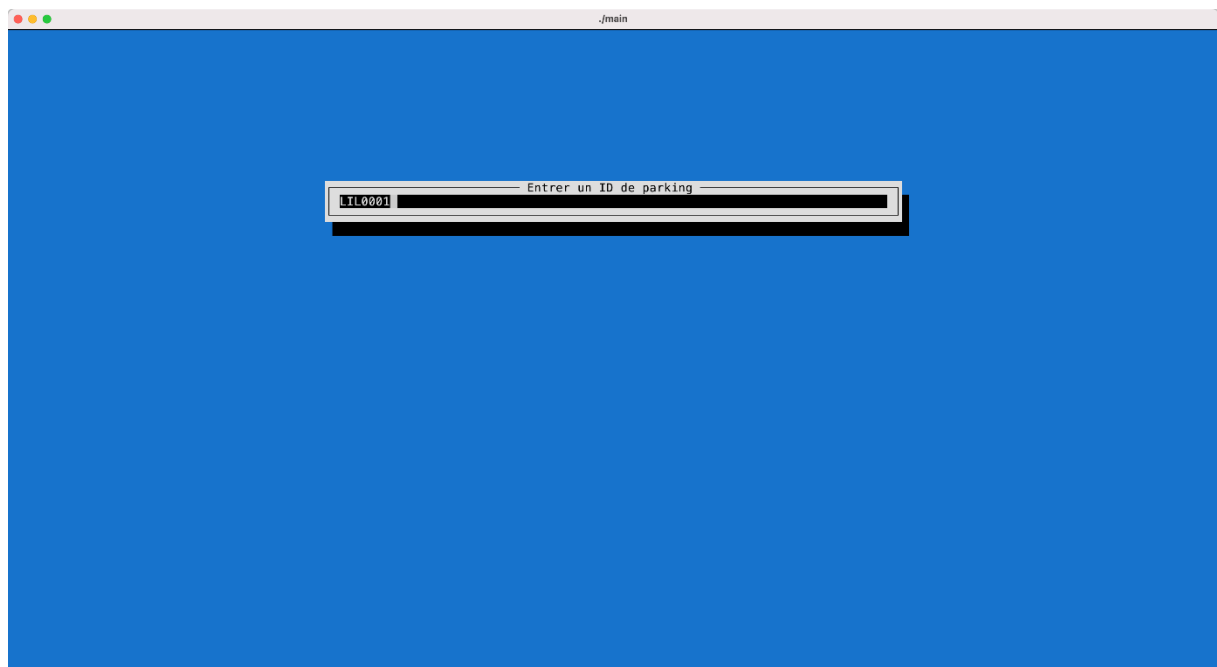
La fonction « Affiche Parkings » affiche la liste des parkings en ne montrant que leurs noms et leurs localisations pour plus de lisibilité.

Fonction « Affiche Parking »



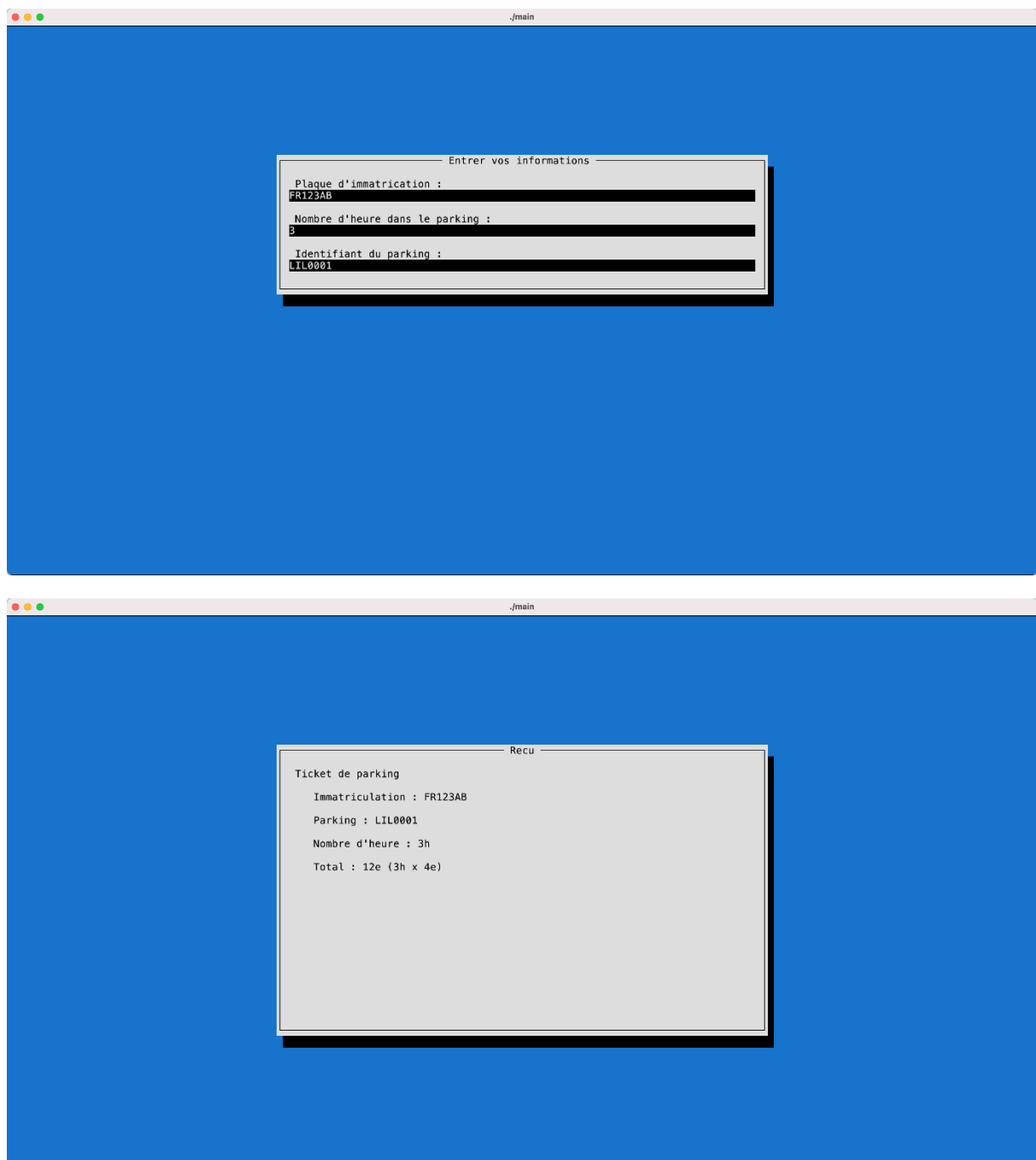
La fonction « Affiche Parking » affiche toutes les informations d'un parking sélectionnée dans la liste de la fonction précédente.

Fonction « Choix Parking »



La fonction « Choix Parking » appelle la fonction « Affiche Parking » selon le parking correspondant à l'ID rentré dans la barre de recherche

Fonction « Entrée Parking »



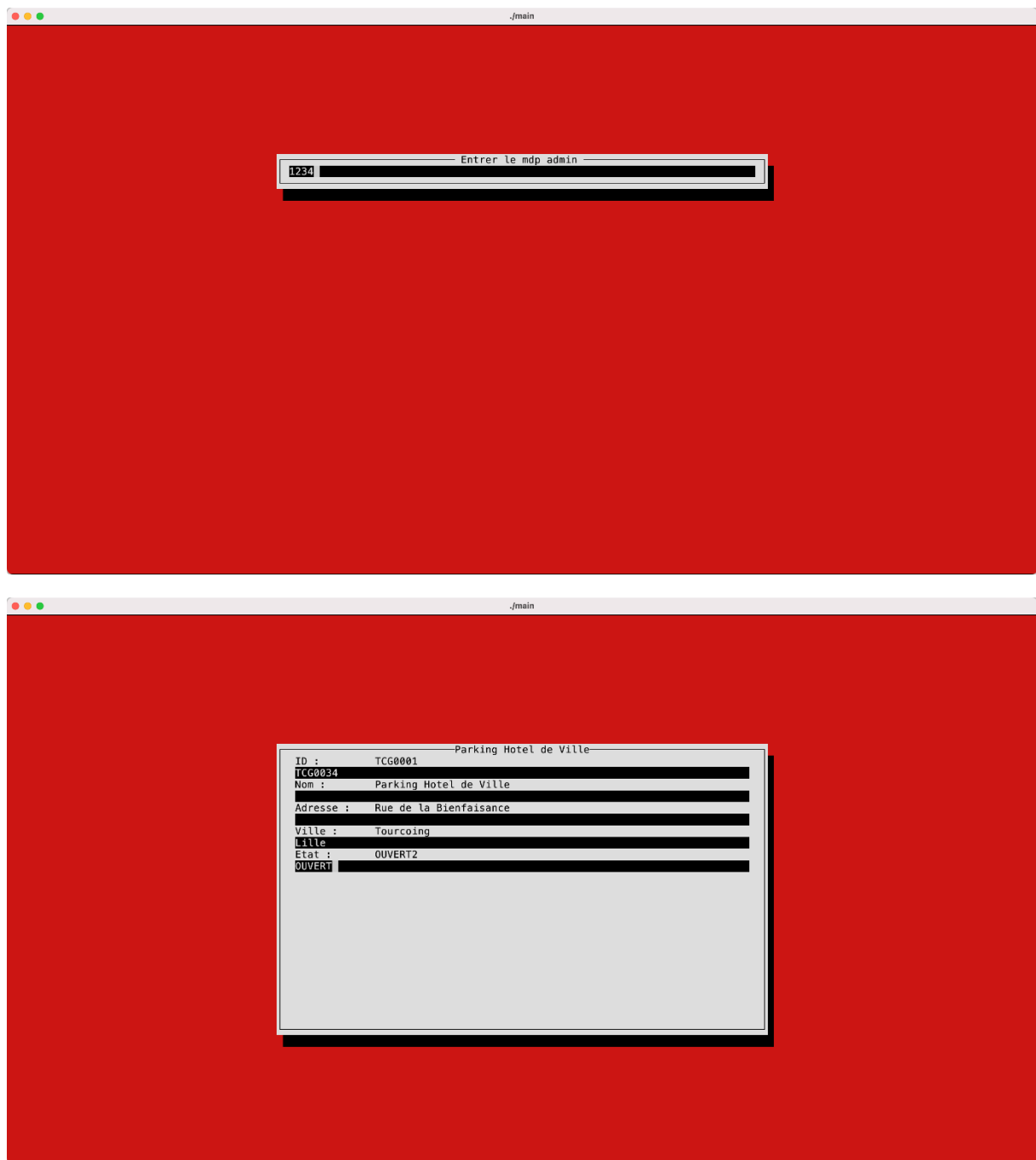
The first screenshot shows a terminal window titled `./main` with a blue background. A white dialog box titled "Entrer vos informations" is centered. It contains three input fields: "Plaque d'immatriculation :" with the value "FR123AB", "Nombre d'heure dans le parking :" with the value "3", and "Identifiant du parking :" with the value "LIL0001".

The second screenshot shows the same terminal window with a white dialog box titled "Recu". It displays the following information:

```
Ticket de parking
Immatriculation : FR123AB
Parking : LIL0001
Nombre d'heure : 3h
Total : 12e (3h x 4e)
```

La fonction « Entrée Parking » demande notre plaque d'immatriculation, notre nombre d'heure dans le parking et l'identifiant du parking dans lequel nous voulons aller et qui nous affiche notre reçu avec le prix à payer puis la fonction enregistre le parking choisi dans un fichier texte log.

Mode Administrateur



Le mode administrateur peut être activé si on entre le bon mot de passe (Défini en tant que constante dans le programme) et permet la modification de tous les attributs d'un ou plusieurs parkings choisis.

Affichage

Pour l'affichage, nous avons utilisé la bibliothèque Ncurses (comme demandé dans la première version du sujet) et avons opté pour un style rappelant les vieux BIOS des ordinateurs de l'époque.

Le fond est bleu pour le mode utilisateur et est changé en rouge pour le mode administrateur.

Expérience personnelle

La difficulté principale rencontrée était l'utilisation de Ncurses avec tout ses problèmes d'affichages. Sinon dans l'ensemble, le programme était assez simple et ne posait pas trop de problème. Nous avons trouvé ce travail intéressant et cela a permis d'améliorer nos connaissances en Langage C.

Conclusion

Le programme respecte au maximum le cahier des charges malgré quelques divergences notamment due au changement brusque de sujet au cours du semestre mais nous avons essayé de se coller au mieux au deuxième sujet avec les bases de ce que nous avons déjà fait avant ce dit changement.