Antoine Breton
Arnaud Dupeyrat
Groupe 4

Spécification TP programmation concurrente
Gestion d'un parking automobile

Document de spécification et conception

Table of Contents

Antoine Breton
Arnaud Dupeyrat
Groupe 4

Introduction

Parking est une application de modélisation d'un parking automobile. Elle peut être divisée en deux parties indépendante : le **gestion du parking** (gestion des places, des priorités, etc) et la **simulation** (arrivage de voitures). Pour une future utilisation de l'application, la partie **simulateur** pourra simplement être remplacée par de vrai capteurs de parking. Voici une rapide description de l'application :

- Elle effectue la gestion des entrées et des sorties de voiture,
- Le parking est composé de 8 places et possède 3 barrières d'entrée et 1 barrière de sortie.
- Les touches clavier permettent de simuler l'arrivée ou la sortie des voitures
- Le programme nous oblige à manipuler des processus, pour pouvoir parvenir à faire travailler simultanément l'ensemble des taches qui composent notre programme.
- Si le parking est plein, la prochaine voiture est choisi en prenant en compte une priorité défini par le cahier des charges.

Nous avons remarqué qu'une des faiblesses du simulateur est bien sur le fait qu'elle n'est ne reflète pas la réalité en certains points. Ainsi, on ne peux pas simuler l'arrivée de deux voitures à deux portes différentes au même moment.

Séparation du travail

Comme demandé dans le cahier des charges, nous avons séparé les différents modules de la manière suivante :

Antoine Breton	• Tache Mère • Tache Sortie
Arnaud Dupeyrat	• Tache Entrée • Tache GestionClavier
Commun	• IPC • Config

Etat de la réalisation

Le programme rendu réalise l'ensemble des fonctionnalités comprises dans le cahier des charges. Bien qu'elles soient satisfaisantes dans le cadre d'une utilisation normale, on pourrait travailler sur l'amélioration de l'application. Un autre axe d'amélioration serait d'améliorer la généricité, en permettant par exemple l'utilisation d'un nombre arbitraire de barrières d'entrée et de sortie.

Antoine Breton
Arnaud Dupeyrat
Groupe 4

Changement vis à vis du de la conception initial

Après le rendu du schéma d'architecture générale de l'application multitâche, nous avons apporté des modifications mineures :

Premièrement, entre la tache Sortie et GestionClavier la structure échangé a été modifié. En effet dans la schéma général, la structure Voiture permettait de transmettre des informations, cette solution bien que soit fonctionnel s'est montré gourmande en mémoire. Nous avons opté pour échanger une structure contenant seulement le numéro de place afin de réduire les données inutile.

Deuxièmement, nous avons opté pour deux mémoire partagé dont le contenu n'était pas précisé entre le processus BarriereEntree et Sortie :

- Une contenant les informations concernant le parking
  - L'état du parking (l'ensemble des voitures garées à leur place respective)
  - Le nombre de places libres
- Une contenant les informations concernant les requêtes effectuées
  - Le nombre de requêtes d'entrée en cours et les requêtes elles-mêmes.

Antoine Breton
Arnaud Dupeyrat
Groupe 4

Structure Utiles au programme

L'application nous a obligé, à manipuler plusieurs structures, En voici le détail.

- **Voiture** : *Utilisée par les Boites au lettre des entrées et par la mémoire partagé MemParking*
  long type;
  char typeUsager;
  unsigned int numeroMineralogique;
  time_t heureArrive;
  time_t heureDepart;
  unsigned int numeroEmplacement;

- **NumeroEmplacement :** *Utilisée par dans la Boite au lettre de la Sortie*
  long type;
  unsigned int numeroPlace;

- **Requete :** *Utilisée par la mémoire partagé MemRequete pour contenir les requêtes*
  time_t heureArrive;
  char typeUsager;

- **MemRequete :** *Mémoire partagée qui contient les requêtes d'entrée.*
  Requete requetes[3];

- **MemParking :** *Mémoire partagée qui contient l'état courant du parking*
  int nbPlace;
  Voiture voitures[8];