



Gestion d'un parking automobile MULTITACHE en C/C++ sous Linux

Département Informatique

ANTOINE Kevin
COURJAULT Amaury
B3428

Introduction

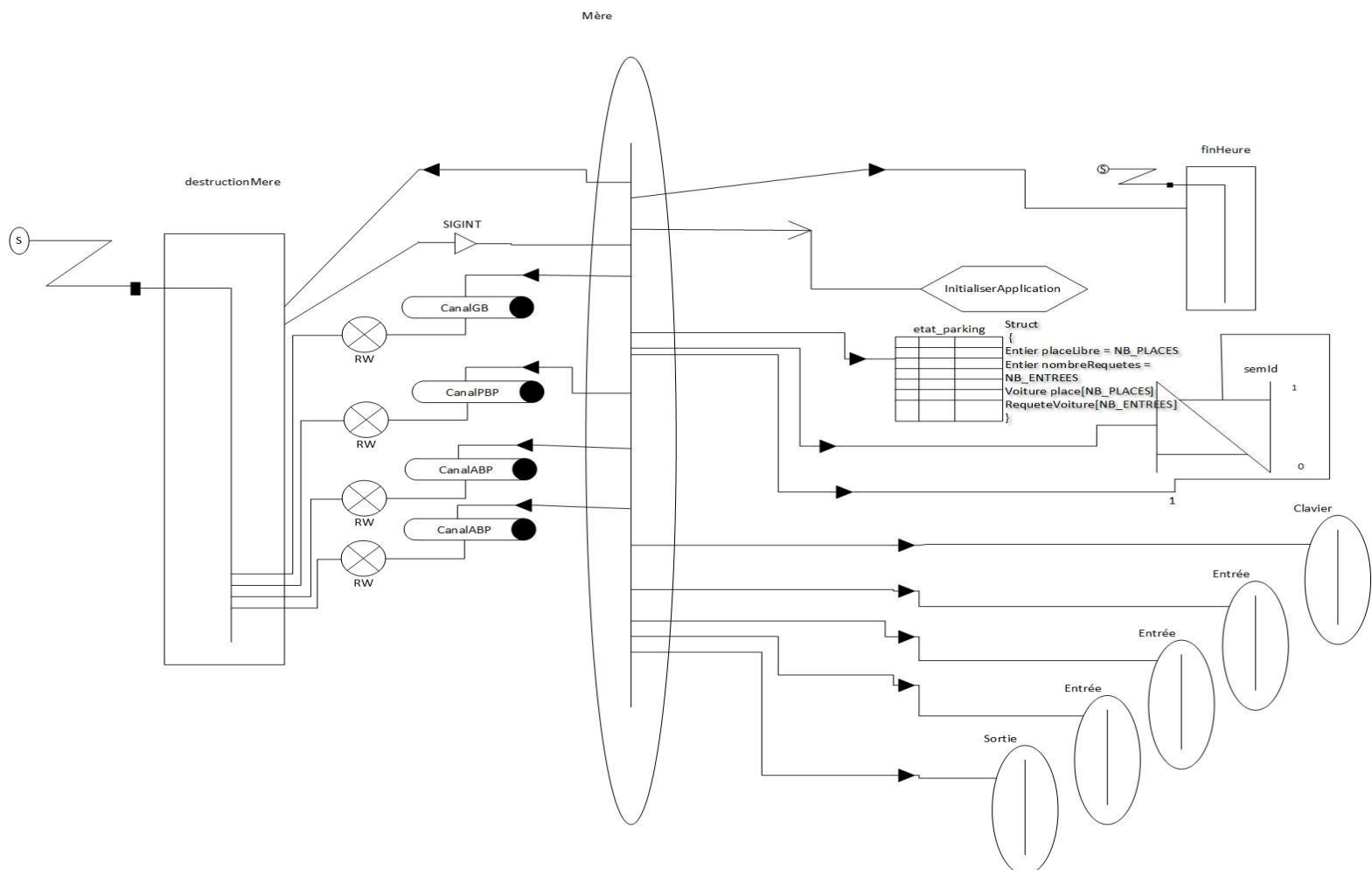
Dans ce TP, nous devons mettre en oeuvre les mécanismes de la programmation concurrentielle pour effectuer la simulation de la gestion des flux d'entrée et sortie dans un parking automobile.

Pour cela, nous nous sommes aperçu qu'il était nécessaire de prévoir une conception "en béton" avant de commencer à fournir des lignes de code.

Conception Générale

A / Tache Mère

A.1 Initialisation Mere :



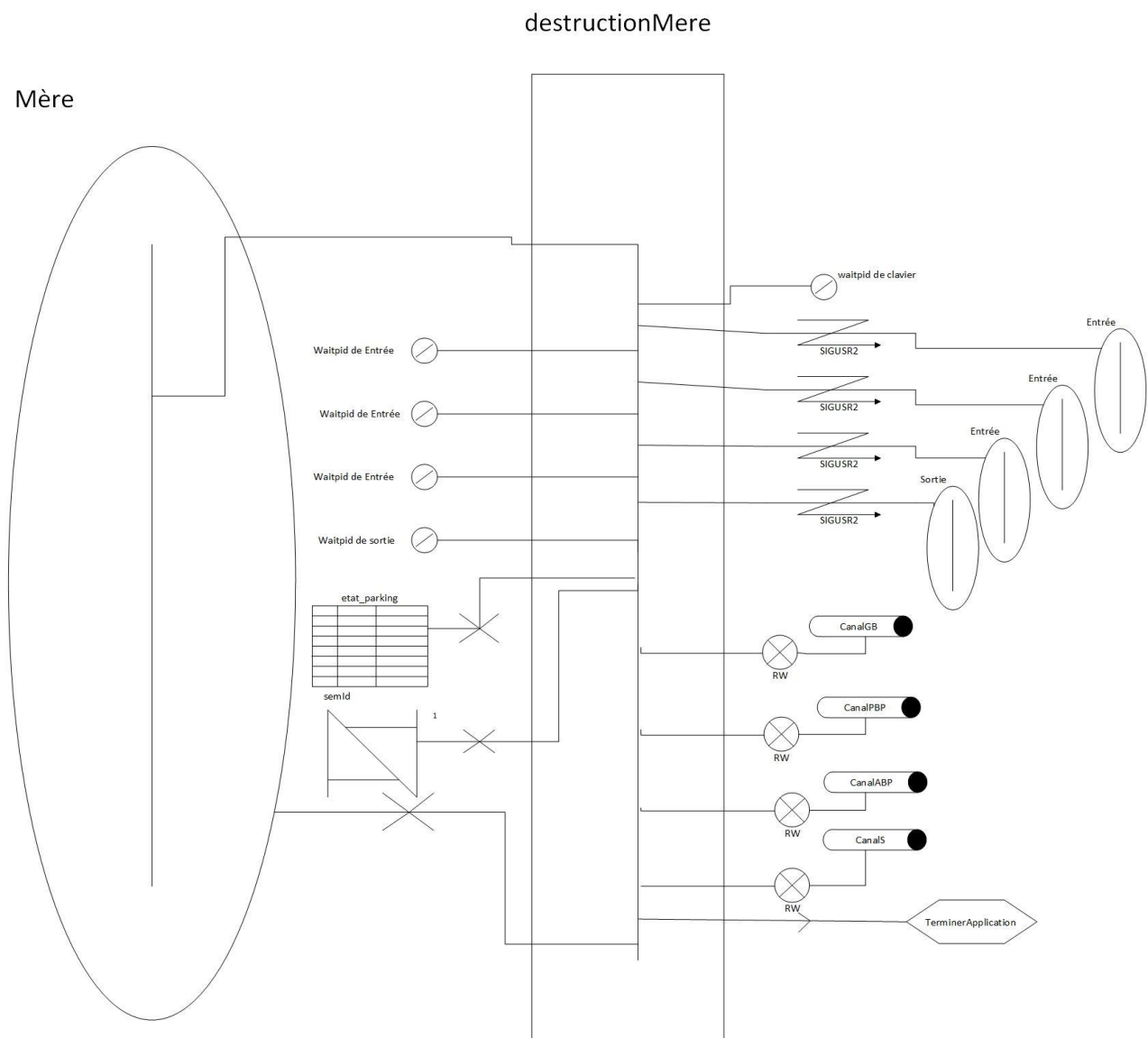
CanalGB est créé par la tâche Mère. Clavier l'utilise en écriture et une tâche Entrée, celle associée à l'entrée Gaston Berger, l'utilise en lecture. Le contenu de ce canal est une RequeteVoiture (structure construite par nos soins).

CanalPBP est créé par la tâche Mère. Clavier l'utilise en écriture et une tâche Entrée, celle associée à l'entrée des professeurs du coté Blaise Pascal, l'utilise en lecture. Le contenu de ce canal est une RequeteVoiture (structure construite par nos soins).

CanalABP est créé par la tâche Mère. Clavier l'utilise en écriture et une tâche Entrée, celle associée à l'entrée des autres du coté Blaise Pascal, l'utilise en lecture. Le contenu de ce canal est une RequeteVoiture (structure construite par nos soins).

CanalS est créé par la tâche Mère. Clavier l'utilise en écriture et la tâche Sortie l'utilise en lecture. Le contenu de ce canal est une RequeteVoiture (structure construite par nos soins).

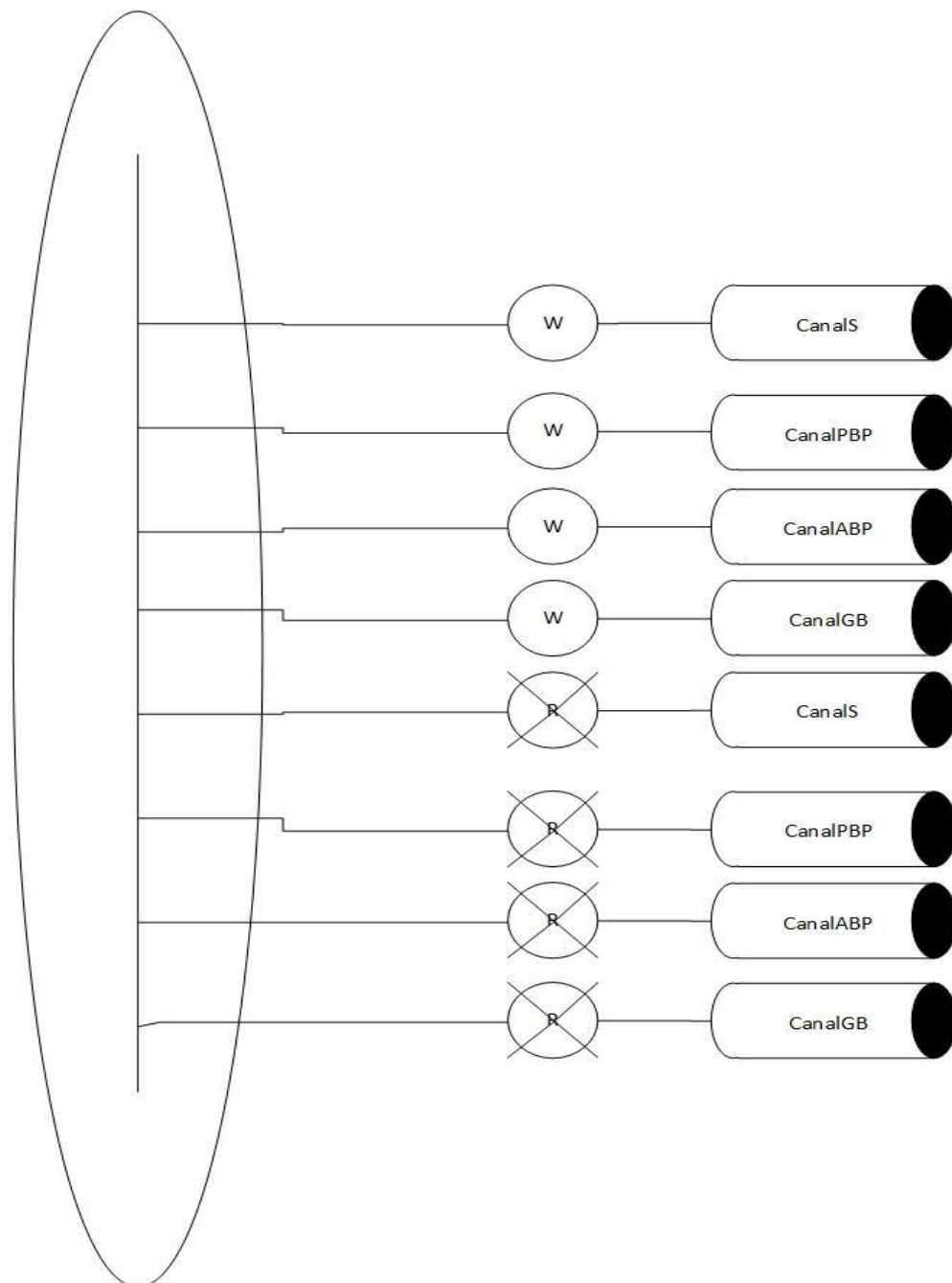
A.2 Destruction Mere:



B/ Tache Clavier

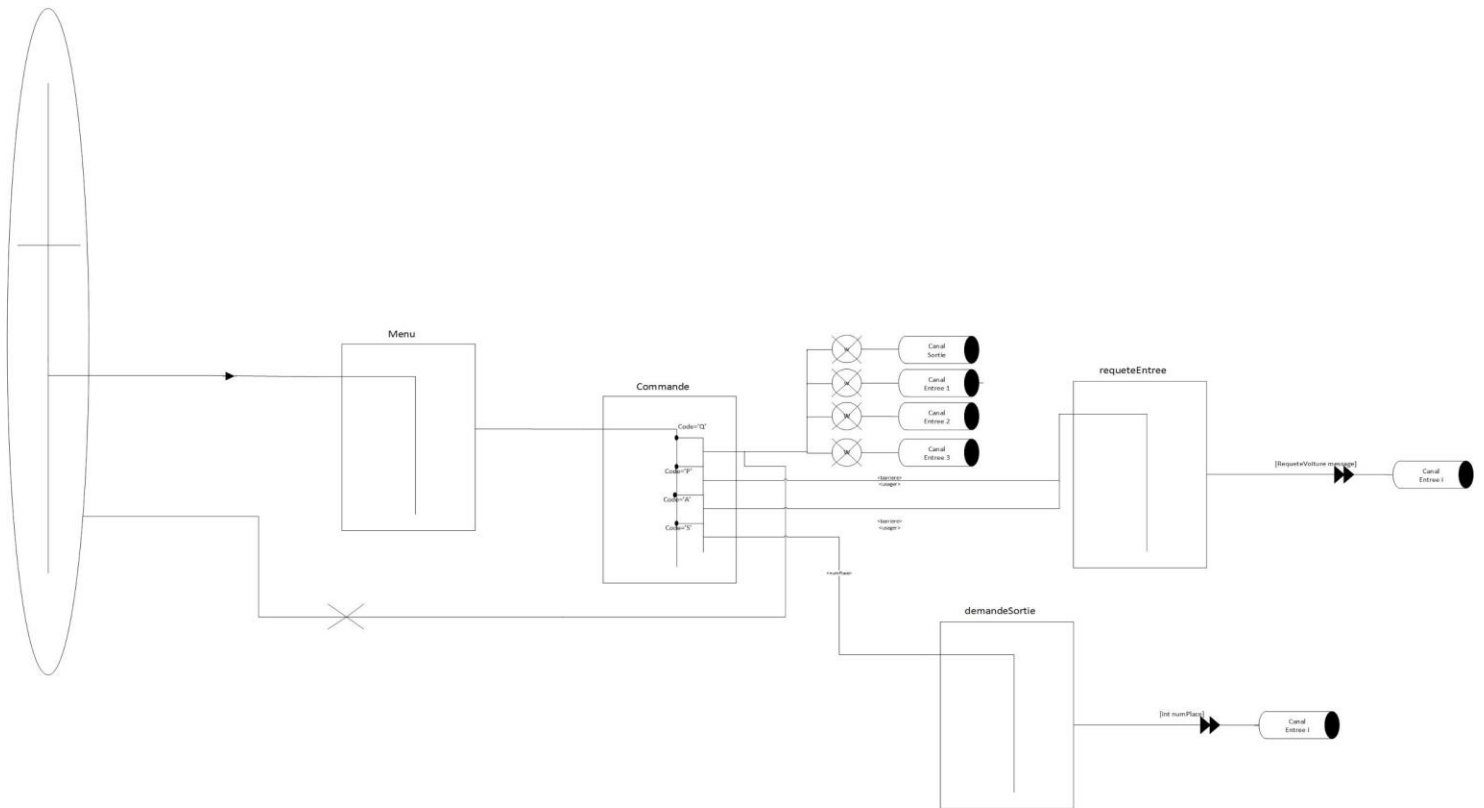
B.1 Initialisation Clavier:

Clavier



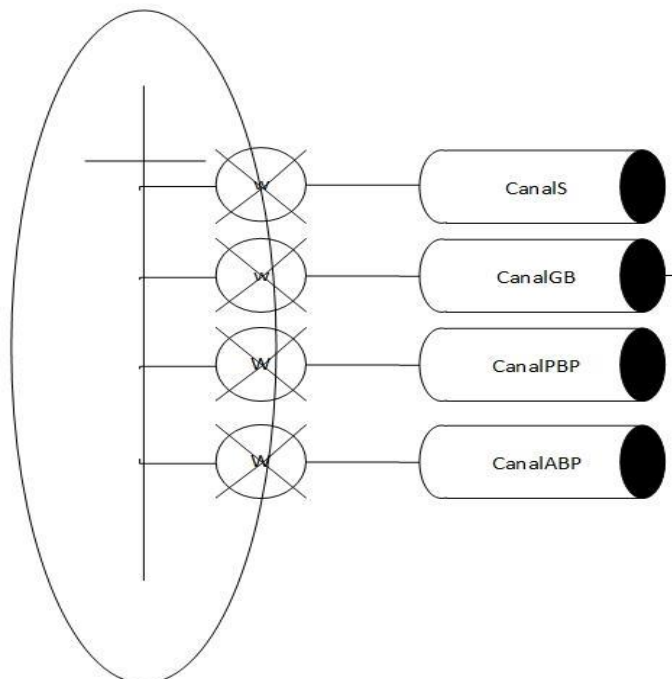
B.2 Moteur Clavier

Clavier



B.3 Destruction Clavier

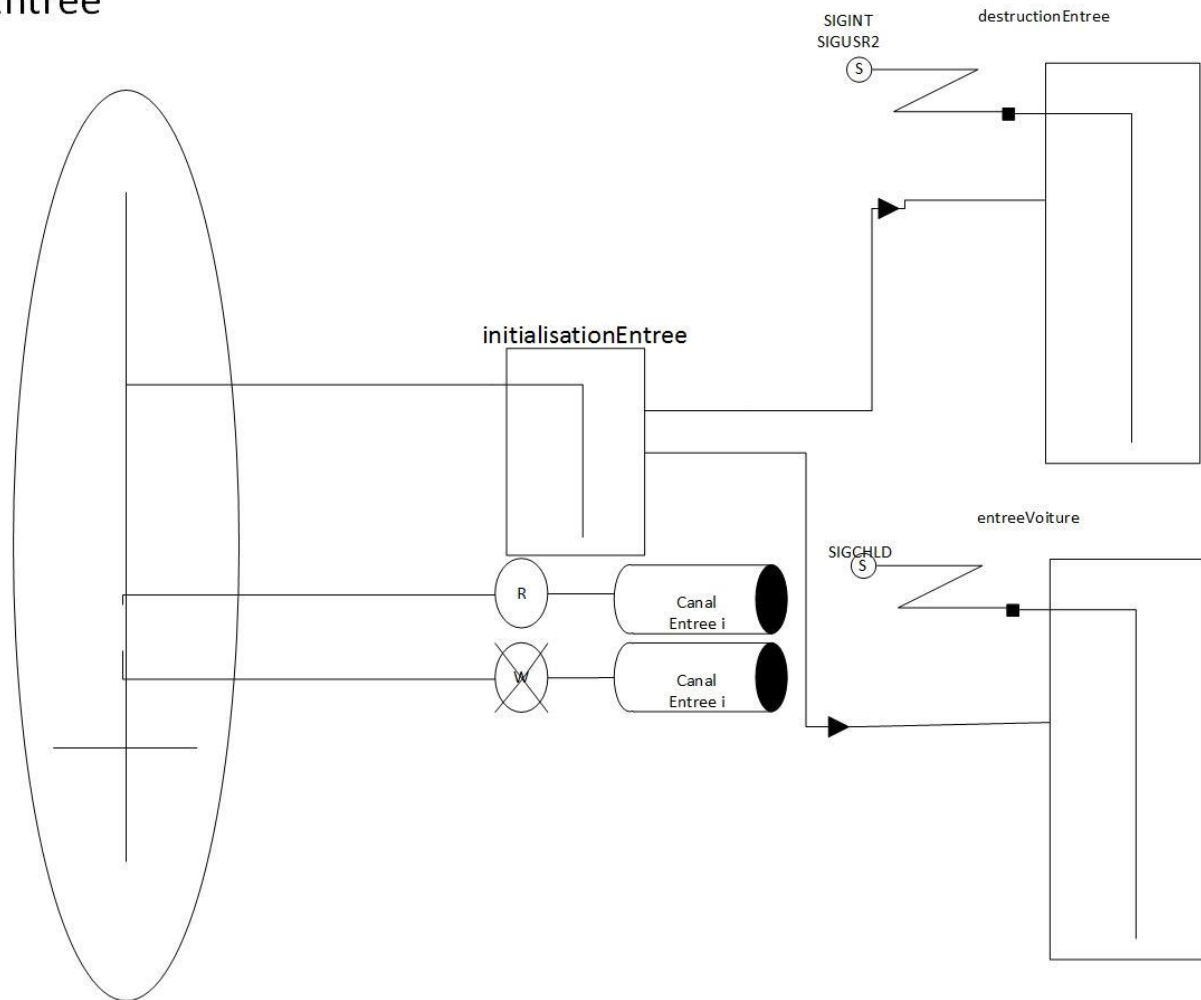
Clavier



C/ Tache Entree

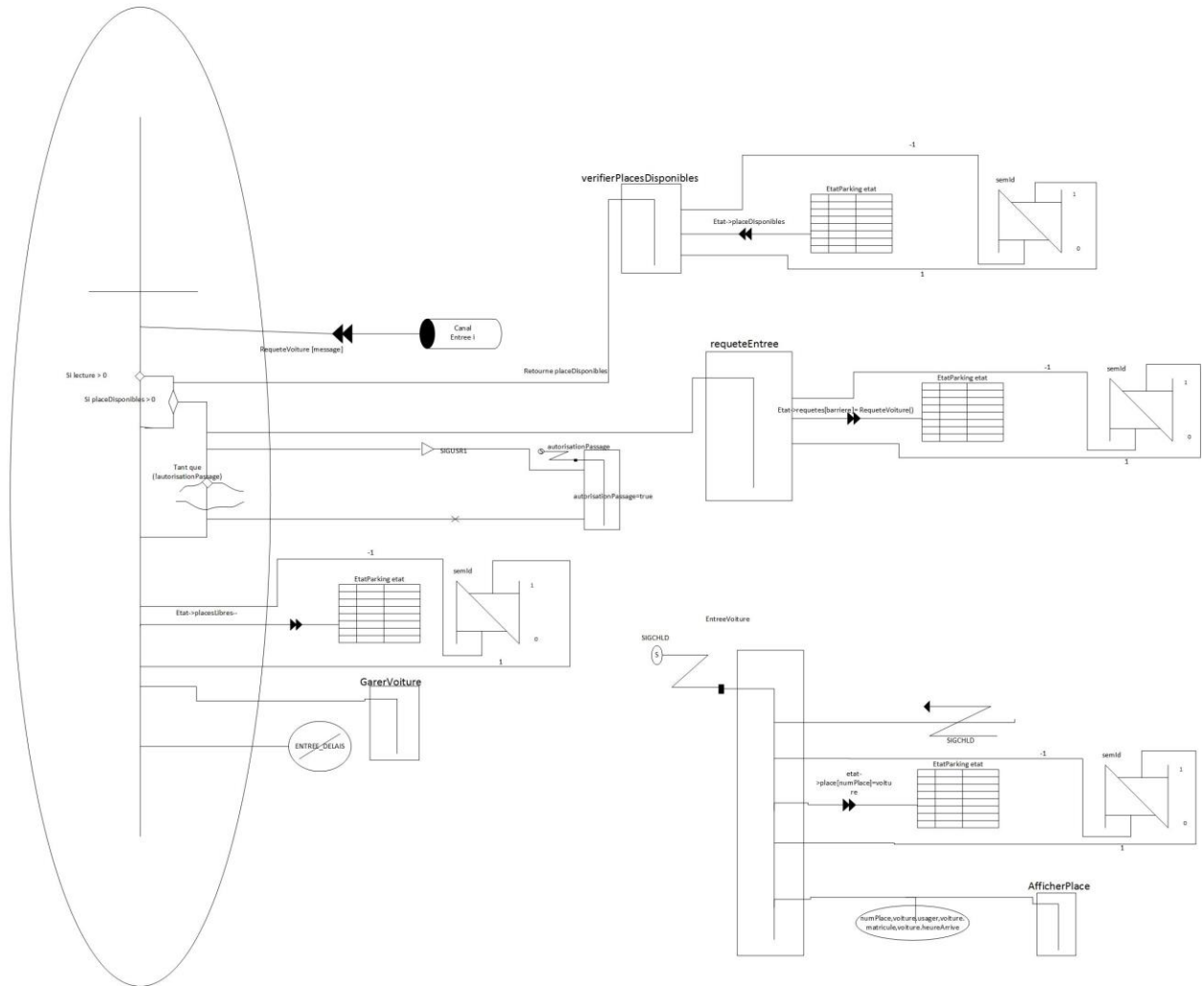
C.1 Initialisation Entree

Entree

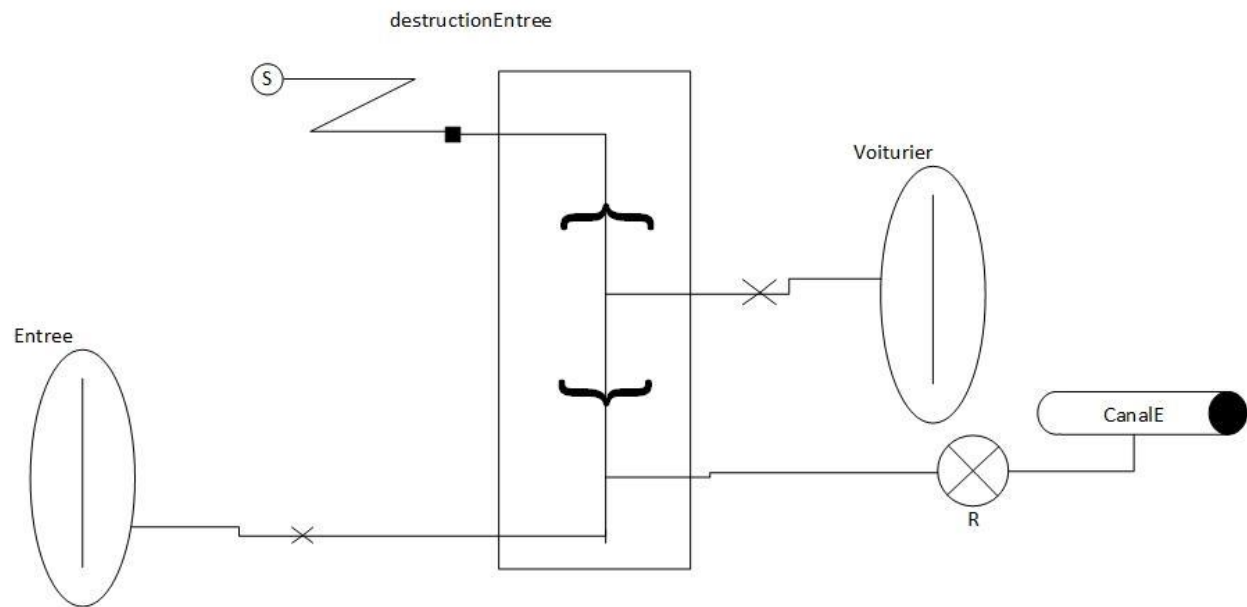


C.2 Moteur Entree

Entree



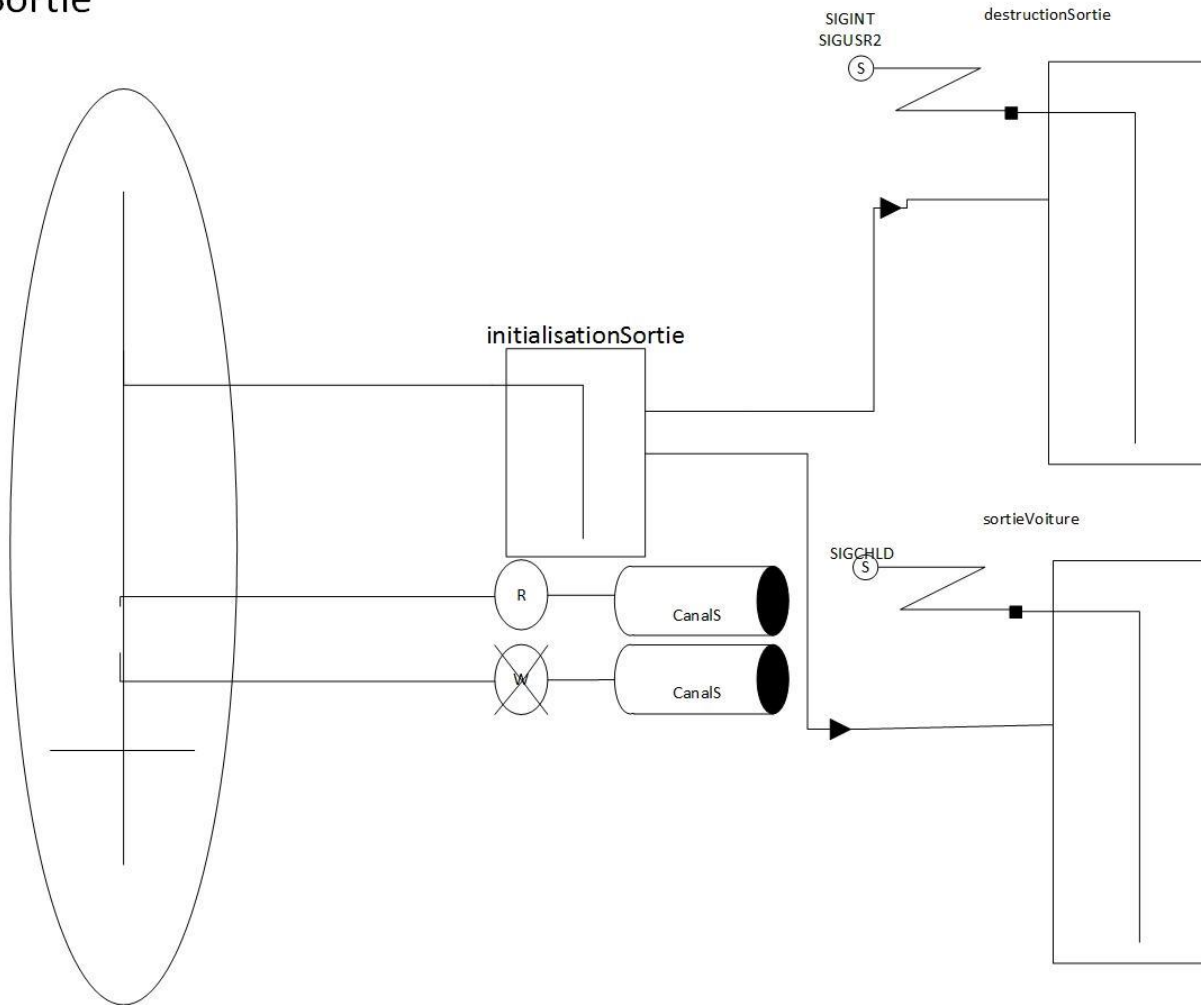
C.3 Destruction Entree



D/ Tache Sortie

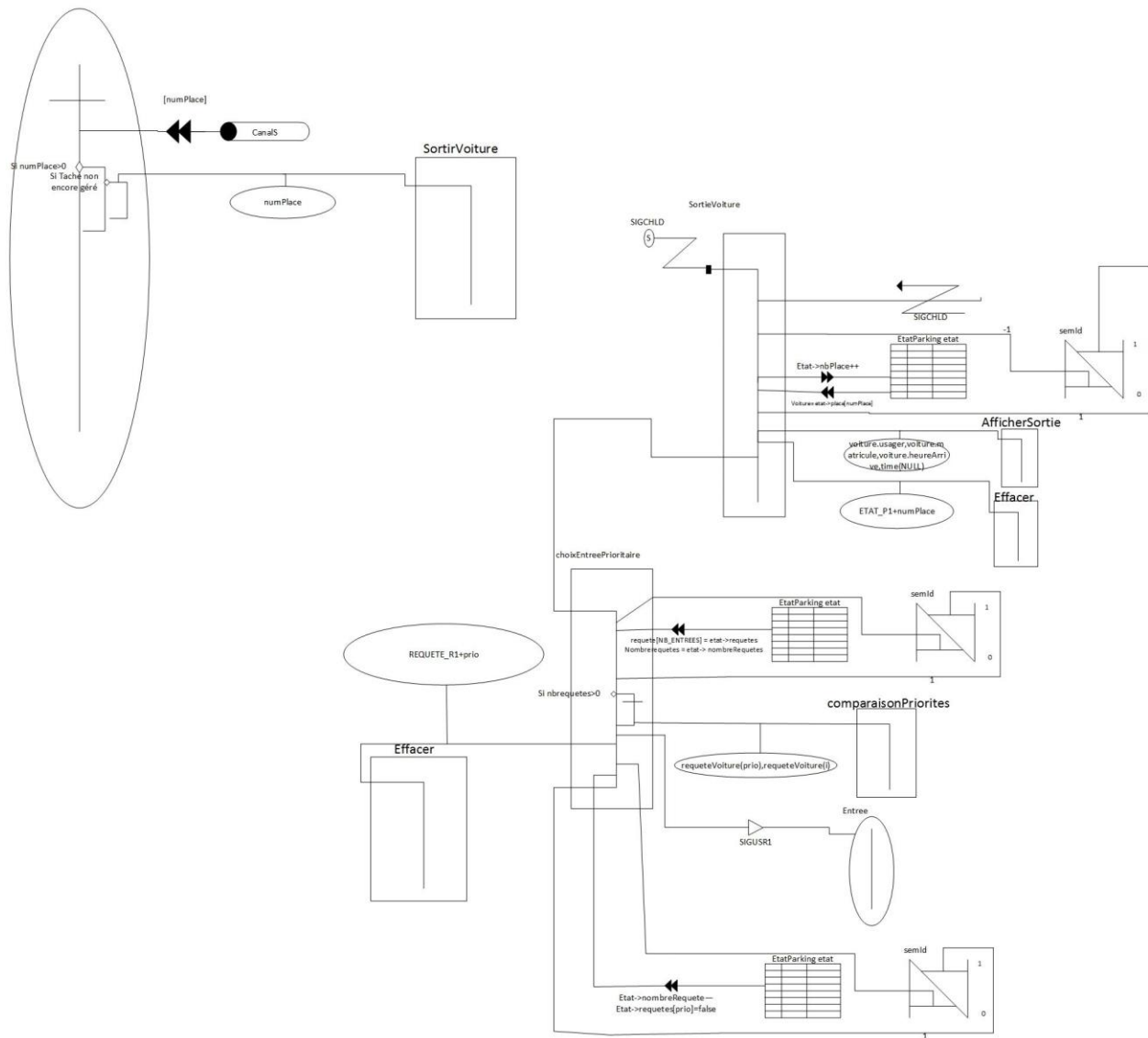
D.1 Initialisation Sortie

Sortie

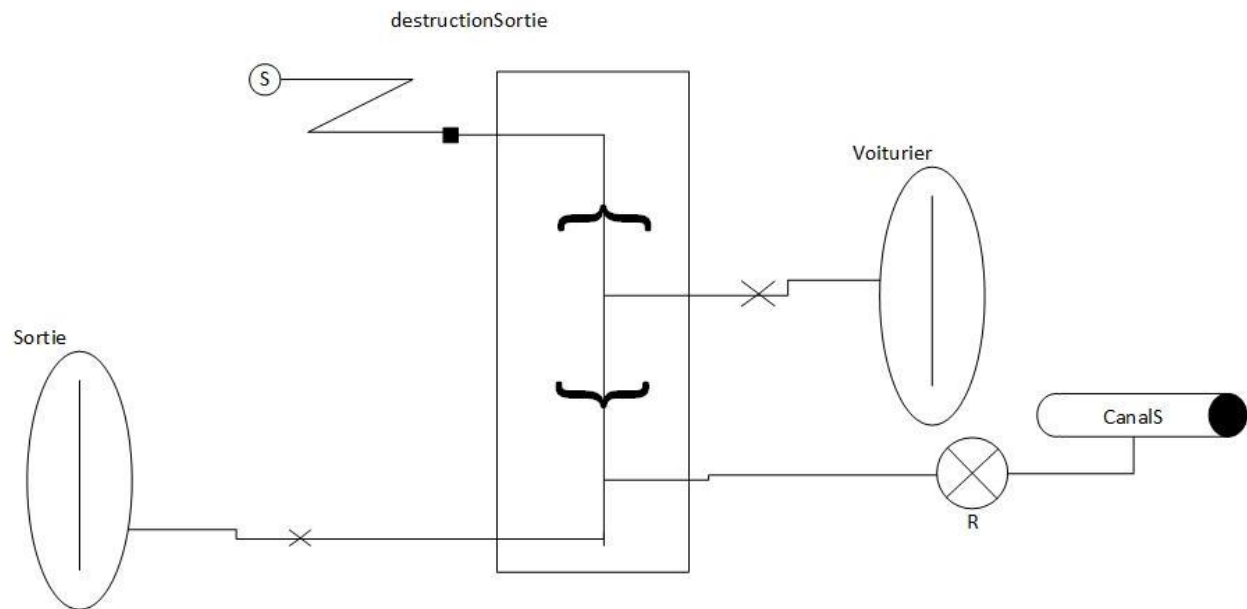


D.2 Moteur Sortie

Sortie



D.3 Destruction Sortie



Specifications

Pour réaliser ce TP, nous avons dû programmer “pas à pas”, voici les différents pas que nous avons atteint :

- Affichage des voitures aux différentes entrées et communication de la tache clavier avec l’entrée concernée
- Sortie des véhicules fonctionnelle, avec affichage des données de sortie (usager, prix, matricule ...)
- Création de requêtes d’entrée et affichage de celles-ci.
- Gestion de l’entrée après qu’une place se soit libérée lorsque le parking était plein avec l’envoi d’un signal de la sortie.

Voici le pas que nous avons tenté de réaliser sans succès :

- Finir l’application proprement si celle-ci se ferme anormalement (CTRL-C) : SIGINT dirigé vers les fonctions de destruction de chaque tâches. **Cependant, les différents IPCs ne se détruisent pas lors d’un arrêt anormal.**

Voici les pas que nous n’avons pas eu le temps de réaliser et aux quels nous avons pensé :

- Modularité de l’application : Notre application fonctionne dans le cas présent mais il faudra modifier le code pour rajouter ou enlever des entrées/sorties, notamment à cause d’un mauvais choix d’IPC : nous aurions dû après coup, privilégier l’utilisation d’une boîte aux lettres à l’utilisation de multiples canaux de communication.
- Réaliser une file de requête pour chaque porte, afin que lorsqu’une requête est prise en compte, une voiture appartenant à la file dont la porte vient de s’ouvrir, puisse se positionner à la barrière et demander un ticket d’entrée (Mise à jour de l’heure de la requête)

Modifications par rapport au schéma d’architecture initial

- Description du contenu de la mémoire partagée et regroupement en un seul bloc
- Modification des voitureurs : une tache voitureur pour réaliser l’entrée d’une voiture et une autre pour s’occuper de la sortie d’une voiture
- Retrait des sémaphores servant à la communication entre Sortie et Entrée.
- Ajout de canaux de communication entre le clavier et les entrées afin que le clavier puisse simuler les entrées.
- Ajout d’un sémaphore d’exclusion mutuelle sur la mémoire partagée, utilisée par le clavier, la sortie, et les entrées.

Conclusion

Le temps a manqué pour terminer cette application de manière à ce que le parking fonctionne en toutes circonstances.

Cela est dû essentiellement au fait que nous avons dû repenser l'architecture plusieurs fois avant d'assimiler les concepts de la programmation multitâche.

Les débuts de la mise en code ont été laborieux, mais l'habitude venant, il nous a manqué que très peu de temps.