

Gestion d'un Parking Automobile MULTITACHE en C/C++ sous Linux.

Salma El ALAOUI TALIBI
Xavier RAZANAJATOVO
B3122

Document de réalisation

1. Architecture

Nous avons apporté les modifications suivantes à l'architecture générale que nous avons conçue pendant la première séance du tp :

- Au lieu d'utiliser une seule mémoire partagée pour assurer la communication entre les tâches entrées et la tâche sortie, nous en avons utilisé deux : une mémoire pour décrire l'état du Parking (un tableau de 8 Vehicules), et une autre qui contient les requêtes, le nombre de places occupées dans le parking et le nombre de requêtes. L'information nombre de requêtes est certes redondante, mais elle nous a permis dans l'implémentation de relâcher la mémoire partagée plus rapidement.
- Nous avons utilisé des canaux de communication nommés aux lieu des boites aux lettres, seulement pour une raison de facilité d'implémentation.
- Nous avons mis en place la synchronisation de fin entre la tâche Mere et la tâche Clavier, après l'auto-destruction de clavier, ce qui n'apparaissait pas dans notre conception initiale.

2. Respect du cahier de charges

Notre solution respecte tout ce qui est requis dans le cahier des charges . Nous pourrions néanmoins travailler sur l'amélioration des performances de notre solution, ou en rajoutant d'autres contraintes, comme des barrières servant à la fois de portes d'entrée et de sortie.

Document de conception

-Nous avons rajouté des documents séparés pour les handlers du signal SIGCHLD dans les tâches Entree et Sortie pour des raisons d'organisation et de clarté.

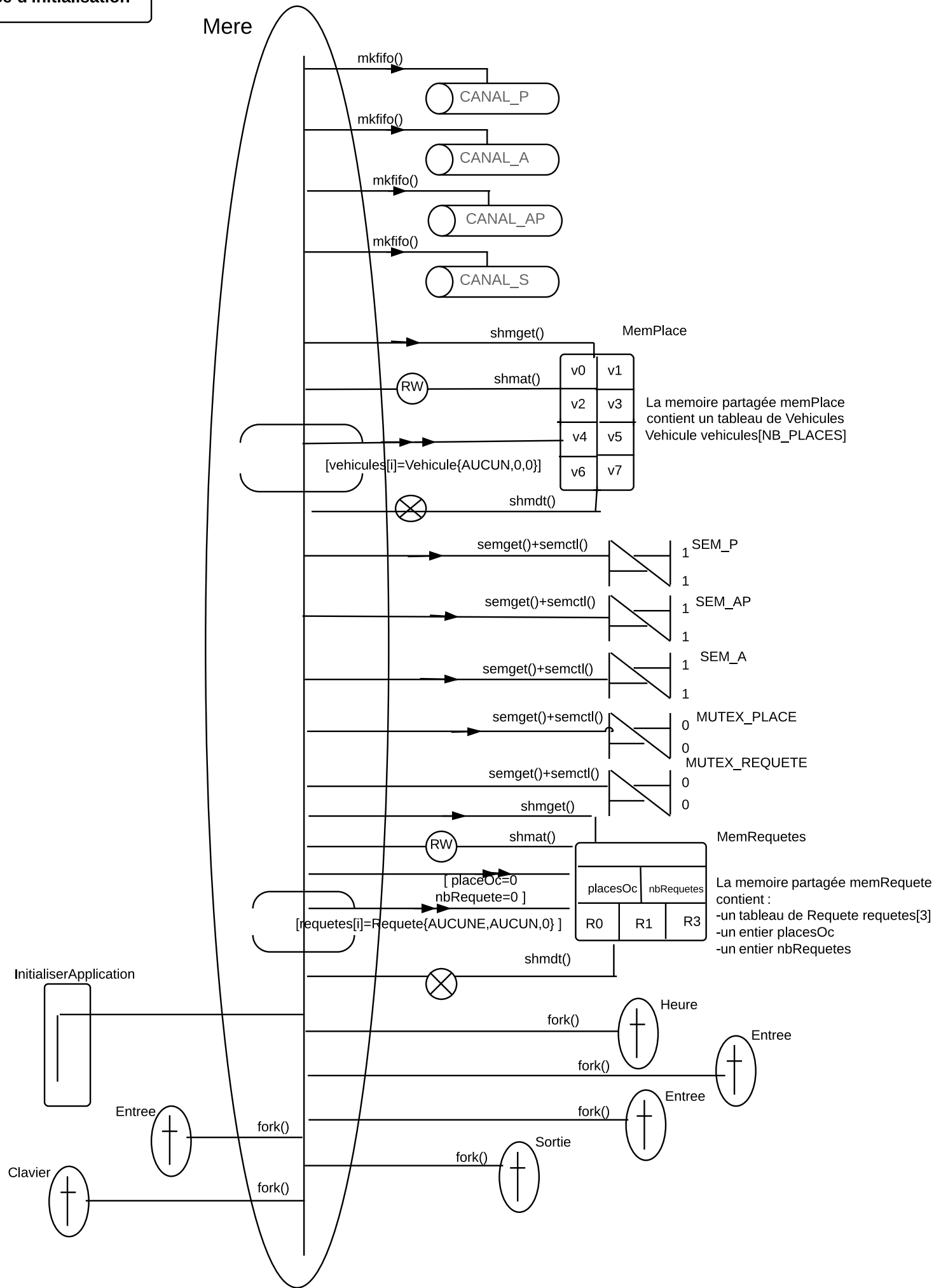
-Précisions pour les différents canaux de communication :

- CANAL_P :
 - producteur : Clavier
 - Consommateur :Entree ayant pour TypeBarriere PROF_BLAISE_PASCAL
 - trame de production : Vehicule(TypeBarriere, noVehicule, time_t).
- CANAL_A:
 - producteur : Clavier
 - Consommateur :Entree ayant pour TypeBarriere AUTRE_BLAISE_PASCAL
 - trame de production : Vehicule(TypeBarriere, noVehicule, time_t).
- CANAL_AP:
 - producteur : Clavier
 - Consommateur :Entree ayant pour TypeBarriere ENTREE_GASTON_BERGER
 - trame de production : Vehicule(TypeBarriere, noVehicule, time_t).
- CANAL_S:
 - producteur : Clavier
 - Consommateur :Sortie
 - trame de production : numéro de la place

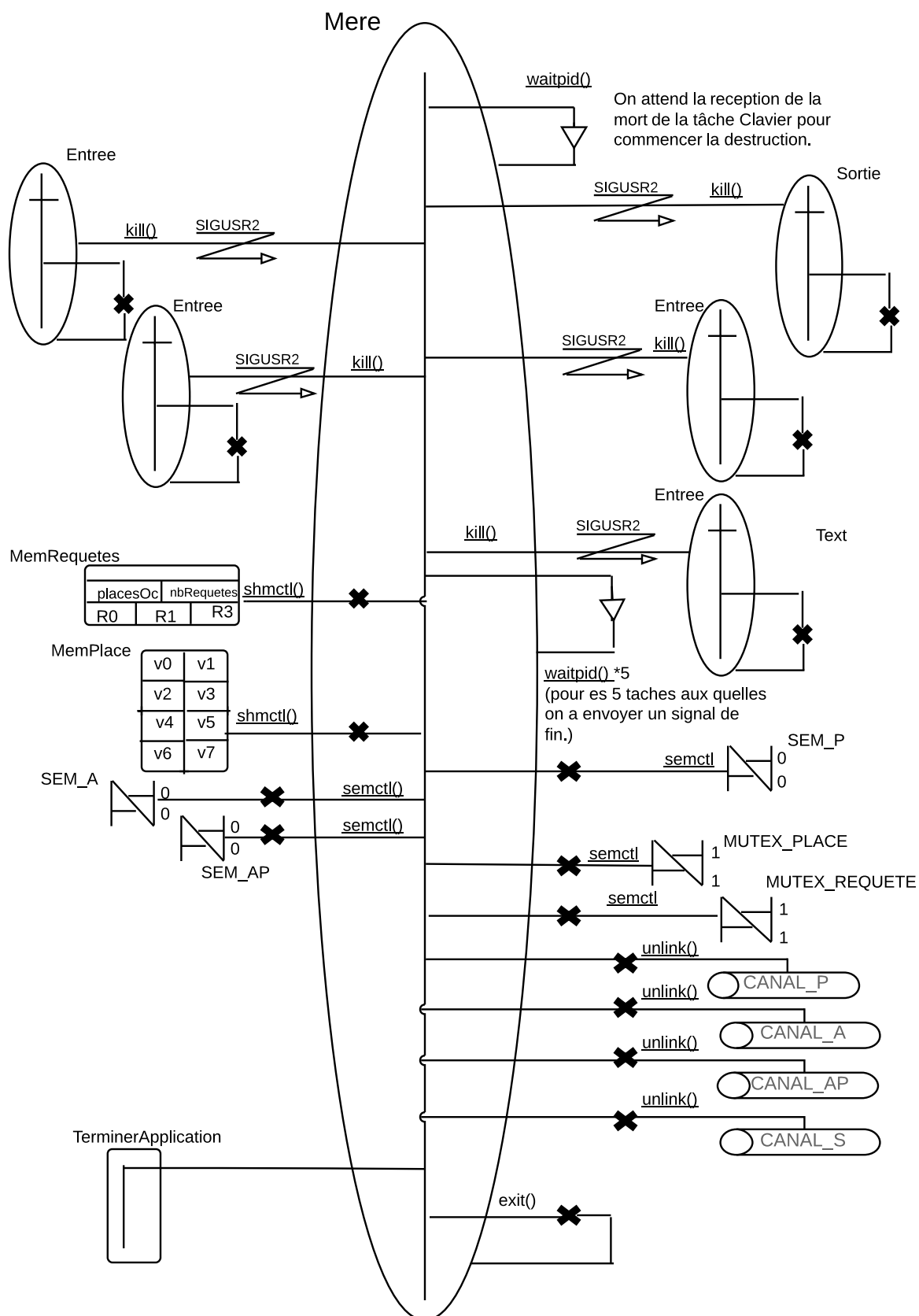
-Précisions pour le paramétrage de la tache Entree :

Pour les 3 taches Entree, on donne en paramètre le type de barriere qu'elle gere pour pouvoir ouvrir le bon canal, ajouter les requêtes au bon endroit...

Mere: Phase d'initialisation

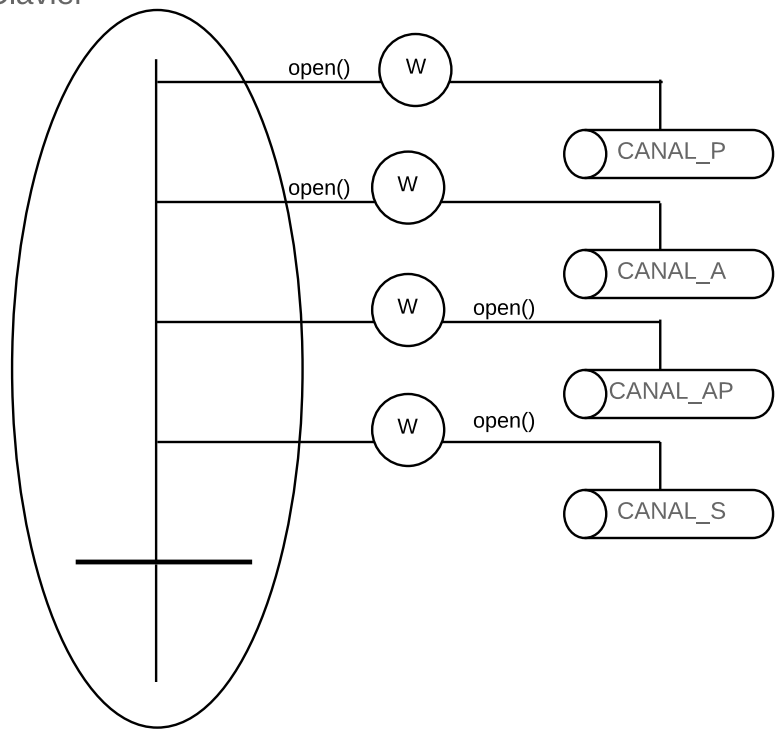


Mere: phase moteur + phase de destruction

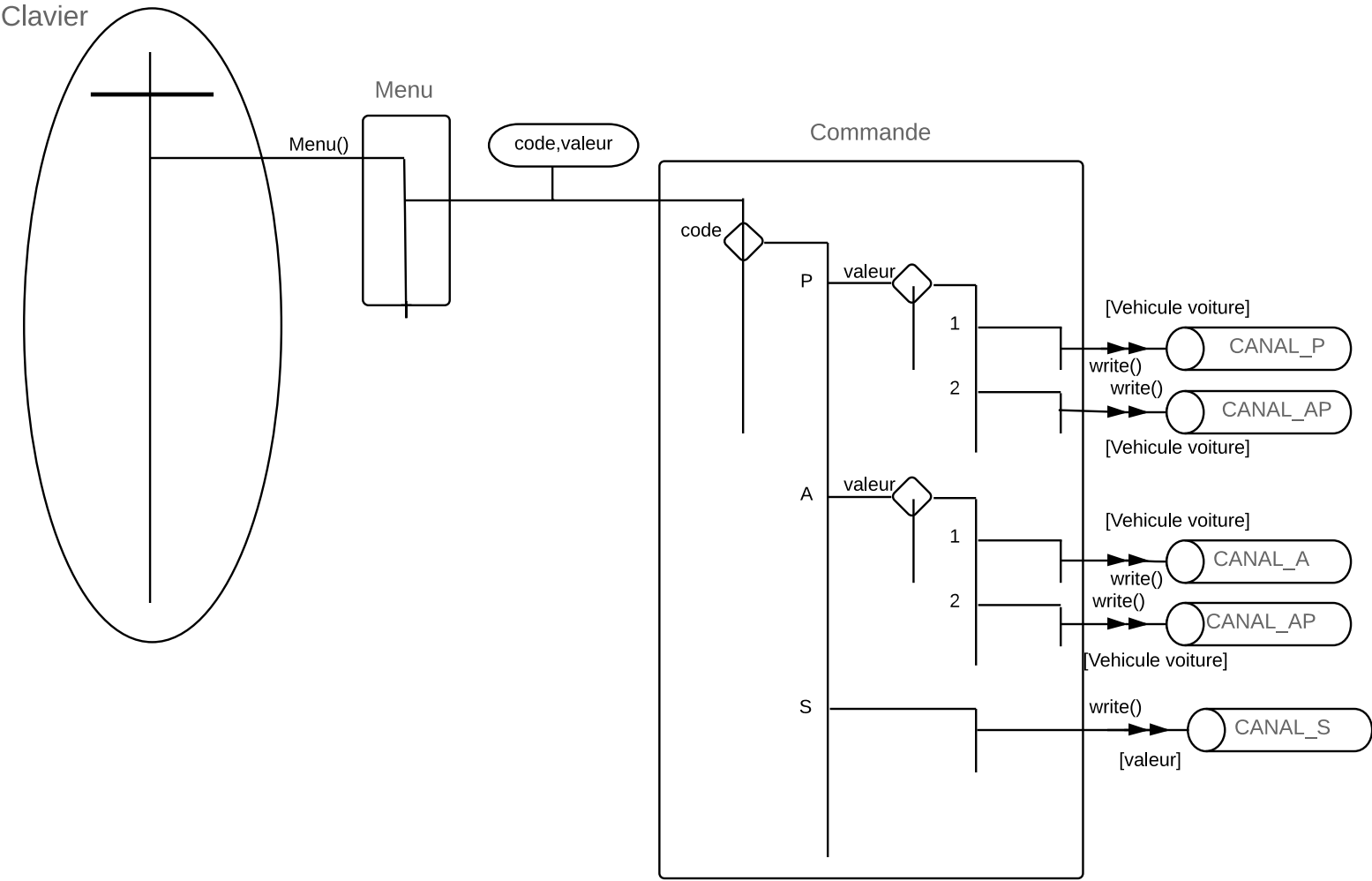


Clavier: phase d'initialisation

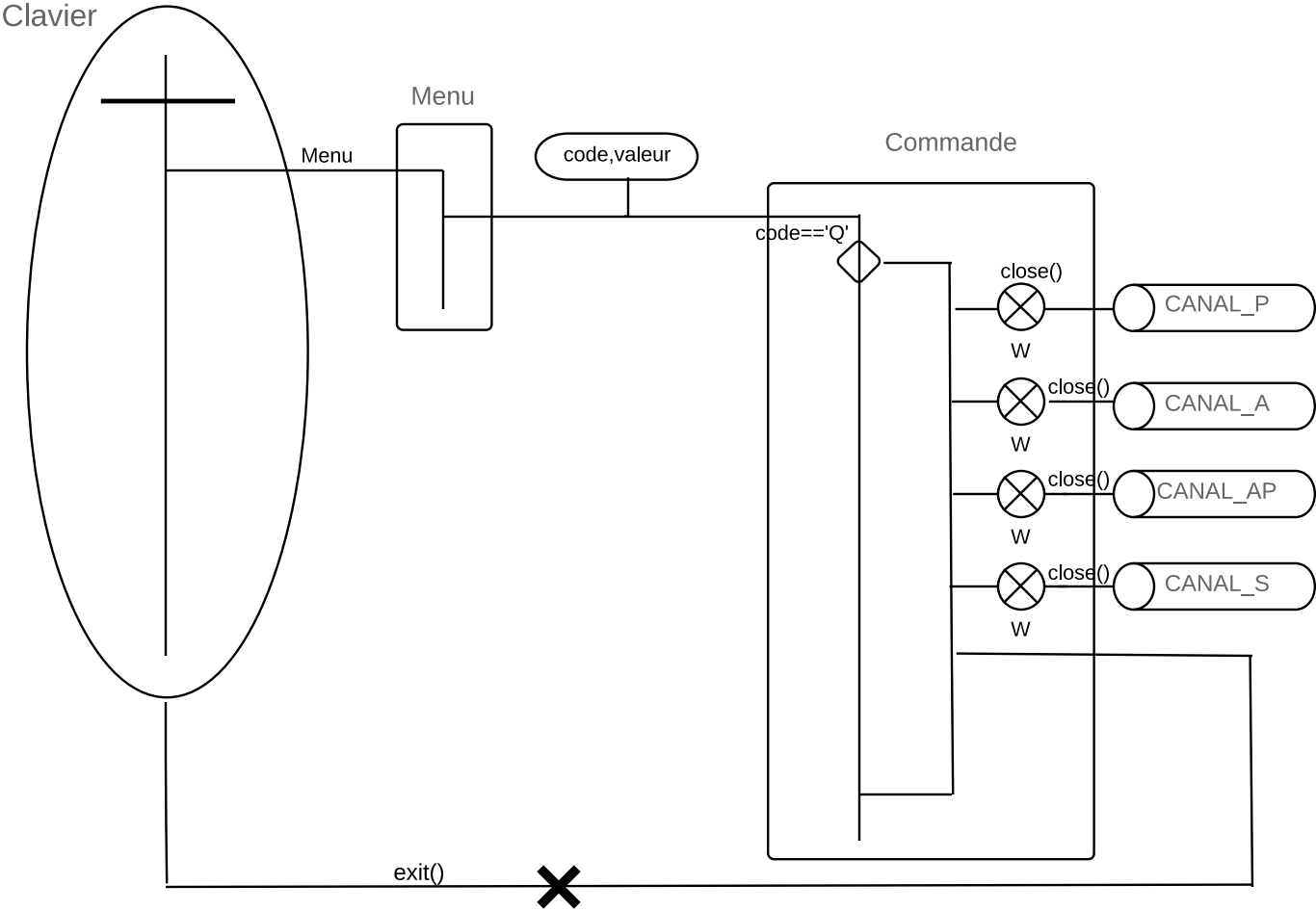
Clavier



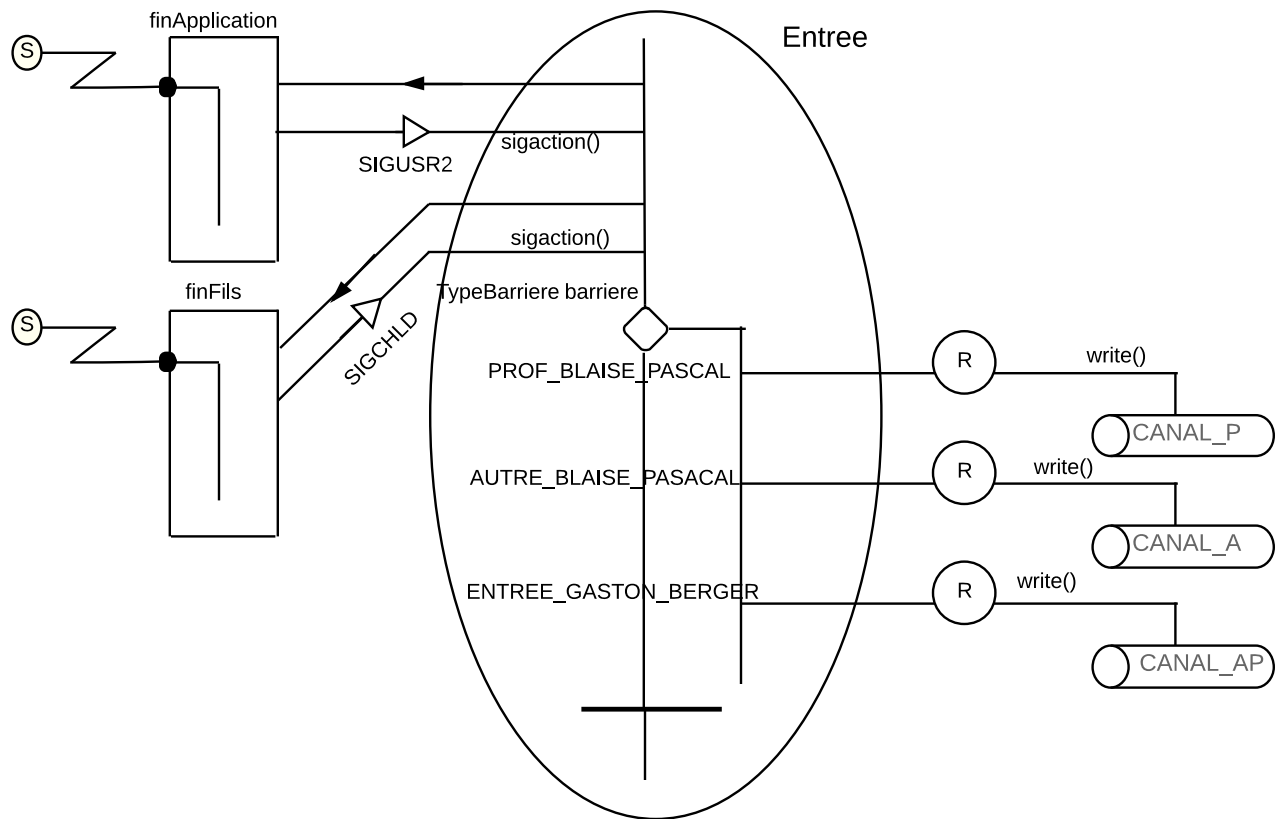
Clavier:phase moteur



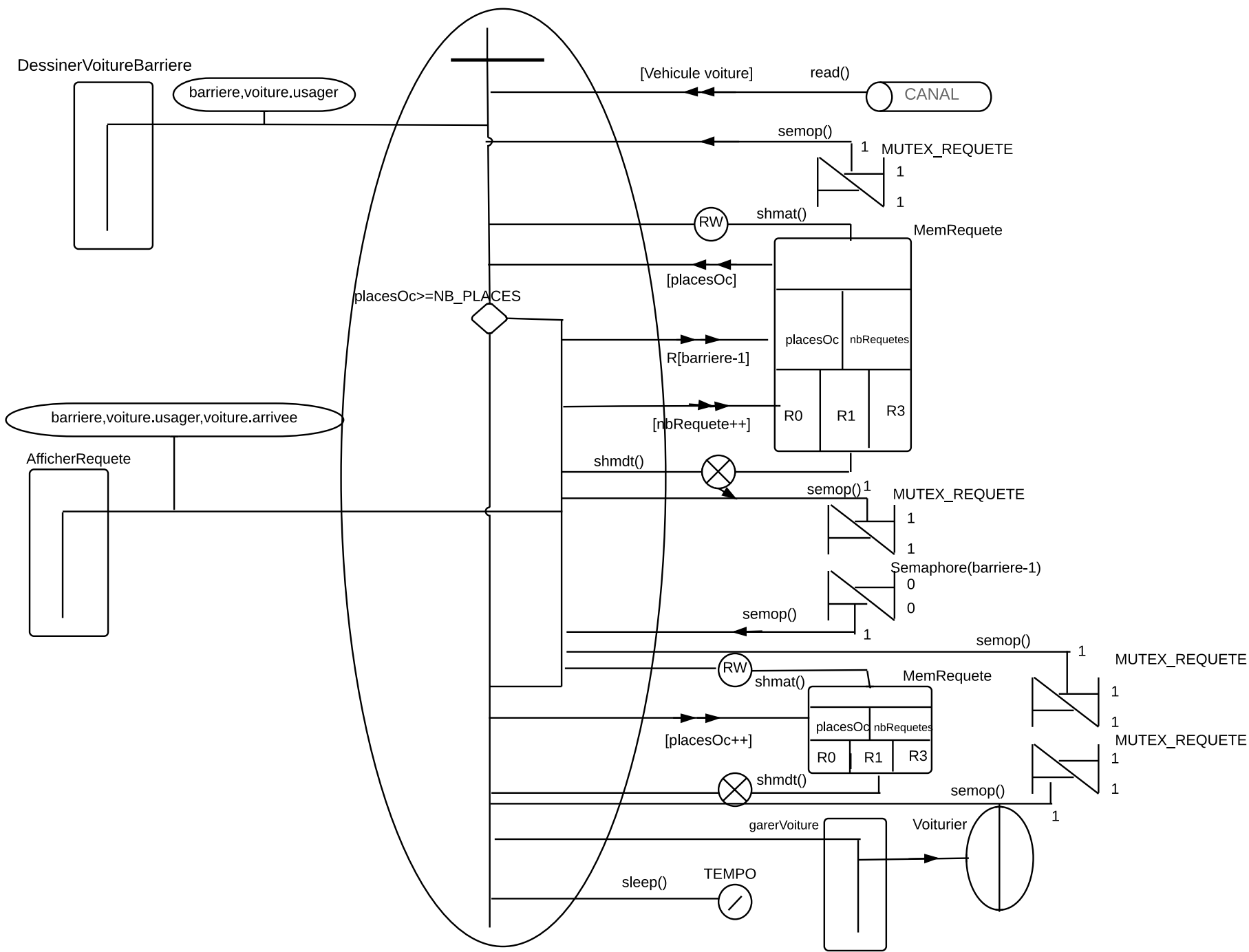
Clavier: Phase de destruction



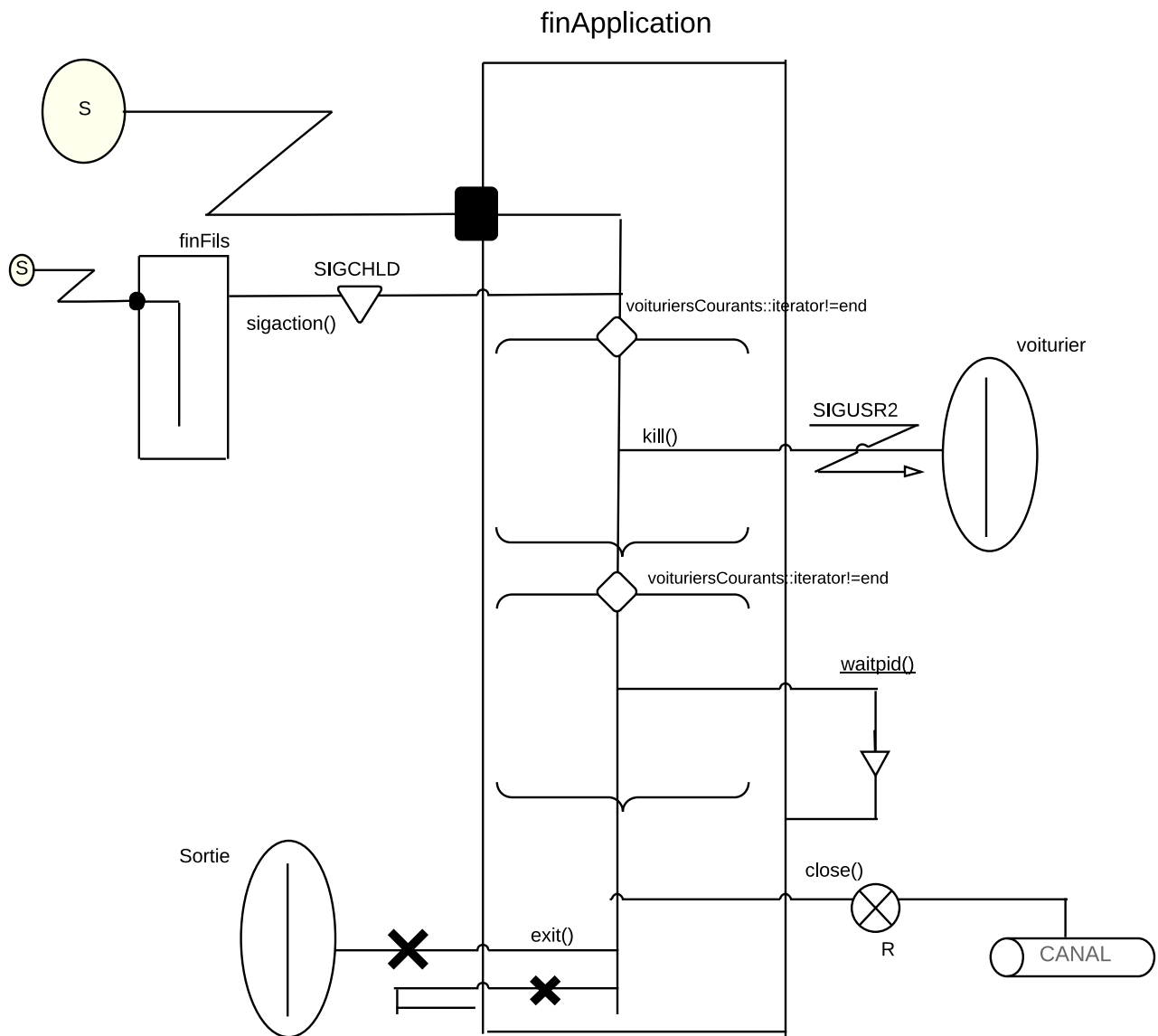
Entree: phase d'initialisation



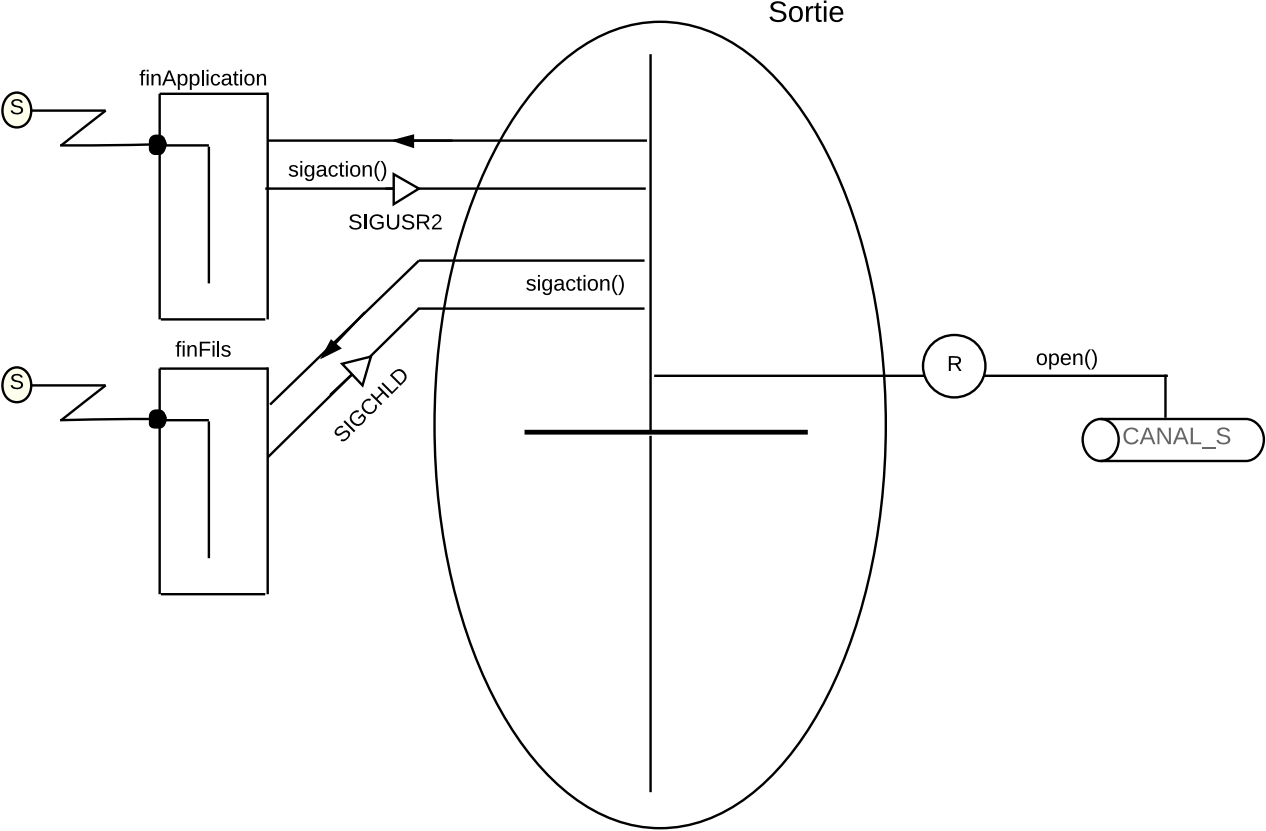
Entree:phase moteur



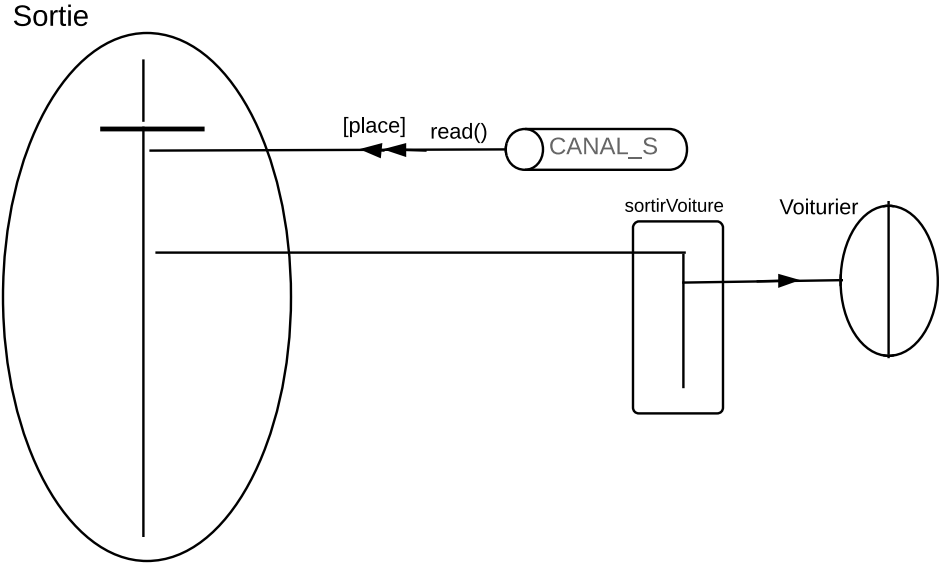
Entree: phase de destruction



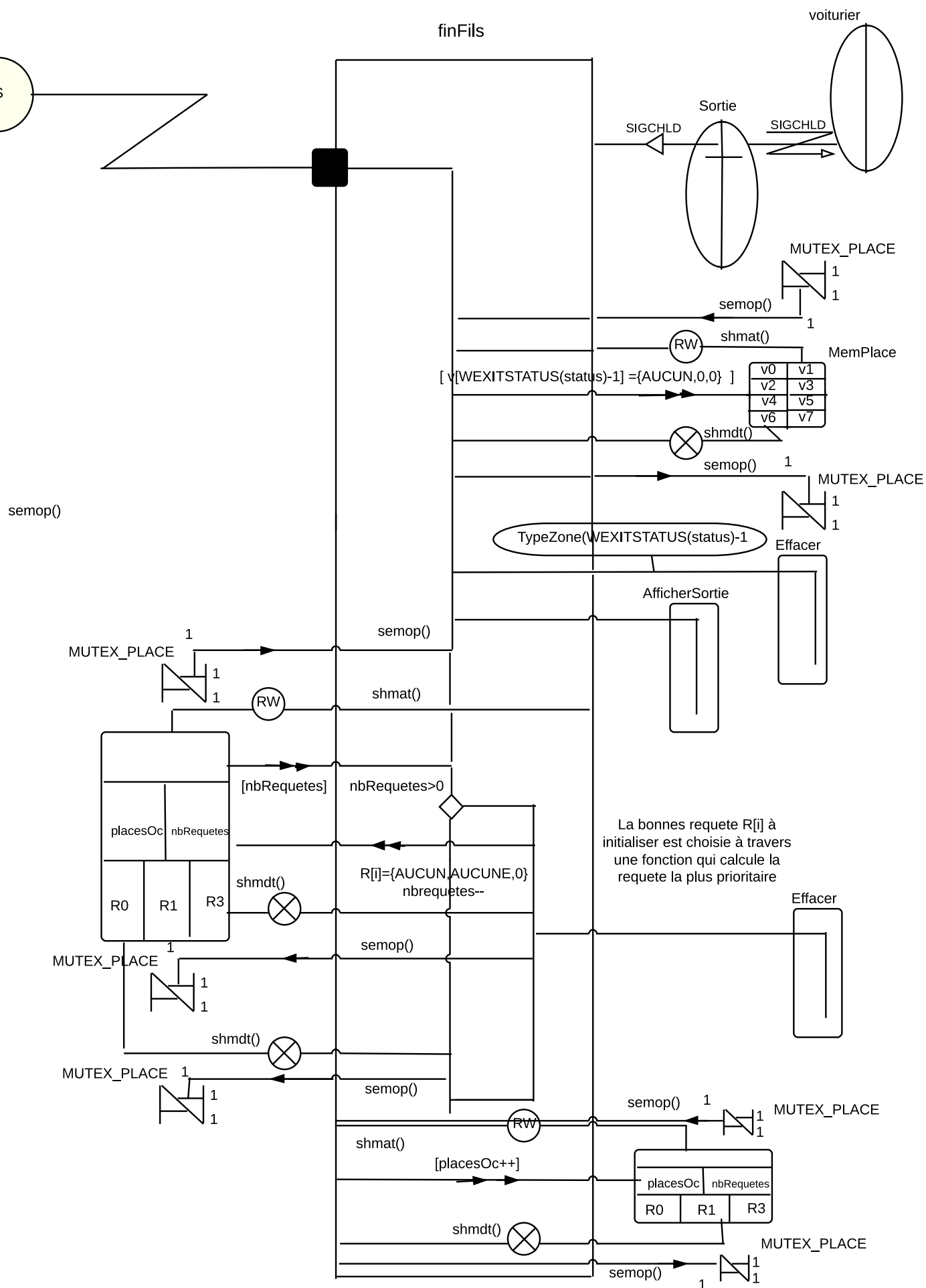
Sortie: phase d'initialisation



Sortie: phase moteur



Sortie: Handler de SIGCHLD (finFils)



Sortie: phase de destruction

