Haute Ecole de la Province de Liège

Informatique industrielle: Labo 2

Rahali Nassim & Schmidt Sébastien - M18

Liège, le 8 décembre 2014

Table des matières

1 Enoncé

Simulation du trafic et de la gestion de l'embarquement des ferries effectuant la traversée de la Manche entre Douvres et Calais ou Dunkerque On considèrera que 2 quais sont disponibles à Calais, 2 quais à Dunkerque et 3 quais sont disponibles à Douvres. Par contre, l'entrée et la sortie du port ne peut se faire que par un seul bateau à la fois.

Il faudra gérer le déplacement de 6 ferries ainsi que l'arrivée et le chargement de véhicules de 3 sortes (voiture, camionnettes et camions). Les camions ne sont pas embarqués sur le même pont que les voitures et les camionnettes. L'embarquement simultané est donc possible, alors que les voitures sont toujours embarquées avant les camionnettes.

L'application contiendra un fichier de configuration qui décrira les caractéristiques nécessaires à la simulation.

L'application de simulation crée des véhicules disponibles pour l'embarquement, gère l'embarquement et le débarquement de ceux-ci. De plus Elle gère le déplacement des navires.

L'interface utilisateur montre au mieux l'évolution des navires dans les ports et en mer.

2 Description

2.1 Situation initiale

Pour commencer, nous considérons que les bateaux sont créés en mer par facilité. De cette façons nous n'aurons pas à initier les sémaphores de façon dynamique. De plus les bateaux sont supposés vides, de cette façon la première fois qu'ils arriveront dans un port, ils ne feront que l'embarquement.

2.2 Entrée dans un port

Lorsqu'un bateau désire rentrer dans un port, il doit se faire par un canal prévu pour un seul bateau à la fois. Quand tous les quais sont occupés, les bateaux souhaitant rentrer dans le port devront attendre dans un canal d'attente. Cette attente est une file, ils seront traités dans l'ordre de leur arrivée.

2.3 Départ de quai d'un bateau

Lorsqu'un bateau arrive à quai, il commence par débarquer les véhicules et ensuite il chargera les véhicules qui sont prévus sur le quai. Dans notre application, nous ne gérons pas d'horaire pour les ferrys, c'est-à-dire que les bateaux partiront une fois que l'ensemble des véhicules sera à bord. Le bateau est donc parfaitement autonome, il n'a pas besoin de consignes venant d'un autre processus pour partir en mer.

2.4 Génération des véhicules

On considère que les véhicules disponibles à l'embarquement sont fourni par le processus Quai. C'est donc lui qui se responsable de la création des différentes ressources pour réaliser cette tâche. Nous avons décidé de simuler l'arrivée de véhicules par des MessageQueues. Comme nous avons deux files de véhicules : une pour les voitures/camionnettes et une pour les camions, la création de deux files de messages sera nécessaire. De plus, les objets créés d'une des deux files aura des priorités différentes afin de modéliser les voitures et les camionnettes.

3 Ressources

3.1 Structures de données

Pour cette application, nous avons besoin d'une seule structure de données pour représenter les bateaux, elle contiendra :

Pid : Le pid des processus bateaux afin que les différents processus puissent envoyer des signaux au processus bateau correspondant.

Position : Indique où se trouve le bateau. Ce champ pourra prendre les valeurs suivantes : En mer, Arrive port, A quai et Quitte port.

Direction : Représente le port vers lequel le bateau se dirige, il y aura donc : Dover, Calais ou Dunkerke.

Nom MessageQueue1 : On aura le nom de la MessageQueue propre à ce bateau représentant les voitures et camionnettes.

Nom MessageQueue2 : On aura le nom de la MessageQueue propre à ce bateau représentant les camions.

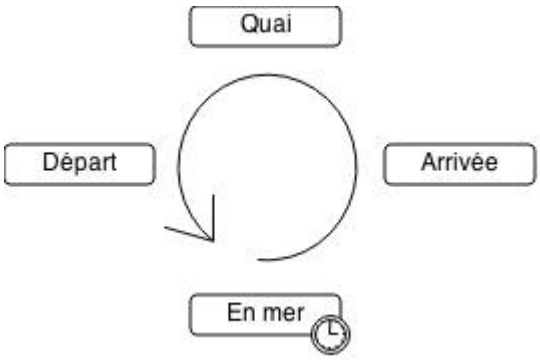

FIGURE 1 – Cycle des différentes positions d'un bateau

3.2 Initialisation des sémaphores

Nom	Valeur	Description
Sem_Port	0	Permet l'entrée des bateaux dans un port
Mutex_Dep	1	Section critique sur la variable globale Cpt_dep
Mutex_Arr	1	Section critique sur a variable globale Cpt_arr
Mutex_Bateau	1	Section critique sur la mémoire partagées des bateaux
Mutex_Shm_Quai	1	Section critique sur la mémoire partagée des quais
Sem_Quai	0	Permet de mettre en attente le processus Quai, il sera réveillé par un bateau
Mutex_Gen_PortX	1	Permet de synchroniser le processus GenVehicule. Cette sémaphore sera présente 3 fois, une pour chaque port

3.3 Mémoires partagées

Nom	Taille	Description
SHM_Quais	(2 ou 3)*sizeof(int)	Permet à un port de connaître l'occupation de ses quais.
SHM_Bateaux	6 * sizeof(struct Bateau)	Permet à tous les processus de connaître les différentes informations propres aux bateaux.

3.4 MessageQueue

Afin de représenter les véhicules présents dans les bateaux, nous avons décidés de créer des files de messages. Comme on pouvait embarquer simultanément véhicules lourds et véhicules légers, nous avons trouvé légitime d'en créer deux par bateaux :

- La première contiendra donc les véhicules «légers» tels que les voitures et les camionnettes. Comme les camionnettes doivent embarquer en premières, on indiquera une priorité plus haute à celles-ci afin qu'elles rentrent en premières dans le bateau.
- La seconde contiendra les véhicules lourds comme les camions.

Comme chaque bateau embarque ses propres véhicules, il faut donc un total de 12 MessageQueues pour représenter le pont de tous les bateaux qui ont été générés et qui effectuent la traversées de la Manche.

3.5 Les signaux

Afin de prévenir la fin de tâche, nous avons décider d'utiliser les signaux POSIX. Les processus qui pourront être susceptible de recevoir ce genre de signaux devront armer ces signaux et modifier le handler de ceux-ci afin d'avoir le résultat souhaité. Nous n'avons pas encore décider d'attribuer un tel signal à une telle action, ce choix sera effectué lors de la programmation car il n'influence en rien une modification quelconque de nos diagrammes de processus ni notre réflexion.

4 Schéma de processus

4.1 Processus Gestion

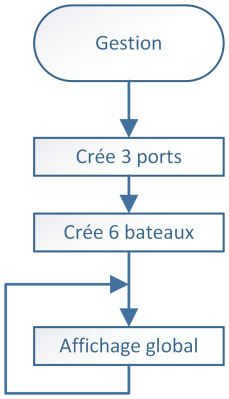

FIGURE 2 – Diagramme du processus Gestion

Le rôle de ce processus est en de créer les différents processus qui seront utiles pour gérer le trafic maritime ainsi que d'afficher les informations globales. Il va donc avoir à créer 6 processus bateaux et ensuite 3 processus ports avec des paramètres différents. Ces paramètres correspondront au nombre de quais que ce port pourra gérer.

Concernant l'affichage, ce processus parcourra les différentes mémoires partagées, détaillées au point 3.3, afin de collecter les informations nécessaires dans le but d'avoir une vue d'ensemble sur l'évolution des différents processus.

4.2 Processus Port

Le processus Port, dans un premier temps, est responsable de la création des processus Quai qui sont rattachés à ce dit port ainsi que du processus GenVehicules spécifique à chaque port. Comme on sait que deux bateaux ne peuvent rentrer en même temps dans le canal du port, le processus Port est en attente d'un bateau. A chaque fois qu'un bateau va se présenter devant le port, il va faire un signal sur la sémaphore SEM_PORT. De cette façon le processus Port exécutera autant de fois qu'un bateau désire rentrer dans le port.

Il aura ensuite la responsabilité de gérer le passage des bateaux aussi bien en entrée qu'en sortie. Chaque fois qu'un bateau aura la possibilité de soit sortir ou bien de rentrer, le processus Port utilisera un signal afin de prévenir que l'action peut être effectuée en toute sécurité. Pour récupérer le pid du bateau afin de lui transmettre un signal, on aura auparavant parcouru la mémoire partagée des bateaux afin de savoir quel bateau signaler.

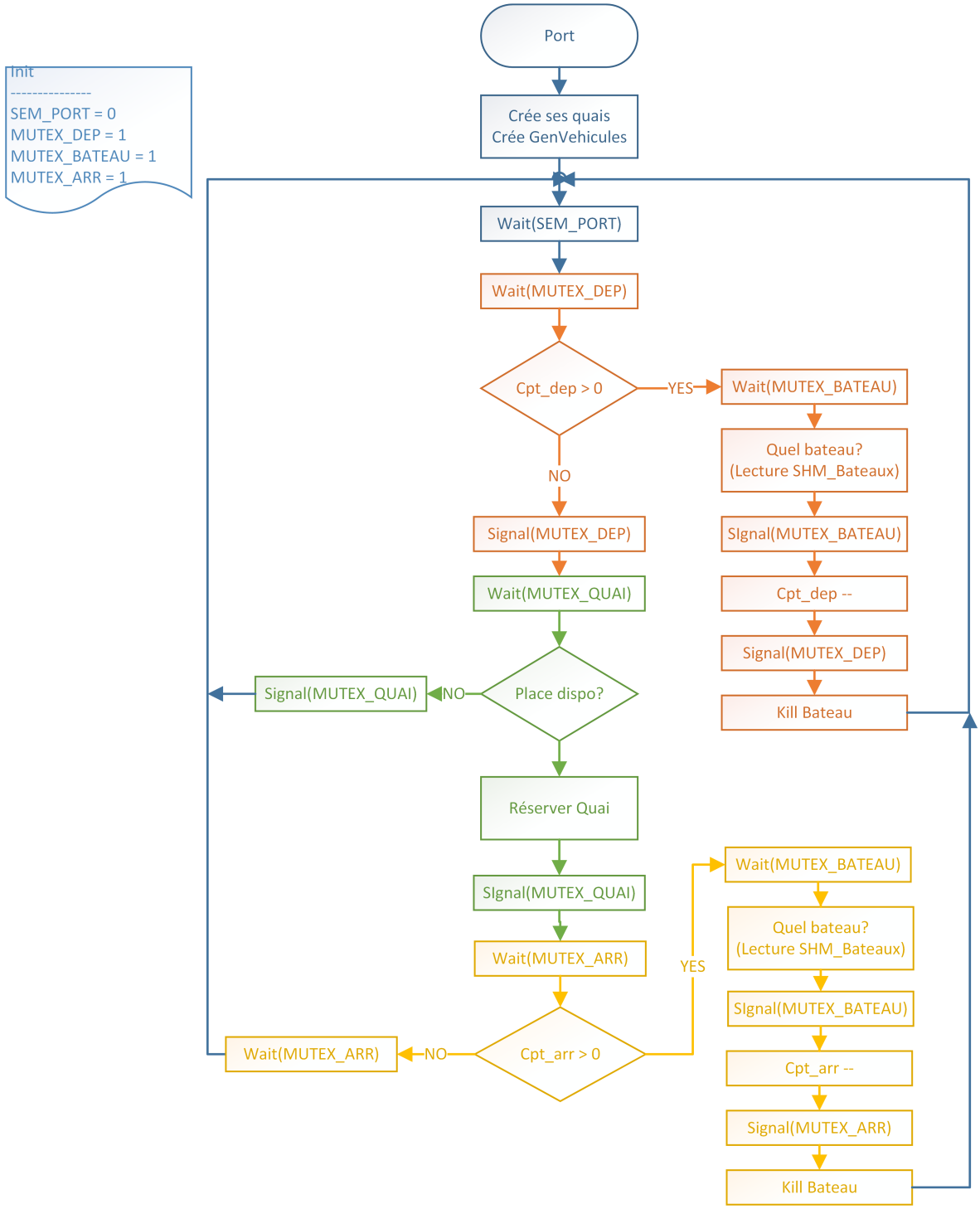

FIGURE 3 – Diagramme du processus Port

4.3 Processus Quai

Les Quais sont charger d'effectuer l'embarquement ainsi que le débarquement des différents véhicules présents à bord du bateau présent à ce quai. Tout d'abord comme les différents véhicules sont représentés par des files de messages, il doit d'abord la vider si celle-ci contient des éléments. Ensuite, après un petit temps d'attente, il devra mettre à bord de nouveaux véhicules désirant effectuer la traversée. Dans ce cas-là, il va débloquer le processus GenVehicules qui aura lui la tâche de remplir les MessageQueues.

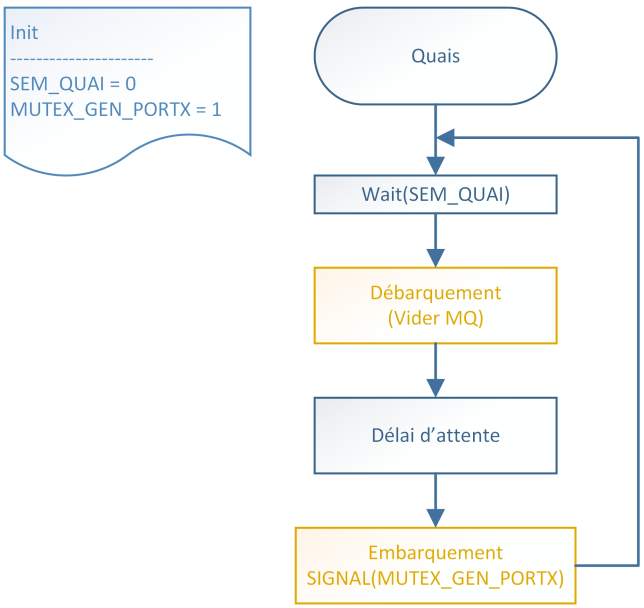

FIGURE 4 – Diagramme du processus Quai

4.4 Processus Bateau

Comme dit au point 2.1, les bateaux sont générés en mer lors du démarrage de l'application. Ensuite on va rentrer dans une boucle qui, en fonction de la position du bateau (récupérée dans une mémoire partagée) va effectuer différentes actions :

En mer : Le bateau va juste générer un nombre aléatoire pendant lequel il va être mis en pause. Ce temps correspondra à la traversée de la Manche. Ce temps sera donc différent en fonction des différents ferry. On peut le comprendre par des conditions météorologiques différentes voir un chargement plus ou moins lourd.

Arrivée : On incrémente le compteur d'arrivée représentant le nombre de bateaux désirant de rentrer dans ce port. Il signalera ensuite sa présence auprès du port et se mettra en attente d'un signal lui permettant d'emprunter le canal.

Quai : Il prévient le quai qu'il est près à débarquer les véhicules à bord s'il y en a et d'en embarquer de nouveau une fois terminé. De nouveau il va se mettre en attente de signal lui permettant de préparer son départ.

Départ : Action similaire à l'arrivée, il signale au port qu'il est prêt à partir en incrémentant un compteur de départ. Il attendra ensuite un signal pour confirmer son départ.

A noter qu'une fois à la fin de chaque situation, le bateau va entrer dans l'état suivant et mettra la mémoire partagée des bateaux à la fin de cette boucle. Par exemple, si le bateau est en mer il arrivera ensuite au port, si il est à quai il va se préparer à partir en prévenant le port d'un départ imminent. De plus, ce processus va devoir être armé sur plusieurs signaux permettant différentes actions. Le handler de ces signaux devra bien évidemment être redéfini.

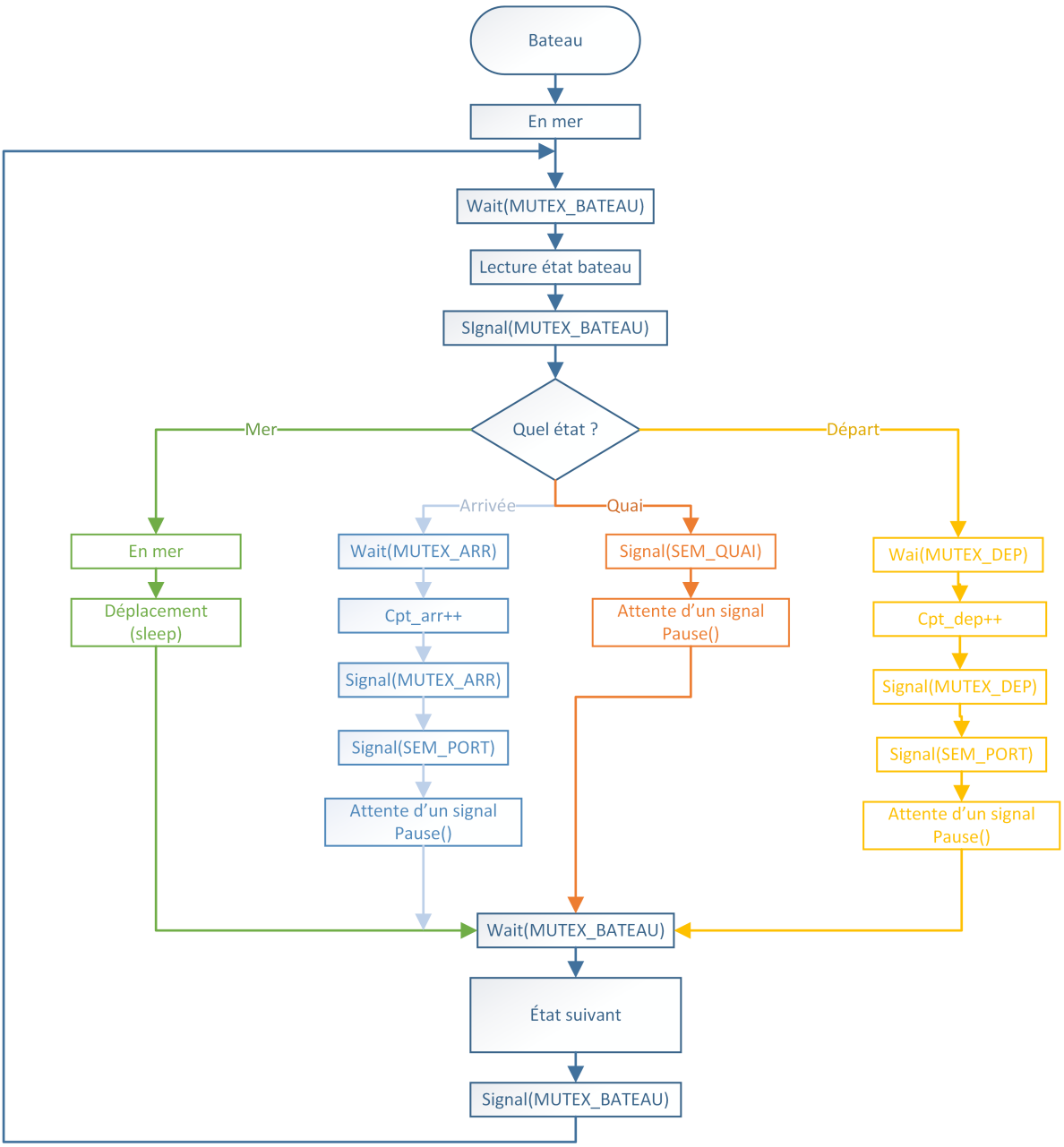

FIGURE 5 – Diagramme du processus Bateau

4.5 Processus GenVehicules

Le rôle de ce processus est de remplir les différentes MessagesQueues des bateaux. Il existe trois répliques de ce processus : chacune d'entre elles sera responsable de la génération des véhicules pour un port. Au moment où un bateau amarré à un quai a besoin de remplir ses différents ponts, il va réveiller ce processus qui est bloqué sur la sémaphore (utilisée sous forme de mutex). Une fois que celui-ci a complété sa tâche, il enverra un signal au processus Bateau correspondant afin de lui signaler la fin de l'embarquement.

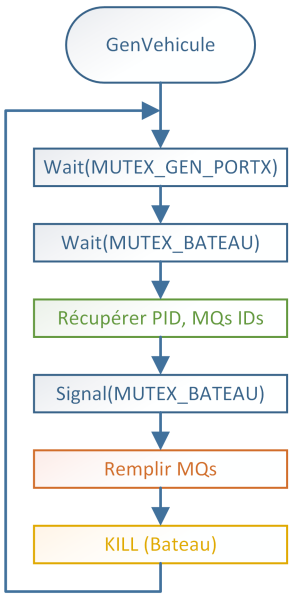

FIGURE 6 – Diagramme du processus GenVehicules

5 Évolutions possibles

5.1 gestion d'un horaire

Dans cette application, nous n'avons pas gérer un système d'horaires que les bateaux auraient dû respecter. nous aurions pu mettre cela en place grâce aux timers qui auraient été activés une fois le bateau à quai. Un signal aurait alors été envoyé au bateau lui signalant un départ imminent. Si l'embarquement n'avait pu être terminé à temps, le processus Bateau aurait envoyé un signal au processus GenVehicule afin de stopper l'embarquement. Les véhicules restant auraient dû être pris en charge par le prochain bateau jetant les amarres à ce port.