

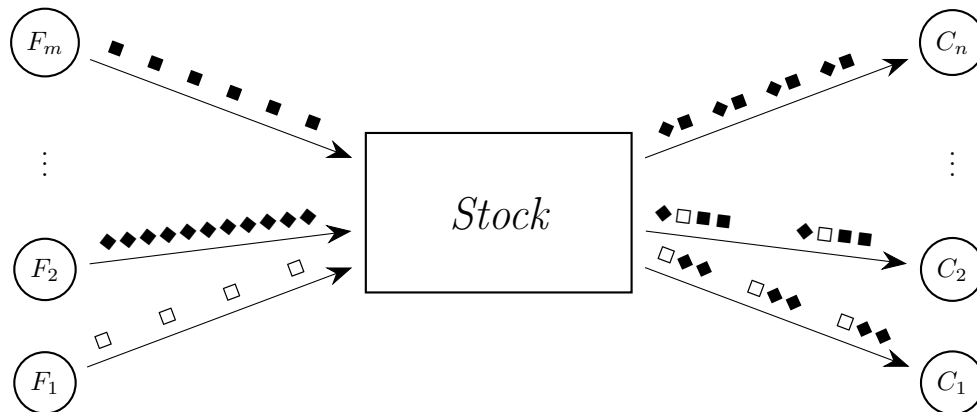
# Informatique Temps-Réel — Travail

## Gestion de stock

Seweryn Dynierowicz (Seweryn.DYNEROWICZ@umons.ac.be)

### 1 Contexte

Dans le cadre de ce travail, il vous est demandé d'implémenter une solution au problème de la gestion d'un stock de marchandises. Trois types d'entités sont impliquées dans cette problématique ; les *fabricants*, le *gestionnaire* du stock et les *clients*. Les *fabricants* produisent chacun un type de produit à un rythme déterminé, le *gestionnaire* du stock est en charge de les entreposer endéans des limites de capacité et les *clients* récupèrent les produits selon leurs besoins.



#### 1.1 Produits

Les *produits* qui interviennent dans ce problème sont caractérisés par les attributs suivants :

- Un **identifiant** représentant le type de produit considéré
- Une **description textuelle** d'un maximum de 256 caractères
- Un **volume** décrivant la place occupée par le produit dans le stock

#### 1.2 Fabricants

Chaque *fabricant* est spécialisé dans la fabrication d'un *produit* spécifique. Les *fabricants* ne disposent de la place que pour entreposer une unité du *produit* qu'ils fabriquent. Lorsque une unité est prête, le *fabricant* notifie le *gestionnaire* de stock. Tant qu'elle n'est pas récupérée, l'unité occupe la chaîne de montage et empêche la fabrication. Chaque *fabricant* est caractérisé par les attributs suivants :

- L'**identifiant** du produit qu'il fabrique
- Le **temps** nécessaire à la fabrication d'une unité du produit

Pendant la journée, la production tourne en continu et chaque unité produite se voit attribuer un **numéro de série** qui l'identifie.

#### 1.3 Gestionnaire de stock

Le *gestionnaire* du stock a la responsabilité de récupérer les *produits* sur les chaînes de montage, de les entreposer et de les expédier sur base des bons de commande venant des *clients*. Celui-ci est caractérisé par un seul attribut :

- Le **volume** total de l'espace de stockage disponible pour tous les *produits* entreposés
- Un **découpage** de ce volume total, réparti entre les différents types de *produits* pour leur stockage.

Le gestionnaire accumule à tout moment les bons de commande provenant des *clients*. Pendant la journée, un bon de commande peut être honoré à partir du moment où l'intégralité des *produits* sont disponibles en quantité suffisante dans le stock. Dans ce cas de figure, les quantités requises sont retirées du stock et envoyées au client. Lorsqu'un *fabricant*

notifie le gestionnaire qu'une unité est prête, celui-ci la récupère et la place dans l'espace dédié à ce *produit* dans le stock pour autant qu'il y ait de la place.

## 1.4 Clients

Les *clients* récupèrent les *produits* selon leurs besoins respectifs. Chaque *client* est caractérisé par les attributs suivants :

- Un **identifiant** auprès du *gestionnaire* de stock
- Un temps **minimum** entre les demandes successives de *produits* auprès du stock
- Un temps **maximum** entre les demandes successives de *produits* auprès du stock
- Une **liste** reprenant les produits qui sont récupérés (*i.e.* **identifiant** et **quantité**).

Un *client* peut envoyer une commande à n'importe quel moment, en journée ou pendant la nuit. Tant que le dernier bon de commande envoyé n'a pas été honoré, le client n'envoie pas de nouvelle commande.

## 2 TRAVAIL DE GROUPE

Implémentez l'application décrite ci-dessus sous la forme de trois processus. Les différents *fabricants* (resp. les *clients*) sont organisés sous la forme de multiples *threads* au sein du processus **fabricants** (resp. **clients**).

Utilisez des signaux temps-réel ainsi que des zones de mémoires partagées aux endroits appropriés.

Votre implémentation doit prendre en compte le cycle jour-nuit. Vous pouvez choisir les unités de temps pour que le temps passe plus vite (*e.g.* une seconde réelle représente une heure dans le système).

**Note :** pour vos tests, vous pouvez manuellement créer vos jeux de tests dans le code à travers des structures/tableaux statiques.

**Conseils :** Commencez par définir les structures pour représenter les différents éléments de l'application, dessinez et documentez les diagrammes d'états des trois types d'entités. Organisez et structurez bien votre code.