

## Laboratoire de Programmation Concurrente semestre automne 2025

### Gestion d'accès concurrents

Temps à disposition : 4 périodes (deux séances de laboratoire)

Récupération du laboratoire : `retrieve_lab pco25 lab03`

## 1 Objectifs pédagogiques

- Mettre en évidence les problèmes liés aux accès concurrents.
- Se familiariser avec les *mutex*.
- Protéger les accès concurrents à l'aide de *mutex*.

## 2 Cahier des charges

Votre objectif est de développer une simulation non-exhaustive du système hospitalier d'une ville. En voici les cinq acteurs principaux :

1. Les ambulances : Elles transportent les patients malades vers les hôpitaux. Une fois un transfert effectué, elles paient le salaire de l'ambulancier (système contractuel) et envoient leur facture à l'assurance maladie.
2. Les hôpitaux : Ils sont le centre névralgique de la simulation. Les ambulances leur amènent des patients malades qu'ils gardent allités en attendant de pouvoir les transférer vers une clinique. En effet, **les hôpitaux ne participent pas au traitement des patients** et délèguent cette tâche aux cliniques spécialisées. Après le traitement d'un patient dans une clinique, le patient est renvoyé à l'hôpital pour y effectuer son séjour de convalescence (5 jours<sup>1</sup>). Lors du départ d'un patient de l'hôpital (transféré vers une clinique et après sa convalescence), une facture est envoyée à l'assurance maladie pour payer les frais du patient. Les hôpitaux sont les seules entités à avoir un staff fixe (non-contractuel). Cela signifie que l'entièreté du service de l'hôpital reçoit un salaire à la fin de chaque jour (indépendamment du nombre de patients allités).
3. Les cliniques : Elles s'occupent de traiter les patients malades transférés depuis les hôpitaux. Chaque jour, pour traiter un patient, elles identifient le matériel médical nécessaire, commandent ce qui leur manque auprès de leurs fournisseurs et appliquent le traitement, mais seulement si les fonds de la clinique sont suffisants pour payer le spécialiste. Une fois le traitement appliqué, le patient est allité le temps de pouvoir le renvoyer dans un hôpital pour qu'il effectue son séjour de convalescence. Une fois que le patient quitte la clinique, la facture est envoyée à l'assurance maladie.
4. Les fournisseurs : Ceux-ci s'occupent de fabriquer et vendre du matériel médical aux cliniques spécialisées. Chaque jour, ils choisissent une ressource à fabriquer et, si leurs fonds permettent de payer le salaire de leurs employés, ils fabriquent cette dernière et l'ajoute à leur stock.
5. L'assurance maladie : Chaque jour, elle reçoit les cotisations de ses clients et paie les factures reçues des ambulances, hôpitaux, et cliniques.

1. Notez que l'on appelle "jour", une itération de la simulation.

Le diagramme présenté dans la Figure 1 résume de façon simplifiée ces interactions. Une synthèse des relations entre entités est également disponible en annexe de ce document.

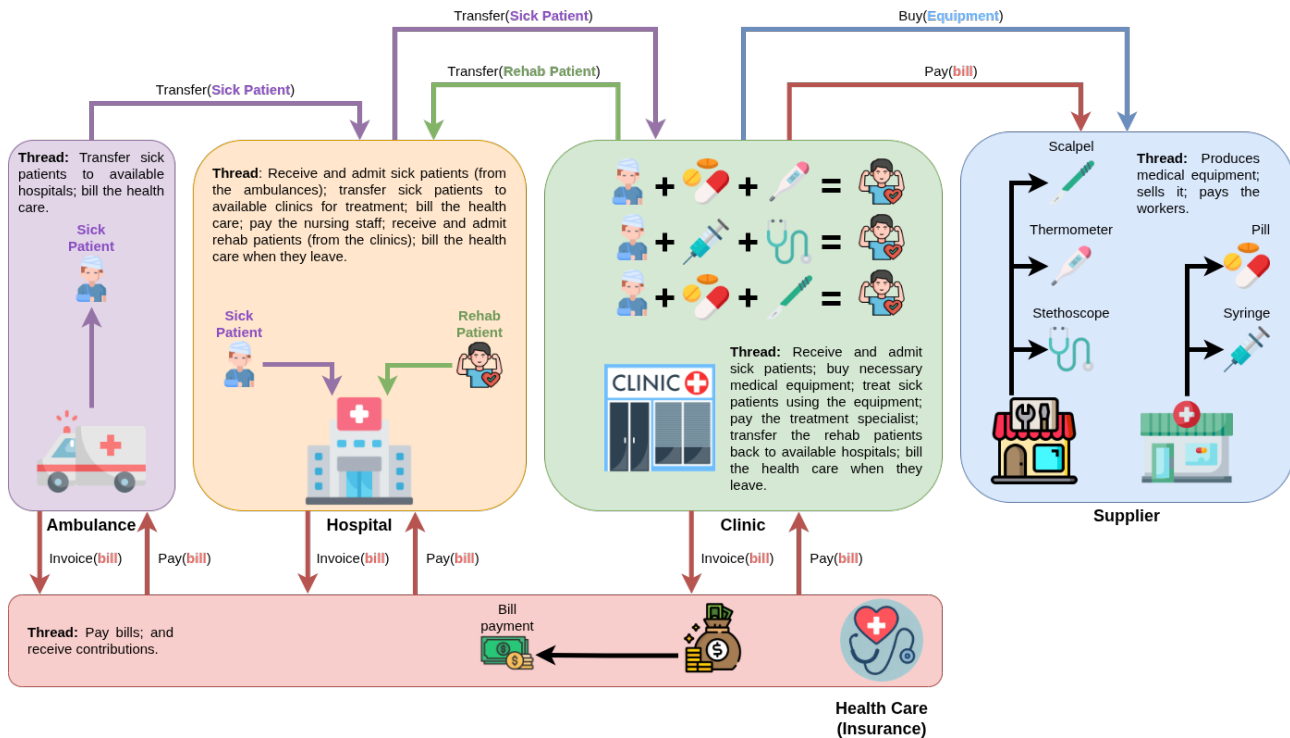


FIGURE 1 – Visualisation simplifiée des cinq acteurs principaux et de leurs interactions dans la simulation.

### 3 Description du fonctionnement

Dans ce laboratoire, vous devrez implémenter la logique régissant les interactions entre les différents acteurs tout en prenant en compte les sections critiques. Il vous sera demandé de compléter les méthodes fournies pour les ambulances, fournisseurs, hôpitaux, cliniques et assurances, selon le comportement décrit ci-dessus. Il est important de garder vos sections critiques les plus courtes possible et de ne protéger que le code nécessaire. Le code à écrire se trouve dans les fichiers suivants :

- `assurance.cpp/h`
- `ambulance.cpp/h`
- `hospital.cpp/h`
- `clinic.cpp/h`
- `supplier.cpp/h`
- `utils.cpp`
- `seller.h`

Dans le code source se trouvent des mots clés `TODO` en commentaires qui peuvent vous aiguiller sur les emplacements où il faut ajouter du code. Il peut être utile aussi de regarder les autres fichiers. **N'hésitez pas à observer les fichiers .h de chacune des classes afin de voir les méthodes et attributs mis à disposition.** Vous pouvez ajouter des méthodes et attributs membres aux classes, mais ne modifiez pas les signatures existantes ni l'architecture proposée. **Ne créez pas de nouveaux fichiers.**

## 3.1 Spécificités des classes

### 3.1.1 Interface vendeur (Seller)

L'interface vendeur (Seller) est une classe abstraite dont héritent tous les acteurs de la simulation. Les méthodes `buy`, `pay`, `transfer`, `invoice` font partie de celles qui doivent être implémentées dans les acteurs. Leur prototype dans les fichiers `.h` sont commentés.

### 3.1.2 Assurance (Insurance)

- La cotisation perçue chaque jour par l'assurance est une valeur constante : `INSURANCE_CONTRIBUTION`.
- La variable de classe `unpaidBills` est un vecteur de paire : `<Seller*, int>`. Il permet de stocker les informations des factures à payer. Une facture ne doit être payée que si l'assurance a les fonds nécessaires.

### 3.1.3 Ambulance (Ambulance)

- Un ambulancier peut transporter plusieurs patients à la fois et n'est payé qu'une fois pour le transfert.
- La variable de classe `nbEmployeesPaid` doit être incrémentée de 1 à chaque transfert effectué.
- Le transfert n'est possible que si le salaire de l'ambulancier peut être payé.

### 3.1.4 Hôpital (Hospital)

- La variable de classe `nbNursingStaff` est initialisée à la création d'un hôpital et est égale au nombre maximum de lits.
- Les hôpitaux peuvent s'endetter en payant les salaires. Si un hôpital a un solde négatif, il n'accepte plus de nouveaux transferts de patients. Pour le reste, il se comporte normalement.
- La variable de classe `nbEmployeesPaid` doit être incrémentée de `nbNursingStaff` à la fin de chaque journée.
- La variable de classe `nbFreed` doit être incrémentée à chaque fois qu'un patient a terminé sa période de convalescence et qu'il quitte l'hôpital.

### 3.1.5 Clinique (Clinic)

- Lors de l'achat d'une ressource médicale auprès d'un fournisseur, la clinique reçoit une facture qu'elle ajoute à sa pile de factures en attente.
- Si une clinique a des factures en attente, elle ne peut pas faire de nouvelle commande de matériel médical. Pour le reste, elle se comporte normalement.
- Le traitement d'un patient n'est possible que si le salaire du médecin spécialiste peut être payé.
- La variable de classe `nbEmployeesPaid` doit être incrémentée de 1 après le traitement de chaque patient.
- La variable de classe `unpaidBills` est un vecteur de paire : `<Supplier*, int>`. Il permet de stocker les informations des factures à payer. Une facture ne doit être payée que si la clinique a les fonds nécessaires.
- La variable de classe `resourcesNeeded` est un vecteur de `ItemType`. Il permet de consulter la liste des ressources médicales nécessaires au traitement des patients de cette clinique.

### 3.1.6 Fournisseur de matériel médical (Supplier)

- La production d’une nouvelle ressource n’est possible que si le salaire de l’employé peut être payé.
- La variable de classe `nbEmployeesPaid` doit être incrémentée de 1 après la production d’une nouvelle ressource.

## 3.2 Spécificités globales

- Le salaire d’un employé est accessible en utilisant `getEmployeeSalary (EmployeeType)`.
- Le prix d’une ressource est accessible en utilisant `getCostPerUnit (ItemType)`.
- Le coût d’un service est accessible en utilisant `getCostPerService (ServiceType)`.
- Chaque acteur possède les attributs `stock (std::map)` et `money (int)` qui doivent toujours être tenu à jour.

## 3.3 Gestion de la concurrence

Chaque acteur possède ses propres sections critiques. Utilisez les *mutex* de `<pcosynchro/pcomutex.h>` pour protéger ces sections.

### Règles simples (à suivre strictement).

- **Ne jamais détenir deux mutex à la fois.** Si vous devez appeler une méthode d’un autre acteur, *relâchez d’abord* votre mutex.
- Maintenir les **sections critiques courtes** (juste ce qui touche aux données partagées).

## 3.4 Conseils d’implémentation

1. Implémentez la gestion de la fin de la simulation dans la fonction `endService` de `utils.cpp` et faites les changements nécessaires dans les méthodes `run` des acteurs.
2. Implémentez les règles de la simulation.
3. Protégez les sections critiques.

## 4 Test

---

Afin de vérifier le bon déroulement de la simulation, plusieurs tests unitaires vous sont fournis. L’exécutable pour démarrer les tests se nomme `unit_tests`. Vous êtes libre d’ajouter autant de tests que vous le souhaitez. Si vous choisissez d’en créer, nous vous demandons d’en décrire le contenu ainsi que les résultats dans votre rapport. Aucun point négatif ne sera attribué à votre évaluation pour vos tests.

## 5 Travail à rendre

---

Créez des en-têtes dans les fichiers modifiés avec vos prénoms et noms.

- Les modalités du rendu se trouvent dans les consignes de laboratoire.
- **Utilisez le script de rendu fourni** pour préparer votre dépôt sur Cyberlearn.
- Décrivez l'implémentation, ses étapes, vos vérifications/ tests et infos pertinentes dans un **mini-rapport**. Décrivez notamment vos choix concernant les sections critiques.
- Travail en binôme obligatoire.

## 6 Barème de correction

---

Conception (rapport)	40%
Implémentation (fonctionnement, gestion correcte de la concurrence)	40%
Documentation, commentaires, bonnes pratiques (code)	20%



	<b>Paielements journaliers</b>	<b>Paielements ponctuels</b>	<b>Émission de factures</b>	<b>Factures à payer</b>	<b>Transfert envoi</b>	<b>Transfert réception</b>
<b>Ambulance</b>	-	Salaire de l'ambulancier lorsqu'il termine un transfert (contractuels)	Pour l'assurance après un transfert d'un patient malade vers un hôpital	-	Patients malades vers les hôpitaux	-
<b>Hôpital</b>	Salaire fixe du staff médical	-	Pour l'assurance après un transfert d'un patient malade vers la clinique; Pour l'assurance après la sortie d'un patient qui a terminé sa convalescence	-	Patients malades vers les cliniques	Patients malades depuis les ambulances; Patients traités depuis les cliniques Uniquement si l'hôpital n'est pas endetté
<b>Clinique</b>	-	Salaire du médecin spécialiste lorsqu'il traite un patient malade (contractuel)	Pour l'assurance après un transfert d'un patient traité et en convalescence vers l'hôpital	Achat de matériel médical auprès d'un fournisseur	Patients traités et en convalescence vers les hôpitaux	Patients malades depuis les hôpitaux (uniquement si la clinique n'est pas endettée et qu'aucune facture n'est en attente)
<b>Fournisseur de matériel médical</b>	-	Salaire de l'employé pour la production d'une nouvelle ressource (contractuel)	Pour les cliniques qui achètent du matériel médical	-	-	-
<b>Assurance</b>	-	-	-	Paielement du service des ambulances; Paielement du service des hôpitaux; Paielement du service des cliniques	-	-

TABLE 1 – Résumé des flux financiers et de transferts pour chaque entité