

# Le CNAM Paris

## FIP1 - mini Projets

L. Mostefaoui  
lotfi.mostefaoui@lecnam.net

## 1. Simulation d'une usine (producteur-consommateur avancé)

### Contexte :

- Simulez une usine avec plusieurs machines (producteurs) et entrepôts (consommateurs).
- Chaque machine produit des objets et les place dans un buffer partagé.
- Chaque entrepôt récupère les objets depuis ce buffer pour les stocker.

### Fonctionnalités à implémenter :

- Gestion du buffer circulaire avec des sémaphores et des mutex.
- Limitation du nombre de ressources disponibles (machines ou entrepôts) avec des sémaphores.
- Affichage en temps réel de l'état du buffer (objets produits, objets consommés).

### Objectifs pédagogiques :

- Comprendre la communication entre threads.
- Appliquer le modèle producteur-consommateur.

---

## 2. Gestion des impressions en réseau

### Contexte :

- Implémentez un gestionnaire d'impression pour un réseau local.
- Plusieurs utilisateurs (threads ou processus) envoient des tâches d'impression à une file d'attente.
- Une imprimante (représentée par un thread) traite les tâches une par une.

### Fonctionnalités à implémenter :

- Gestion de la file d'attente avec une structure partagée.
- Synchronisation des accès avec des mutex.
- Ajout d'une priorité pour certaines tâches urgentes.
- Simulation du temps d'impression pour chaque tâche.

## Objectifs pédagogiques :

- Gestion de ressources limitées.
  - Application des mutex et des sémaphores.
- 

### 3. Système de contrôle d'accès biométrique (IPC avec tubes)

#### Contexte :

- Simulez un système de contrôle d'accès à un bâtiment.
- Le processus principal gère les utilisateurs, et des processus enfants simulent des lecteurs biométriques.

#### Fonctionnalités à implémenter :

- Communication entre le gestionnaire et les lecteurs via des **tubes nommés**.
- Authentification des utilisateurs avec des identifiants simulés.
- Journalisation des accès dans un fichier.

## Objectifs pédagogiques :

- Communication inter-processus (IPC).
  - Gestion des fichiers et synchronisation.
- 

### 4. Simulation d'un carrefour routier intelligent (signaux et threads)

#### Contexte :

- Implémentez un système pour gérer un carrefour avec 4 routes.
- Les voitures (threads) doivent attendre un feu vert pour passer.
- Le système doit contrôler les feux tricolores.

#### Fonctionnalités à implémenter :

- Synchronisation des voitures avec des signaux ou sémaphores.
- Gestion des feux de circulation avec des threads.
- Simulation de priorités (par exemple, ambulances).

## **Objectifs pédagogiques :**

- Synchronisation avancée avec signaux.
  - Gestion multi-threadée des événements.
- 

## **5. Serveur de chat multiclients (sockets et threads)**

### **Contexte :**

- Implémentez un serveur de chat où plusieurs clients peuvent se connecter.
- Les messages des clients doivent être diffusés à tous les autres clients.

### **Fonctionnalités à implémenter :**

- Utilisation des sockets pour la communication réseau.
- Un thread par client pour gérer les connexions simultanées.
- Journalisation des messages échangés.

## **Objectifs pédagogiques :**

- Introduction aux sockets.
  - Gestion des threads dans un serveur.
- 

## **6. Système de commande de restaurant (mutex et sémaphores)**

### **Contexte :**

- Simulez un restaurant où des serveurs prennent des commandes, et des cuisiniers préparent les plats.

### **Fonctionnalités à implémenter :**

- Les commandes sont ajoutées à une file d'attente.
- Les cuisiniers récupèrent les commandes pour les traiter.
- Synchronisation avec des sémaphores pour limiter les commandes en attente.

## **Objectifs pédagogiques :**

- Modèle producteur-consommateur.
  - Synchronisation avancée des threads.
-

## 7. Simulation d'un ascenseur (threads et mutex)

**Contexte :**

- Implémentez un système d'ascenseur pour un bâtiment de 5 étages.
- Les utilisateurs (threads) demandent à monter ou descendre.

**Fonctionnalités à implémenter :**

- Gestion des requêtes d'appel et de destination.
- Simulation du mouvement de l'ascenseur.
- Synchronisation des utilisateurs pour accéder à l'ascenseur.

**Objectifs pédagogiques :**

- Synchronisation avancée des threads.
  - Gestion des événements concurrents.
- 

## 8. Gestionnaire de tâches système (IPC et threads)

**Contexte :**

- Créez un gestionnaire pour exécuter et surveiller des tâches (processus).
- Le gestionnaire crée des processus enfants pour chaque tâche et communique avec eux.

**Fonctionnalités à implémenter :**

- Communication entre processus avec des tubes ou des signaux.
- Affichage de l'état des tâches (en cours, terminé, etc.).
- Journalisation dans un fichier.

**Objectifs pédagogiques :**

- Utilisation des IPC (tubes ou signaux).
- Gestion de processus multiples.