

## Projet SY40 : la confiturerie

Une confiturerie dispose d'une chaîne pour le remplissage des bocaux. Un processus **Bocal** contrôle une machine qui assure l'alimentation (l'arrivée) des bocaux. Un autre processus **Valve** contrôle l'ouverture et la fermeture de la valve d'une unité de remplissage. La valve ne doit être ouverte qu'après l'arrivée d'un bocal. Le processus **Valve** doit utiliser un processus **Horloge** pour mesurer un délai de 30 secondes, après lequel il ferme la valve. Il doit alors signaler au processus Bocal que le remplissage est terminé, et qu'il peut enlever le bocal et en placer un nouveau.

<u>Processus Bocal</u>	<u>Processus Valve</u>	<u>Processus Horloge</u>	
Début	Début	Début	
Tant que Vrai alors	Tant que Vrai alors	Tant que Vrai alors	
Place un Bocal	Ouvre la valve		
.....	.....	Compte_30s	
Enlève le Bocal	Ferme la valve	.....	
.....	Fin Tant que	Fin Tant que	
Fin	Fin	Fin	

Développez une solution fonctionnelle à ce problème en tenant compte des aspects de synchronisation.

**a) Implémentez la solution en utilisant des processus et des sémaphores IPC**

b) On ajoute une autre machine d'alimentation en bocaux, processus Bocal\_2. Il faut s'assurer qu'un seul processus à la fois, Bocal ou Bocal\_2, place un bocal dans l'unité de remplissage. De plus, les nouveaux bocaux (processus Bocal\_2) sont deux fois plus grands que les anciens. Le processus Valve doit donc obtenir un délai deux fois plus long, e.g. 60 sec (pour les nouveaux bocaux seulement). Pour cela, il devra se synchroniser pour fermer la valve après deux itérations du processus Horloge (le délai dans Horloge est fixe et ne peut être changé). On vous demande de modifier les processus pour assurer une bonne synchronisation pour ce nouveau système. Complétez les trois processus pour assurer la synchronisation appropriée.

c) Développez une solution fonctionnelle à ce problème en tenant compte des aspects de synchronisation. Et implémenter la solution en utilisant des threads et des moniteurs

### Contexte :

Je vous demande de transposer ce sujet en utilisant les concepts vus pendant les cours. Cette transposition se voudra cohérente et sa mise en œuvre respectera la logique suivante :

- Utilisation des mécanismes de communication et de synchronisation présentés en cours
- Implémentation effective, debug & tests.
- Conseils d'implémentation :
  - o Pensez Programme modulaire ;
  - o Attention aux objets résiduels (Posix & IPC) ;
  - o Attention à l'indentation du code pour permettre une meilleure lisibilité du code ;
  - o Attention aux fuites mémoire et utilisation des pointeurs ;
  - o Favoriser le caractère dynamique de l'application ;
  - o Utilisation de bibliothèques et de plusieurs fichiers sources conseillés avec un makefile.

Pour les soutenances :

- o Préparez vos jeux de tests et vos cas d'usage AVANT la soutenance.
- o Les soutenances seront organisées comme suit : 10 minutes de présentation de la problématique, pas de support de présentation de type Powerpoint. La veille de votre soutenance, vous devez transmettre par courriel le rapport du projet en format PDF (Pas de rapport papier) ainsi que les sources compressées et le jeu de données (+ Makefile) soit à [philippe.descamps@utbm.fr](mailto:philippe.descamps@utbm.fr) ou à [jocelyn.buisson@utbm.fr](mailto:jocelyn.buisson@utbm.fr)