Systèmes temps réel

# I - Study of an assembly station

## Question 1

Il risque d’y avoir un problème de synchronisation entre les deux robots et le plateau tournant.
Probleme de producteur/consommateur

## Question 2

Dans la configuration suivante, le robot A n’effectue que l’action Déposer et le robot B n’effectue que l’action de prendre.

Ne pas prendre un objet si le bac est vide.

Ne pas déposer dans un bac si tous les bacs sont pleins.

=> On a donc besoin de deux semaphores. *c’est pas des mutex ? NOoowpe”*

Chaque sémaphore représente des ressources disponibles.

On a besoin de vérifier qu’on a bien quelque chose à déposer, on vérifie qu’on a bien une ressource.

Semaphore A représente le nombre de bac vide P(place\_vide)

Semaphore B représente le nombre de bac plein P(place\_pleine)

|  |  |
| --- | --- |
| Robot\_A | Robot\_B |
| P(place\_vide)Deposer\_Ra()V(place\_pleine) | P(place\_pleine)Prendre\_Rb()V(place\_vide) |

## Question 3

Contraintes :

* Si on fait une action (prendre/déposer), le plateau ne peut pas tourner : on doit examiner le plateau.
* Chaque robot n’a accès qu’à un seul bac à la fois : on peut déposer et prendre simultanément
* On peut examiner et faire une autre action en même temps

Fonction Déposer\_Ra() :

Deposer\_Ra()

{

 P(plateau); //Pour empêcher les deux d’examiner en même temps et examiner en tournant

 While(examiner() != LIBRE)

 {

 P(rotation\_plateau\_poser);

 Tourner();

 V(rotation\_plateau\_poser);

 }

 P(rotation\_plateau\_prendre);

 V(plateau);

 Poser\_la\_piece();

 V(rotation\_plateau\_prendre)

}

---------------------------------------------------------------------

Fonction Prendre\_Rb() :

Prendre\_Rb()

{

 P(plateau) ;

 while (examiner() != OCCUPE)

 {

P(rotation\_plateau\_prendre);

Tourner();

v(rotation\_plateau\_prendre);

 }

 P(rotation\_plateau\_poser);

 V(plateau);

 Enlever\_la\_piece();

 V(rotation\_plateau\_poser) ;

}