**TD 1**

1. **Etude d’un poste d’assemblage**

**Question 1**

Problème producteur/consommateur => 2 sémaphores. (pas exclusion mutuelle)

Traiter d’abord les cas particuliers (tapis vide, plateau plein etc). Ensuite on rentre dans les fonctions et on synchronise

Producteur/consommateur :

Déposer => |\_\_|\_\_|\_\_|\_\_|\_\_|\_\_|\_\_|\_\_|\_\_| => prendre, lire

Limité (tampon circulaire)

Fonctions : déposers(), prendre()

Vérifier si on a une case de libre.

Trouver dans le texte ce qui est une ressource. On veut déposer = consommer une ressource case libre => case libre-- => moins de ressources cases libres disponibles

**Question 2**

Bac\_libre = 4

Bac\_occ = 0

Sémaphore = nombre de cases libres. P décrémente le sémaphore

|  |  |
| --- | --- |
| P(cases\_libres) = diminue le nombre de cases occupées | P(cases\_occupées) = diminue le nombre de cases occupées |
| Deposer\_Ra() | Prendre\_Rb() |
| V(cases\_occupées) = augmente le nombre de cases occupées | V(cases\_libres) = augmente nombre de cases libres |

Chercher P en premier (blocage). Si P atteint 0, bloque le processus

Prendre = consommer place occupée

**Question 3**

Un seul peut faire tourner en même temps. Deux peuvent prendre/poser en même temps. Deux peuvent examiner en même temps mais pas examiner et tourner en même temps, ni tourner et prendre ou tourner et poser.

Deposer\_Ra()

{

 P(m3)

 While(examiner() != LIBRE)

 {

 P(m1)

 Tourner();

 V(m1)

 }

 P(m2)

 V(m3)

 Poser\_la\_piece();

 V(m2)

}

Prendre\_Rb()

{

 P(m3)

 While(examiner != OCCUPE)

 {

 P(m2)

 Tourner();

 V(m2)

 }

 P(m1)

 V(m3)

 Enlever\_la\_piece();

 V(m1)

}