# SETR TD 1

## Exercice 1 :

Donner le pseudo code réalisant le modèle Producteur Consommateur à partir des fonctions et des objets élémentaires de SCEPTRE.

Notes de l’oral

* Producteur ne peux pas produire si buffer plein
* Consommateur doit être bloqué si la file est vide.
* 2 tâches : Prod (boucle sur l’envoie) & consommateur (boucle sur la réception)

On suppose qu’on va avoir 2 tâches, P1 & P2, qui vont incrémenter un compteur.

A -> Mx = 4

(On imagine que P2 se lance après la ligne ADD A +1)

P1 : P2 :

Acc A = 4 LDA A <- Mx LDA A <- Mx Acc A = 4

Acc A = 5 ADD A +1 ADD A +2 Acc A = 6

Acc A = 5 STA A -> Mx STA A -> Mx Acc A = 6

A = Mx = 5

Section critique.

P(V) = bloquant si section critique occupée, passante sinon.

Sémaphore :

* Verrou : Boolean (vrai / faux = section critique occupée)
* Acompte :

Exercice 2 : Prendre vendre

En pseudo code

P(s) :

cpt-- ; Ne peut être interrompu (si un autre processus décrémente ce n’est pas bon)

Si cpt < 0 Alors

< Placer processus en file d’attente >

Fsi

V(s)

cpt++ ;

Si cpt <= 0 alors

< Réveiller un processus en file d’attente >

Fsi

En sémaphore : avec des primitives élémentaires

P(s)

ENTRER (R1)

cpt = cpt -1

Si cpt < 0 alors

ENVOYER (TACHE-COURANTE, F)

EFFACER(E1)

SORTIR(R1)

ATTENDRE (E1) <- reste bloquée ici

Else

SORTIR (R1

Fsi

V(s)

ENTRER (R1)

cpt = cpt +1

si cpt <=0 alors

RETIRER (T,F)

SIGNALER (E1, T)

Fsi

SORTIR (R1)

Pour le producteur :

Prod : (cpt1 = n)

P(s1)

Ecrire dans mémoire

V(s1)

Conso : (cpt2=0)

P(s2)

Lire mémoire

V(s2)

# Ordonnancement :

## Exercice 1 :