



## Module SETR TD n° 1

### Projet SCEPTRE :

#### Exercice 1:

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

#### Exercice 2:

Donner le pseudo code réalisant les fonctions Prendre et Vendre des sémaphores à partir des Primitives et des Objets élémentaires de SCEPTRE.

### Ordonnancement :

#### Exercice 1 :

Soient trois tâches A, B et C périodiques et indépendantes ayant les caractéristiques suivantes :

A ( $C=2$ ,  $r_0=0$ ,  $R=T=6$ ); B ( $C=2$ ,  $r_0=0$ ,  $R=T=8$ ); C ( $C=3$ ,  $r_0=0$ ,  $R=T=12$ )

Donner l'ordonnancement obtenu sur une période de 24 unités de temps (PPCM des périodes des 3 tâches) pour les politiques d'ordonnancement R.M., E.D.F. et L.L.F.

#### Exercice 2 :

On considère un système constitué de deux tâches périodiques indépendantes A et B, s'exécutant sur un seul processeur. Ces deux tâches sont définies par :

TA : ( $r_0=0$ ,  $C_a=2$ ,  $R_a=4$ ,  $T_a=4$ )

TB : ( $r_0=0$ ,  $C_b=5$ ,  $R_b=10$ ,  $T_b=10$ )

Ces deux tâches sont-elles ordonnanciables par EDF?, par RM?

On utilisera le test de terminaison pour vérifier le respect des échéances pour RM.

Conclusions?

#### Exercice 3:

Soit un système composé de deux tâches périodiques indépendantes :

T1 : ( $C_1=6$ ,  $R_1=T_1=12$ ) et T2 : ( $C_2=15$ ,  $R_2=T_2=30$ )

- Les tâches sont ordonnancées avec EDF, y a-t-il des échéances ratées?
- Même question avec RM. Que conclure?

#### Exercice 4 :

Un système possède une tâche d'arrière-plan dont l'exécution demande 100 ms. La tâche d'avant-plan s'exécute toutes les 50 ms et son exécution demande 25 ms. Chaque changement de contexte est effectué en 100  $\mu$ s.

Modéliser ce problème sous la forme d'une configuration de tâches d'un système Temps Réel. Donner les caractéristiques des deux tâches. Les éventuelles contraintes de temps sont-elles respectées.

Donner la date de fin de la tâche de fond en utilisant le théorème de la zone critique.

## Le Projet SCEPTRE

### 2.2. Les Opérateurs T. R. de SCEPTRE

#### ◆ Les primitives définies par SCEPTRE

CATEGORIE	OPERATION ELEMENTAIRE	PARAMETRES	FONCTION
GESTION DES TACHES	DEMARRER	TACHE	Lancer l'exécution de la tâche
	ARRETER	TACHE	Arrêter la tâche
	CONTINUER	TACHE	Continuer l'exécution de la tâche
	SE TERMINER		Terminer l'exécution de la tâche qui l'exécute
	CHANGER-PRIORITE	TACHE, nouvelle priorité	Donner à la tâche la nouvelle priorité
	ETAT	TACHE	Fournir l'état de la tâche
	PRIORITE	TACHE	Fournir la priorité de la tâche
	TACHE COURANTE		Fournir l'identité de la tâche qui l'exécute
	STATUT		Fournir le statut de l'opération élémentaire qui vient d'être exécutée par la tâche
SIGNALISATION	SIGNALER	EVENEMENT, TACHE	Signaler à la tâche que l'événement est arrivé
	ATTENDRE	Liste d'EVENEMENTS	Attendre que l'un (au moins) des événements de la liste soit arrivé
	ARRIVE	Liste d'EVENEMENTS	Prédicat qui est vrai si tous les événements de la liste sont arrivés
	EFFACER	Liste d'EVENEMENTS	Remettre les événements de la liste dans l'état non arrivé

## Le Projet SCEPTRE

### 2.2. Les Opérateurs T. R. de SCEPTRE

#### ◆ Les primitives définies par SCEPTRE

CATEGORIE	OPERATION ELEMENTAIRE	PARAMETRES	FONCTION
COMMUNICATION ENTRE TACHES	ENVOYER	ELEMENT, FILE	Envoyer l'élément à la queue dans la file
	RETIRER	ELEMENT, FILE	Retirer le premier élément de la file
	VIDE	FILE	Prédicat qui est vrai si la file est vide
	PLEINE	FILE	Prédicat qui est vrai si la file est pleine
EXCLUSION MUTUELLE	ENTRER	REGION	Demander la propriété exclusive de la région
	SORTIR	REGION	Relâcher la propriété exclusive de la région