

# TD2 - Conception du système d'E/S

ADENOT Paul, BRODU ETIENNE, GAUDIN Maxime,  
GOLUMBEANO Monica, RICHARD Martin

1<sup>er</sup> novembre 2010

## Table des matières

1	Q2	2
2	Q3	2
3	Q5	2

## Résumé

Réponse aux 5 questions du TD2 :

1. Compléter le schéma
2. Démontrer comment on peut "fabriquer" le mode *synchrone* sur le mode *asynchrone*
3. « Il y un bug » : Reprendre la solution en mode asynchrone et poser vous la question « Comment le processus appelant récupère l'@IORB pour le libérer ? Proposer une solution
4. **p.14**, est-ce que le champ « N tâche responsable de la requête d'E/S » de la structure de données IORB est nécessaire ? Justifier
5. En algorithmie de haut niveau, donner l'algo de la commande shell `ls`

## 1 Q2

Le mode synchrone peut être construit sur le mode asynchrone en plaçant simplement un point de synchronisation sur la ressource (au sens large). On placera alors juste un appel à `WAIT` sur le sémaphore *requête traité*, juste après l'appel de `MONES`. On aura là le comportement d'un mode synchrone : l'exécution sera bloquée entre l'appel à la primitive d'entrée sortie et le moment où cette primitive d'entrée sortie retournera.

## 2 Q3

L'adresse du buffer de l'échange est contenu dans la structure `tab_ech` et dans `l_IORB`, `l_IORB` n'est donc pas utile dans le contexte du processus utilisateur. Le processus utilisateur ne doit pas pouvoir y accéder, d'autant plus que ce n'est pas lui qui l'instancie. Cependant, la procédure `MONES_ASYN` ne peut pas se charger de libérer `l_IORB` car il est utilisé hors du contexte de la procédure. La tâche d'interruption en RMX devras donc libérer `l_IORB` lorsque elle n'en a plus besoin, après avoir envoyé le signal sur le sémaphore requête servie.

Attention, il peut apparaître un conflit avec le numéro de la tâche responsable de l'échange puisque il est potentiellement appelé après la destruction de `l_IORB` tel que décrite précédemment. La question 4 répondra de l'utilité de ce numéro.

## 3 Q5

**Algorithme** Procedure LS

**Constante**

    TAILLE\_POSTE = Taille maximum d'un poste

**Variable**

    RepCourant : *Nom du répertoire courant*

    Fichiers : *Tableau de descripteurs de fichier du répertoire courant*

    N : *Nombre de postes du répertoire courant*

**Debut**

    Initialiser RepCourant au répertoire courant

    Ouvrir le descripteur de fichier associé à RepCourant

    N ← Nombre de poste de RepCourant

    Initialiser Fichier ( Taille : N \* TAILLE\_POSTE )

**Pour** F allant de 1 à N **Faire**

        Fichiers[F] ← Descripteur du poste F

        Afficher sur la sortie standart les attributs de Fichier[F]

**FinPour**

    Libérer la mémoire

**Fin**