TD2 - Conception du système d'E/S

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

5 novembre 2010

Table des matières

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 ls 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 ou 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 tache d'interruption en RMX devras donc libérer l'IORB lorsque elle n'en à 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 précedemment. La question 4 répondra de l'utilité de ce numéro.

3 Q4

La tâche responsable de la requête d'E/S va créer un sémaphore de synchronisation, initialisé à 0 et le transmettre à MONES_ASYNC. C'est l'attente d'un jeton dans ce sémaphore qui va permettre de synchroniser la fin de la requête d'E/S avec la tâche qui la demande. L'IORB contient l'adresse de ce sémaphore. La tâche d'intéruption qui traite la requête ajoutera un jeton dans ce sémaphore une fois l'opération d'E/S réalisée.
~

Le numéro de la tâche responsable de la requête n'est pas utilisé par le système de gestion d'entrées/sorties du système, la seule communication effective entre le système d'E/S et la tâche est la synchronisation finale. Ainsi, le sémaphore "requête servie" suffit pour remplir ce rôle.i

4 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 courrant
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
```