C. KAISER CONSERVATOIRE NATIONAL DES ARTS ET METIERS

COURS SYSTEMES ET RÉSEAUX INFORMATIQUES B
SYSTEMES INFORMATIQUES

MEMOIRE COMMUNE :
CONTRÔLE DE CONCURRENCE PAR SÉMAPHORES
EXEMPLES DE PROGRAMMES

•••••• TABLE DES MATIERES

50 Schémas sémaphores privés et sémaphores collectifs

PHILOSOPHES_V1

PHILOSOPHES_V2

PHILOSOPHES_V_TABL

60 Schemas d'allocation de ressources

ALLOC_MEMOIRE_V1_PRIV

ALLOC_MEMOIRE_MODULE_PRIV

ALLOC_DISQUE_V1_PRIV

70 Gestion de transfert disque

ALLOC_DISQUE_V1_COLL

•••••• LE PAQUETAGE GENERAL STANDARD SI_B

Ce paquetage fournit des types et des procédures standard utilisables pour le cours.

On l'invoque par:

with SI_B; use SI_B;

On peut alors utiliser les types abstraits suivants:

a)-des sémaphores avec les opérations P, V, E0 --voir le cours

b)-des objets partagés implantés chacun dans une cellule de mémoire qui est accessible par une opération de lecture, appelée VALUE_OF, d'écriture, appelée SET, ou d'échange atomique, appelée EXCHANGED_VALUE_OF. Les objets partagés standard sont:

des entiers du type SHARED_INTEGER,

des booléens du type SHARED_BOOLEAN,

des caractères du type SHARED_CHARACTER.

Ce paquetage fournit encore:

c)-des entiers partagés du type SHARED_CC, qui sont pourvus des opérations VALUE_OF et SET et qui produisent un effet de bord à chaque opération SET. cet effet de bord consiste à imprimer un * dans la colonne X de la ligne courante du fichier de sortie standard, avec X égal à la valeur écrite par SET.

d)- des opérations pour imprimer des résultats dans le fichier de sortie standard. Ce sont des écritures d'un caractère, d'une chaîne ou d'un entier, toutes appelées PUT, le passage à la ligne NEW_LINE et la détermination de la colonne courante SET_COL. Ces opérations sont des renommages du module standard TEXT_IO de ADA.

e)- une fonction NUMEROTATION qui fournit des numéros distincts à chacun des processus d'un groupe

f)- des fonctions sur des vecteurs et des tirages aléatoires.

REMARQUE : Les sémaphores sont simulés avec une tâche serveur et les appels de P(S), V(S) et E0(S, X) sont des rendez-vous vers la tâche qui simule le sémaphore S. Ainsi ces appels sont des points de synchronisation (au sens de la sémantique du langage Ada). Ceci entraîne que les variables partagées, qui peuvent être implantées avec une copie locale à chaque tâche (c'est autorisé en Ada, pour pouvoir faire des optimisations) sont alors mises à jour(c'est à dire que toutes les copies reçoivent alors la même valeur).

EXTRAIT DU MANUEL ADA.

TÂCHES

Les tâches sont des entités dont lesexécutions se déroulent en parallèle au sens suivant.
On peut considérer que chaque tâche est exécutée par un processeur logique qui lui est propre.
Différentes tâches se déroulent indépendamment, sauf en des points où elles se synchronisent.

VARIABLES PARTAGÉES

Si deux tâches lisent ou mettent à jour une variable partagée (c'est à dire une variable accessible par les deux tâches), alors aucune d'elles ne peut supposer quoi que ce soit sur l'ordre dans lequel l'autre effectue ses opérations, sauf aux points où elles se synchronisent). Entre deux points de synchronisation d'une tâche qui lit ou met à jour une variable partagée, une implémentation a le droit d'utiliser une copie locale (registre, cache) de cette variable.

POINTS DE SYNCHRONISATION

Les principaux points de synchronisation sont :
le début d'un rendez-vous entre deux tâches,
la fin d'un rendez-vous entre deux tâches,
le début d'une täche,
la terminaison d'une tâche.

Si on veut que chaque accès à une variable partagée Var soit un point de synchronisation des copies de cette variable partagée, il faut utiliser le pragma Atomic et déclarer var comme atomique avec : pragma atomic(Var).