

ED SYNCHRONISATION

1. Section critique

La programmation de la réservation de place d'avions peut s'envisager suivant deux méthodes. Le schéma général du programme serait le suivant:

```
var f : fichier;
      enrg : t_enregistrement;
début répéter
    lecture_terminal ( type, vol, date );
    cas type dans consultation: consulter;
      réservation: réserver;
      sortie;;
    fincas;
  jusqu'à type = sortie;
fin;
```

Pour les deux méthodes, la procédure `consulter` est la suivante:

```
procédure consulter
début début_consult;
  lire_enrg (f, vol, date, enrg);
  places_libres := enrg.nb_place;
  fin_consult;
  afficher ( " places disponibles: ", places_libres);
fin;
```

Dans la première méthode, la procédure `réserver` est la suivante:

```
procédure réserver;
début début_reserv;
  lire_enrg (f, vol, date, enrg);
  si enrg.nb_place > 0 alors
    afficher ( "il y a de la place; en voulez-vous une? " );
    lecture_terminal ( réponse );
    si réponse = "oui" alors
      enrg.nb_place := enrg.nb_place - 1;
      écrire_enrg (f, vol, date, enrg);
    finsi;
  sinon afficher ( "plus de place " );
  finsi;
  fin_reserv;
fin;
```

Dans la deuxième méthode, la procédure `réserver` est la suivante:

```
procédure réserver;  
var OK: booléen;  
début début_reserv;  
  lire_enrg (f, vol, date, enrg);  
  si enrg.nb_place > 0 alors  
    enrg.nb_place := enrg.nb_place - 1;  
    écrire_enrg (f, vol, date, enrg); OK := vrai;  
  sinon OK := faux;  
  finsi;  
  fin_reserv;  
  si OK alors afficher ( "place réservée ");  
  sinon afficher ( "plus de place ");  
  finsi;  
fin;
```

Le système fournit, pour chaque fichier ouvert, un mécanisme de verrouillage au moyen de deux opérations. Ces opérations n'ont aucun effet sur les opérations de lecture et d'écriture sur le fichier.

- procédure `verrouiller (f)` met le processus demandeur dans une file associée au fichier s'il y a déjà un processus qui a verrouillé le fichier, sinon pose un verrou sur le fichier.
- procédure `déverrouiller (f)` réactive un des processus de la file associée au fichier s'il y en a et enlève le verrou s'il n'y en a pas.
-

A- Que doivent être les opérations de `début_reserv` et `fin_reserv` dans chacune des méthodes? Justifiez.

B- Que doivent être les opérations de `début_consult` et `fin_consult` dans chacune des deux méthodes? Justifiez.

C- Comparer les avantages et inconvénients des deux méthodes, et dites celle qui vous semble préférable.

2. A propos d'interblocage

Dans le service de gestion d'un magasin, un employé est chargé d'enregistrer les commandes des clients dans un fichier COM, et d'éditer les bons de commandes correspondants, sur une imprimante IMP. D'autre part, un processus facturation, lancé périodiquement lit les commandes à facturer dans COM et édite les factures correspondantes sur l'imprimante IMP. Notons qu'une commande peut concerner plusieurs articles, et donc qu'un bon de commande ou une facture peut comporter plusieurs lignes.

A- Sachant que l'opération de transfert élémentaire (lecture, écriture) concerne une ligne de commande, montrer que l'état de COM peut être incohérent, de même que les impressions sur IMP. En déduire les règles d'accès à COM et à IMP.

B- Afin de résoudre le problème précédent, on fournit deux procédures qui garantissent un accès exclusif à une ressource R:

- `réserver(R)` autorise l'accès à R par le demandeur si R est libre ou bloque le demandeur si R est occupée.

- libérer(R) autorise l'accès à un autre demandeur s'il y en a en attente, sinon indique que R est libre.

On propose la solution suivante :

```
processus employé :  
début tant qu'il y a des commandes à enregistrer faire  
    réserver(COM);  
    enregistrer une commande dans COM;  
    réserver(IMP);  
    éditer un bon de commande sur IMP;  
    libérer(COM);  
    libérer(IMP);  
    fait;  
fin;  
processus facturation;  
début répéter indéfiniment  
    réserver (IMP);  
    réserver (COM);  
    tant qu'il y a des commandes à facturer dans COM faire  
        lire la prochaine commande dans COM;  
        éditer la facture correspondante sur IMP;  
    fait;  
    libérer(IMP);  
    libérer(COM);  
    attendre la prochaine période de facturation;  
    fait;  
fin;
```

B.1- Donner une définition de l'interblocage. Montrer que la programmation risque de conduire à un interblocage.

B.2- Modifier le processus de facturation pour supprimer le risque d'interblocage.

B.3- Montrer que cette programmation permet d'éditer de manière consécutive, toutes les factures pour une période donnée.