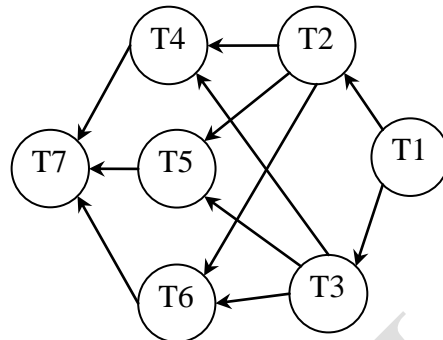


Exercice n°1 : (5 pts)

On considère le graphe de précédences suivant :



- Exprimer ce graphe à l'aide de Parbegin Parend.

Exercice 2 : (7 pts)

- Donner une solution pour le problème de plusieurs producteurs/ plusieurs consommateurs à l'aide des moniteurs avec conditions de kessels.

Exercice 3 : (7 pts= 1 + 2 + 5)

On considère un système contenant un nombre fixe N de processus et une seule classe de ressource à M instances. Chaque processus peut demander k exemplaires et la priorité est donnée à celui qui demande le moins d'instances. On désire gérer l'allocation de cette classe de ressources à l'aide des moniteurs classiques.

- A/ Donner la forme générale de chaque processus.
- B/ Donner les structures de données principales
- C/ Ecrire la solution correspondante.

Bon Courage

Correction de l'Epreuve Finale (23/24)

Exercice n°1 : (5 pts)

- Expression à l'aide de Parbegin Parend

```
Debut
T1 ;
Parbegin T2 ; T3 Parend ;
Parbegin T4 ; T5 ; T6 Parend ;
T7 ;
Fin.
```

Exercice 2 : (7 pts)

Forme générale d'un processus :

Processus Prod(i : entier) ;

var art :

Debut

```
-
M.dem_dep() ;
Deposer(art) ;
M.fin_dep()
-
```

-0,5pts-

Fin.

Processus Cons (i : entier) ;

var art :

Debut

```
-
M.dem_prel() ;
Prelever(art) ;
M.fin_prel() ;
-
```

Fin.

Solution 1 : On utilise 1 seul compteur

M : **Moniteur** ;

const N = ;

var cpt : entier ; pcours, ccours : boolean ; -0,25pts-

//cpt : compte le nombre d'articles dans le tampon.

entree procedure dem_dep () ;

Debut

wait((cpt < N) et non pcours) ; - 2pts-

pcours := vrai

Fin ;

entree procedure dem_prel () ;

Debut

wait((cpt > 0) et non ccours) ;

ccours := vrai - 2pts-

Fin ;

entree procedure fin_dep () ;

Debut

cpt := cpt + 1 ; pcours := faux

Fin ; -1pts-

entree procedure fin_prel () ;

Debut

cpt := cpt - 1 ; ccours := faux

Fin ; -1pts-

Initialisation

Debut

cpt := 0 ; pcours := faux ; ccours := faux -0,25pts-

Fin.

Solution 2 : On utilise deux compteurs

M : Moniteur ;

const N=.... ;

var np, nv : entier ; pcours, ccours : booleen ; **-0,25pts-**

//np et nv: comptent respectivement le nombre de cases pleines et vides.

entree procedure dem_dep () ;

Debut

wait((nv>0) et non pcours) ;

nv :=nv-1 ; pcours :=vrai

Fin ;

entree procedure dem_prel () ;

Debut

wait((np>0)et non ccours) ;

np :=np-1 ; ccours :=vrai

Fin ;

entree procedure fin_dep () ;

Debut

np :=np+1 ; pcours :=faux

Fin ;

entree procedure fin_prel () ;

Debut

nv :=nv+1 ; ccours :=faux

Fin ;

Initialisation

Debut

np :=0 ; nv :=N ; pcours :=faux ; ccours :=faux **-0,25pts-**

Fin.

Exercice 3 : (7 pts= 1 + 2 + 5)

A/ Forme générale d'un processus:

Processus P (i :entier) ;

ressources

Debut

Le processus peut libérer par parties les

préalablement acquises.

-

M.demander(k, i) // k est le nombre de ressources demandées.

< Utiliser les ressources >

M.liberer(k)

-

Fin.

B/ Les structures principales

- f : une liste explicite dont chaque élément est un enregistrement de structure id : identité du processus ayant une demande pendante et nb : le nombre d'exemplaires demandés. Cette liste est triée par ordre croissant du nombre d'instances demandées. A cette liste, sont associées les primitives suivantes :
 - insérer(f, i, k) insère un élément contenant i et k dans la liste en conservant l'ordre croissant.
 - extraire(f) : supprime le premier élément de la liste.
 - vide(f) : retourne vrai si la liste est vide.
 - premier(f) : retourne le premier élément de la liste.
- c : un tableau de conditions de taille N dont c[i] est la condition sur laquelle Pi se bloque.

C/ La solution :

M : Moniteur ;

const N=... ; M=...;

var c : Tableau [1..N] de condition ;

- 0,25pts-

dispo : entier ;

<déclaration de la file f + ses procédures d'accès>

entree procedure demander (k : entier, i : entier) ;

Debut

Si (dispo<k) **Alors** insérer (f, i, k) ; c[i].wait; extraire(f) **Fsi** ;

 dispo :=dispo-k ;

- 2,25pts-

Fin ;

entree procedure liberer (k : entier) ;

Debut

dispo :=dispo+k ;

Tant que non vide(f) et (dispo<=premier(f).nb) **Faire**

 c[premier(f).id].signal() ;

- 2,25pts-

Fait

Fin ;

Initialisation

Debut

dispo :=M ;

- 0,25pts-

Fin.