

SYSTEMES ET RESEAUX INFORMATIQUES EXAMEN DU 14 SEPTEMBRE 1999

partie portant sur l'enseignement des SYSTÈMES INFORMATIQUES CORRIGÉ

Question 1 (2 points)

On considère ci-dessous une programmation des procédures Prendre et Restituer du paquetage Memoire et on vous demande d'utiliser et d'initialiser les sémaphores Mutex et S_Stock pour prendre en compte la concurrence d'accès à ce paquetage.

```
with SRI_B; use SRI_B; with Table_Generique; use Table_Generique;
paquetage body Memoire is
  type D_A is (Disponible, Alloue);
  package Table is Table_Generique(Taille => 40, Element => D_A, Init => Disponible);
  -- initialement toutes les capsules de Table sont disponibles
  Mutex, S_Stock : semaphore;

  procedure Prendre(A : out Positive) is
    Succes : boolean; K : Table.Index;
  begin
    P(S_Stock);
    P(Mutex);
    Table.Rechercher(K, Disponible, Succes) ;
    --recherche une place K telle que Table(K) = Disponible
    Table.Changer(K, Alloue);
    V(Mutex);
    A := K;
  end Prendre;

  procedure Restituer(B : in Positive) is
  begin
    P(Mutex);
    Table.Changer(B, Disponible);
    V(Mutex);
    V(S_Stock);
  end Restituer;

begin
  -- initialisation des sémaphores
  E0(S_Stock, 40);E0(Mutex, 1);
end Memoire;
```

Question 2 (2 points)

On considère ci-dessous une programmation des procédures Deposier et Retirer_5 du paquetage Tampon et on vous demande d'utiliser et d'initialiser les sémaphores Mutex_P, S_Plein et S_Vide pour prendre en compte la concurrence d'accès à ce paquetage.

```
with SRI_B; use SRI_B;
paquetage body Tampon is
  Tamp : array (0.. 19) of Positive;
  Tete, Queue : Integer := 0;
  Mutex_P, S_Plein,S_Vide : semaphore;
```

```

procedure Deposer(A : in Positive) is
begin
  P(S_Vide);
  P(Mutex_P);
  Tamp(Queue) := A;
  Queue := (Queue + 1) mod 20;
  V(Mutex_P);
  V(S_Plein);
end Deposer;

```

```

procedure Retirer_5 ( B : out array(1..5) of Positive) is
begin
  for I in 1..5 loop
    P(S_Plein);
    B(I) := Tamp(Tete);
    Tete := (Tete + 1) mod 20;
    V(S_Vide);
  end loop;
end Retirer;

```

```

begin
  -- initialisation des sémaphores
  E0(Mutex_P, 1); E0(S_Plein, 0);E0(S_Vide, 20); ;
end Tampon;

```

```

-- VARIANTE bloquant plus longtemps le tampon
procedure Retirer_5 ( B : out array(1..5) of Positive) is
begin
  for I in 1..5 loop P(S_Plein); end loop;
  B := Tamp(Tete.. (Tete + 4) mod 20);
  Tete := (Tete + 5) mod 20;
  for I in 1..5 loop V(S_Vide); end loop;
end Retirer;

```

Question 3 (1 point)

On considère ci-dessous une programmation des procédures Attendre et Envoyer du paquetage Signalisation et on vous demande d'utiliser et d'initialiser S_Signal le tableau de sémaphores pour prendre en compte la concurrence d'accès à ce paquetage.

```

with SRI_B; use SRI_B;
paquetage Signalisation is
  S_Signal : array(1..40) of semaphore;

  procedure Attendre (C : in Positive) is
  begin
    -- attendre en utilisant le paramètre C
    P(S_Signal(C));
  end Attendre;

  procedure Envoyer(D : in Positive) is
  begin
    -- signaler en utilisant le paramètre D
    V(S_Signal(D));
  end Envoyer;
begin
  -- initialisation des sémaphores
  for I in 1..40 loop E0(S_Signal(I),0); end loop;
end Signalisation

```

Question 4 (3 points)

Pour étudier l'optimisation des transferts disque obtenue en réordonnant les requêtes pour réduire les déplacements du bras, on trace une suite de 20 pistes qui sont fournies au disquaire et on compare trois résultats obtenus en faisant porter l'optimisation sur des paquets successifs de 5 requêtes, :

- 1) à l'ancienneté (fifo), c'est à dire en servant les requêtes selon leur arrivée,
- 2) le plus proche voisin de la position courante du bras (ppv),
- 3) l'ascenseur aller et retour (aar)

La position initiale du bras est à la piste 53, et pour l'ascenseur, le premier déplacement se fait selon les pistes croissantes.

La suite de pistes est la suivante, par paquets de 5 valeurs :

suite	98	83	37	22	14	24	65	67	36	02	34	09	12	17	18	04	11	16	43	91
-------	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

Calculer le trajet total fait par la tête de lecture-écriture pour chacune des optimisations, en complétant les calculs suivants pour les 4 paquets de 5 valeurs de la suite:

fifo	98	83	37	22	14	24	65	67	36	02	34	09	12	17	18	04	11	16	43	91
trajet	45	15	46	15	8	10	41	2	31	34	32	25	3	5	1	14	7	5	27	48

ppv	37	22	14	83	98	67	65	36	24	02	09	12	17	18	34	43	16	11	4	91
trajet	16	15	8	69	15	31	2	29	12	22	7	3	5	1	16	9	27	5	7	87

aar	83	98	37	22	14	02	24	36	65	67	34	18	17	12	09	04	11	16	43	91
trajet	30	15	61	15	8	12	22	12	29	2	33	16	1	5	3	5	7	5	27	48

On constate que sur le premier paquet de 5 valeurs, le trajet total s'élève à 129 pour fifo, à 123 pour ppv et 129 pour aar. L'optimisation semble peu efficace. Est ce pareil pour toute la trace?

fifo =414 ppv =386 aar =356 L'optimisation joue et aar est le meilleur

Question 5 (2 points)

Pour améliorer l'efficacité de l'optimisation, on envisage de faire des paquets de 10 valeurs. Avant de modifier les programmes et d'écrire une procédure Tampon.Retirer_10, on veut connaître l'effet sur la suite de la question précédente.

La suite devient une suite de 2 paquets de 10 valeurs

suite	98	83	37	22	14	24	65	67	36	02	34	09	12	17	18	04	11	16	43	91
-------	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

Refaire les calculs pour les trois politiques fifo,ppv et aar par paquets de 10 sur la suite donnée. Quelles valeurs obtient-on pour le trajet total fait par la tête de lecture-écriture et quelles conclusions peut-on en tirer?

ppv	65	67	83	98	37	36	24	22	14	02	04	09	11	12	16	17	18	34	43	91
trajet	12	2	16	15	61	1	12	2	8	12	2	5	2	1	4	1	1	16	9	48

aar	65	67	83	98	37	36	24	22	14	02	04	09	11	12	16	17	18	34	43	91
trajet	12	2	16	15	61	1	12	2	8	12	2	5	2	1	4	1	1	16	9	48

fifo =414 (inchangé) ppv =230 aar =230

L'optimisation joue encore plus et ppv et aar sont équivalents

Si on avait commencé aar selon les pistes décroissantes on aurait eu aar =241

aar	37	36	24	22	14	02	65	67	83	98	91	43	34	18	17	16	12	11	09	04
trajet	16	1	12	2	8	12	63	2	16	15	7	48	9	16	1	1	4	1	2	5