Université A. Mira Bejaia
Faculté des Sciences Exactes
Département d'Informatique
Année d'étude : 3eme Année Licence Académique

Février 2021

CORRECTION DES EXERCICES DES ANCIENS EXAMENS

EXERCICE 1

Partie I

Q1) Répondre par Vrai ou Faux

a) La primitive « ***fork L*** » crée un processus fils qui va exécuter la tâche ***L***. **Faux**

Justificatif : elle crée un processus fils qui va commencer son exécution à l'étiquette L.

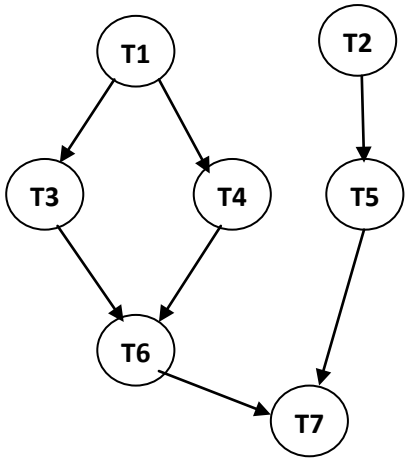
b) Les sémaphores binaires sont ceux qui ne sont utilisés que par deux processus. **Faux**

Q2) Laquelle des techniques d'allocation mémoire ci-dessous souffre de la fragmentation interne ?

b) Partitions multiples fixes

Partie II

1) Le graphe de précédence :



Solution Algorithme Fork, Join et Quit

```
N1 = N2 = 2 ;
        Fork L1 ;
        T1 ;
        fork L2 ;
        T3 ;
        Goto E1 ;
    L1 : T2 ;
        T5 ;
        Goto E2 ;
    L2 : T4 ;
    E1 : join N1 ;
        T6 ;
    E2 : join N2 ;
        T7 ;
```

EXERCICE 2

$Taille_{MC}$ = 32 Mo
$Taille_{Prgm}$ = 4 Ko

Partie 1 (Pagination)

- On a le nombre de bits pour représenter les numéros de cadre (Nb_{Bits_cadre})= 18 bits

1. Nombre de cadres de page :
$Nb_{Cadres} = 2^{NbBits\_cadre} = 2^{18} =$ 262144 cadres

2. Taille d'un cadre de page :
$Taille_{Cadre} = Taille_{MC} / Nb_{Cadres}$
$= 32 \text{ Mo} / 2^{18}$

$= 2^5 * 2^{20}/2^{18}$

$= 2^{25-18} = 2^7$

$= 128$ Ø

3. Nombre de pages du programme :

$Nb_{Page} = Taille_{Prgm} / Taille_{Page}$

$= 4$ Ko$/ 128$ Ø

$= 2^2 * 2^{10} / 2^7$

$= 2^{12-7}$

$= 32$ pages

Partie 2 (Segmentation paginée)

On a $Taille_{cadre} = 512$ Ø

1. Taille des segments :
   - $Taille_{S0} = 3 * 512 = 1536$ Ø
   - $Taille_{S1} = 2 * 512 = 1024$ Ø
   - $Taille_{S2} = 3 * 512 = 1536$ Ø
2. Taille des tables de pages des segments : on a :

- la taille d'une table de page ($Taille_{table\_page}$)= nombre d'entrée de la table * $Nb_{Bits\_cadre.}$

  La taille de la table de pages du segment S0 = 3*18,

  La taille de la table de pages du segment S1 = 2*18,

  La taille de la table de pages du segment S2 = 3*18.

- La taille des tables de pages des segments =la somme des $taille_{table\_page}$

  = 3*18+ 2*18+3*18

  = 8*18

  = 18 Ø (ou 144 bits)

Exercice 3 :

1) Cette solution assure-t-elle l'exclusion mutuelle ? Justifier votre réponse ?

Réponse : Oui, elle assure l'exclusion mutuelle, puisque au plus un processus accède à sa SC.

Explication :

Cas où un seul processus entre en SC:

- Si par exemple **P_A** termine l'exécution du code d'entrée (P(S), P(S)) **avant P_B**, il (**P_A**) **accède à la <SC>** /* *il ne se bloque pas, puisque le sémaphore S est initialement à 2 passe à 1 puis à 0**/ et **P_B se bloque** au niveau du premier P(S) /* *S passe à = -1**/
- En quittant la SC, **P_A,** exécute le code de sortie (V(S), V(S)) /**A la fin, la valeur du sémaphore est S = 0**/ et libère le processus **P_B** pour qu'il exécute sa **<SC>.**
- **Si P_A** exécute P(S) une autre fois avant la sortie de **P_B,** il et se bloque. /* *S = -1**/

Cas où aucun processus n'entre en SC:

- Si **P_A** termine l'exécution seulement du premier P(S) de son code d'entrée en SC /* S passe de 2 à 1*/ mais, avant d'exécuter le deuxième P(S), P_B exécute aussi le premier P(S) de son code d'entrée en <SC> /* *S passe de 1 à 0**/, puis, en poursuivant l'exécution de leur code d'entrée en <SC>; aucun processus n'accède à la <SC>. /* *puisque le sémaphore S passe de 0 à-1 puis de -1 à -2**/

2) Cette solution assure-t-elle la progression (Pas d'interblocage) ? Justifier votre réponse.

Non, on peut avoir **un cas d'interblocage**

- Si **P_A** termine l'exécution du premier P(S) de son code d'entrée en SC, **P_A** *ne se bloque pas* /* S passe de 2 à 1*/ et continue mais, avant d'exécuter le deuxième P(S),
- si P_B exécute aussi le premier P(S) de son code d'entrée en SC, **P_B** ne se bloque pas aussi /* *S passe de 1 à 0*/ et continue,
- puis, en poursuivant l'exécution de leur code d'entrée en SC; aucun processus n'accède à la <SC>.

Un des processus exécute P(S) et se bloque. /**S passe de 0 à -1*/*

Un autre processus exécute P(S) et se bloque. /* *S passe de -1 à -2*/*

- **Donc, un cas d'interblocage**

3) Modifier cette solution pour remédier au problème soulevé (**sans modifier l'initialisation du sémaphore**).

On enlève un **P(S) et V(S)** soit pour le **processus A ou** pour le **processus B**.

Exemple : On **enlève P(S) et V(S)** pour le **processus A**.

Processus *A*

```
{
        While (true){
                P(S);
                <SC>;
                V(S);
        }
}
```

***S* = *2*;**

Processus *B*

```
{
        While (true){
                P(S); P(S);
                <SC>;
                V(S); V(S);
        }
}
```