

ET N°3 Corrigé

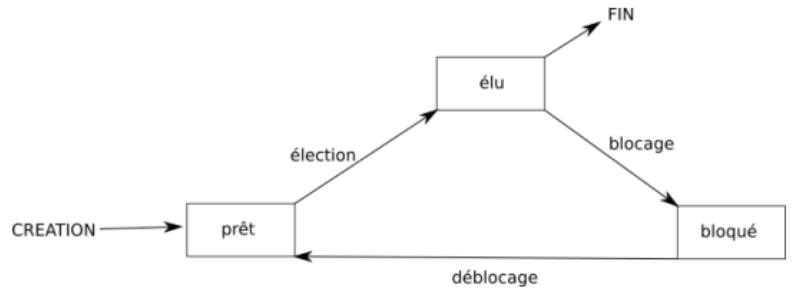
Processus / SGBD / Files

Exercice 1 : Processus

1/a/ Un processus dans l'état « élu » est un processus en train de s'exécuter.

1/b/

Voir le cours :

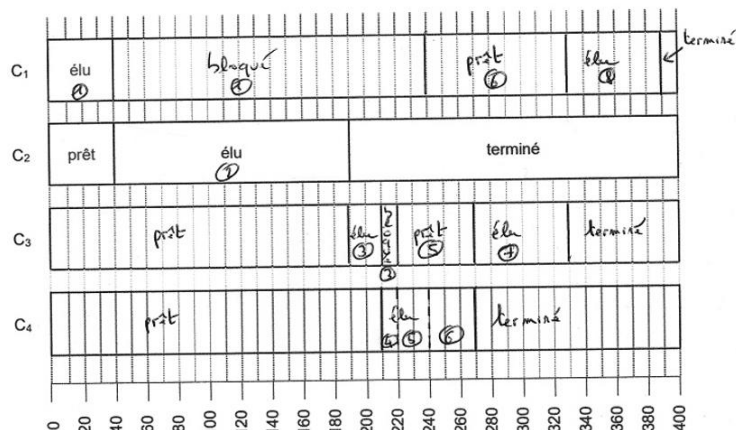


2/a/ C'est le mode FIFO : First In First Out (premier entré, premier sorti).

2/b/

A la création des processus, ils sont tous en état « prêt » et enfilés, on a donc cet ordre : C1 – C2 – C3 – C4

Rappel : un processus doit impérativement être dans un état « prêt » avant d'être élu.



En détail :

0 ms : C1 s'exécute pendant 20 ms (il est défilé) puis est bloqué (①). File : C2 – C3 – C4.

40 ms : C2 passe à l'état élu et s'exécute entièrement (②). File : C3 – C4.

190 ms : C3 passe à l'état élu (il est défilé) pendant 20ms puis est bloqué 10 ms (③). File : C4.

210 ms : C4 passe à l'état élu (il est défilé) pendant 10ms (④). File : vide

220 ms : C3 passe à l'état prêt (il est enfilé). C1 lui, est toujours bloqué. C4 poursuit son exécution pendant 20 ms. (⑤). File : C3

240 ms : C1 passe à l'état prêt (il est enfilé). C4 termine son exécution. File : C3 – C1.

270 ms : C3 passe à l'état élu (il est défilé) et s'exécute entièrement (⑥). File : C1

330 ms : C1 passe à l'état élu (il est défilé) et s'exécute entièrement (⑦). File : vide.

Tous les processus sont terminés.

Remarque : aucune justification n'est demandée dans le sujet.

3/a/ Deux méthodes :

Sous forme de phrase

P1 verrouille le fichier_1 et P2 verrouille le fichier_2. P1 attend le fichier_2 avant de pouvoir effectuer les calculs (et donc libérer le fichier_1). P2 attend le fichier_1 avant de pouvoir effectuer les calculs (et donc libérer le fichier_2). Nous avons donc une situation d'interblocage

Sous forme de graphe orienté



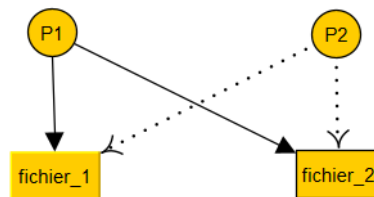
Présence d'un cycle : P1 – fichier_1 – P2 – fichier_2 – P1. C'est une situation d'interblocage.

3/b/

Sous forme de phrase

P2 doit verrouiller le fichier_1 puis le fichier_2. Or, le fichier_1 sera déjà verrouillé par P1, P2 sera donc bloqué et P1 pourra s'exécuter.

Sous forme de graphe orienté



Pas de cycle donc pas d'interblocage.

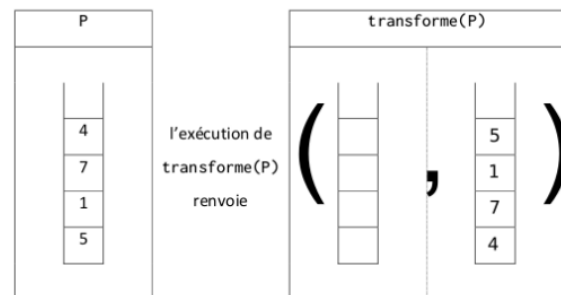
Remarque : P2 libérera d'abord fichier_2 et fichier_1. Il faudra aussi le modifier dans le programme.

Exercice 2 : Piles

1/

	Etape 0 Pile d'origine P	Etape 1 empiler(P,8)	Etape 2 depiler(P)	Etape 3 est_vide(P)
	<div>4 7 1 5</div>	<div>8 4 7 1 5</div>	<div>4 7 1 5</div>	<div>4 7 1 5</div>
Retour de la fonction		None	8	False

2/



3/

```
def maximum(P):
    m=depiler(P)
    while not est_vide(P):
        v = depiler(P)
        if v > m:
            m = v
    return m
```

4/

a/ Il suffit de mettre place une boucle qui s'arrêtera quand la pile P sera vide. À chaque tour de boucle, on dépile P, on empile les valeurs précédemment dépilées dans une pile auxiliaire Q et on incrémente un compteur de 1. Une fois la boucle terminée, on crée une nouvelle boucle où on dépile Q et on empile P avec les valeurs dépilées pour retrouver l'état initial de la pile. Il suffit ensuite de renvoyer la valeur du compteur.

b/

```
def taille(P):  
    cmp = 0  
    Q = creer_pile()  
    while not est_vide(P):  
        v = depiler(P)  
        empiler(Q,v)  
        cmp = cmp + 1  
    while not est_vide(Q):  
        v = depiler(Q)  
        empiler(P,v)  
    return cmp
```

Exercice 3 : SGBD

1

Pour effectuer des requêtes sur une base de données relationnelle, on utilise le langage SQL

2a

ATOME (Z : INT, nom : TEXT, Sym : TEXT, L : INT, C : INT, masse_atom : FLOAT)
VALENCE (Col : INT, Couche : TEXT)

2b

l'attribut Z peut jouer le rôle de clé primaire car il existe un Z unique pour chaque élément chimique.

l'attribut C va jouer le rôle de clé étrangère car cet attribut va permettre d'établir une "liaison" avec l'attribut Col de la table VALENCE

2c

ATOME (Z : INT, nom : TEXT, Sym : TEXT, L : INT, #C : INT, masse_atom : FLOAT)
VALENCE (Col : INT, Couche : TEXT)

Remarque : souligner un attribut est le code pour le désigner « clé primaire » et le précéder d'un hashtag est pour le rôle de clé étrangère). On peut tout à fait le signaler à l'aide d'une phrase.

3a

On obtient la liste de nom d'atomes suivante :
aluminium, argon, chlore, magnesium, sodium, phosphore, silicium, soufre

3b

On obtient la liste des colonnes :
1, 2 ,3 ,4 ,5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18

4a

```
SELECT nom, masse_atom  
FROM ATOMES
```

4b

```
SELECT Sym  
FROM ATOMES  
INNER JOIN VALENCE ON ATOMES.C = VALENCE.Col  
WHERE Couche = 's'
```

5

```
UPDATE ATOMES  
SET mass_atom = 39.948  
WHERE nom = 'argon'
```