Sujet de bac 21. NSIJ2P01

Correction

Exercice 1

Partie A : Manipulation de listes.

1. Affichage obtenu :

```
8
[8, 7, 18, 16, 12, 9, 17, 3]
```

2. `print(notes[2:5])`

Partie B : Tri par insertion

1.

```
6       while i>=0 and liste[i] > liste[indice_courant]:
7           liste[i-1] = liste[i]
```

2. Après 1° passage :

notes = [7, 8, 18, 14, 12, 9, 17, 3]

3. Après 3° passage :

notes = [7, 8, 14, 18, 12, 9, 17, 3]

Partie C : Tri fusion

1. Cet algorithme est récursif car il coupe la liste en 2 sous listes et cherche à trier chaque sous liste puis recombine les 2.

2. Pour trier deux tas il suffit de poser sur un troisième tas la carte la plus petite des deux tas et recommencer.

3.

```
8     tri_fusion (liste, i_debut, i_partage)
9     tri_fusion (liste, i_partage +1 , i_fin)
10    fusionner (liste, i_debut, i_partage , i_fin)
```

4. Cette ligne charge en mémoire la fonction floor de la bibliothèque math.

Partie D : Comparaison des tris.

1- Le tri fusion a été utilisé, cette figure illustre la fusion des valeurs triées.

2. $\mathcal{O}(n^2)$ → Tri par insertion

$\mathcal{O}(n \log_2 n)$ → Tri fusion

3. Le tri par insertion est composé de deux itérations imbriquées → complexité quadratique

Le tri fusion découpe en deux chaque liste (complexité $\mathcal{O}(\log_2 n)$) et fusionne chaque liste triée (complexité $\mathcal{O}(n)$).

Exercice 2

1. Clés primaires :
   - relation clients → id Client
   - relation Articles → id Articles.

2. Domaine de Email : chaîne de caractères de longueur max 50 caractères

3. FOREIGN KEY (id Client) REFERENCES Clients(idClient)

Partie B : Site WEB.

1. La méthode GET transmet les données en les ajoutant à l'adresse du serveur. La quantité de données à transmettre est ainsi limitée à 2000 caractères.
La méthode POST ajoute directement les données dans la trame d'envoi ce qui est plus sécurisé et permet d'ajouter une grande quantité de données.

2. HTTPS assure le cryptage des données par un cryptage symétrique. L'échange de la clé de cryptage est effectué par un cryptage asymétrique.

3. Il est préférable de vérifier le format avant d'envoyer la requête pour éviter que le SGBD ne rejette la requête (contrainte de domaine).

Partie C : SQL

1. SELECT IdArticle, Libelle FROM Articles
WHERE PrixEnCentimes < 1500;

2. Cette requête renvoie l'identifiant du client, l'email, l'identifiant de la commande et l'adresse de livraison des utilisateurs dont le paiement n'est pas validé.

3. SELECT a.Libelle FROM Articles As a.
Join ArticlesCommande AS ac
ON a.idArticle = ac.idArticle
WHERE idCmd = 1345;

4. INSERT INTO Articles (Libelle, Description, PrixEnCentimes) SET ('imperméable', 'Cet imperméable se replie en forme de pochette', 999);

Partie D : Adaptation du modèle

1. Il faut modifier :
   - la relation Articles en ajoutant un attribut nombre (domaine INT) qui regait le nombre d'articles en stock.
   - la relation Client en ajoutant un attribut Adresse Livraison (VARCHAR) qui recevra l'adresse de livraison par defaut de l'utilisateur.

2. Modifier Stock ← Stock - 1
   par Stock ← Stock - quantité.

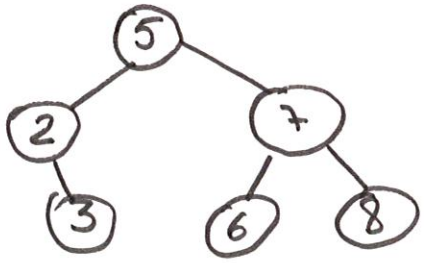
Exercice 3

Partie A.

1. . Noeud racine : 5
   . Fils gauche : 2
   . Fils droit : 7.

2. Noeuds de la branche : 5 - 2 - 3

3.



Partie B

1. __init__ est le constructeur de la classe. Cette méthode est appelée lors de l'instanciation de la classe (création d'un objet).

2. L'élément n'est pas inséré s'il existe déjà dans l'arbre.

3. arbre = ABR (5)
   arbre . insereElement (2)
   arbre . insereElement (3)
   arbre . insereElement (7)
   arbre . insereElement (8)

Partie C

1. Parcours infixe

2. Complexité du tri fusion : $O(n \cdot \log_2 n)$

   Complexité du tri par insertion / sélection : $O(n^2)$

Le tri fusion est plus performant que les deux autres.

Exercice 4.

Partie A.

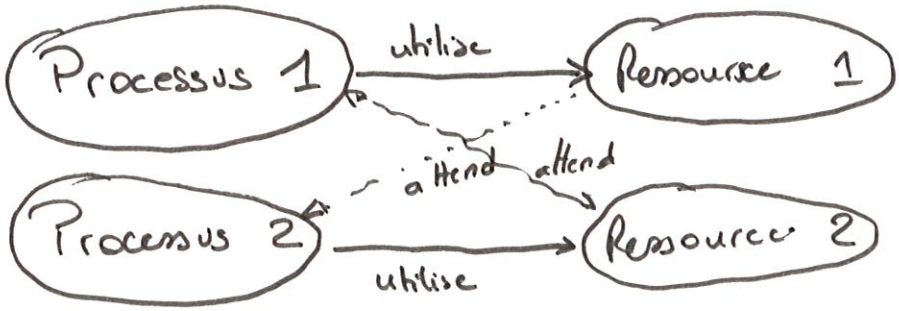
1. Il y a découpage pour fluidifier les échanges. En effet ceci offre la possibilité de router chaque paquet par des routes différentes et l'envoi en une fois d'une grosse donnée oblige à renvoyer l'ensemble en cas d'erreur de transmission.

2.a Les sommets représentent les routeurs et les arêts les liens entre routeurs.

b. Rip utilise le nombre de sauts.

Partie B

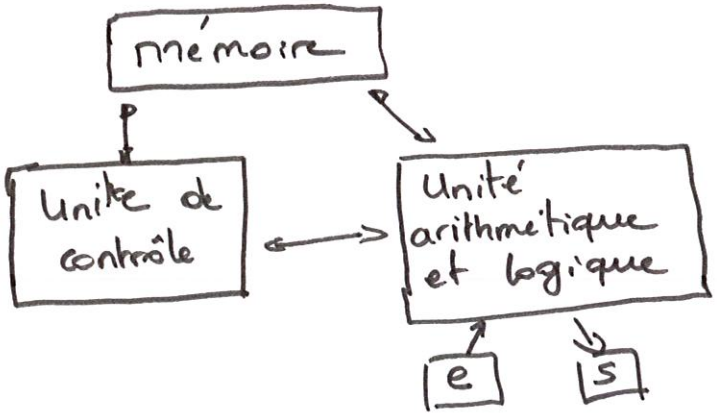
1. Un interblocage intervient quand deux processus attendent mutuellement des ressources utilisées par l'autre.



2. Ceci peut être évité en connaissant à tous moment l'état de toutes les ressources
→ Difficile (impossible) à assurer totalement.

Partie II

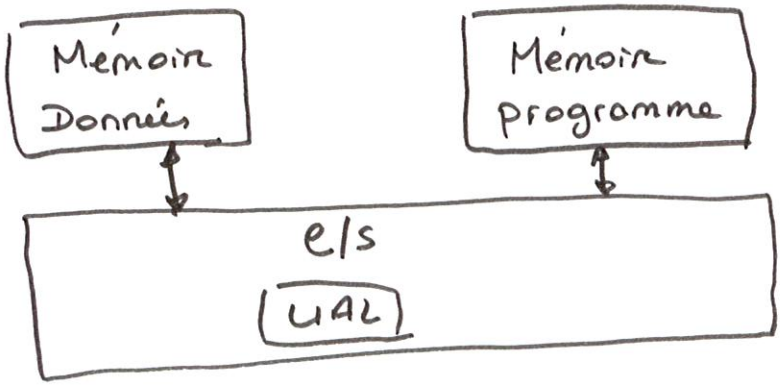
1.



2. Ces deux registres sont contenus dans l'unité de contrôle.

- Le compteur programme contient l'adresse mémoire de l'instruction à appeler
- Le registre d'instruction va ensuite recevoir cette instruction pour la décoder ensuite.

3.



4. Mémoire vive : mémoire qui perd son contenu lorsqu'elle n'est plus alimenté -> mémoire volatile.

Mémoire morte : mémoire qui conserve les informations même en cas de perte d'alimentation.

Dans un microprocontrôleur, il faut conserver le programme (travail à faire) même en cas de perte d'énergie

Partie D.

1. Avoir plusieurs processeurs permet d'effectuer plusieurs tâches en même temps.

2. Les bus ont des vitesse différentes car les périfériques ont des vitesses différentes (mémoire plus rapide qu'un module wifi). Dans le cas d'un unique bus la vitesse serait fixée à la vitesse la plus lente.

3. . Encombrement
   . Consommation
   . Echauffement

4. . Capacité de stockage de données
   . Maintenance

Exercice 5

Partie A.

1. Clés primaires :

relation Films : idFilm

relation Abonnés : idCpt

2. Domaine idFilm : entier (INT)

Description : caractères (VARCHAR)

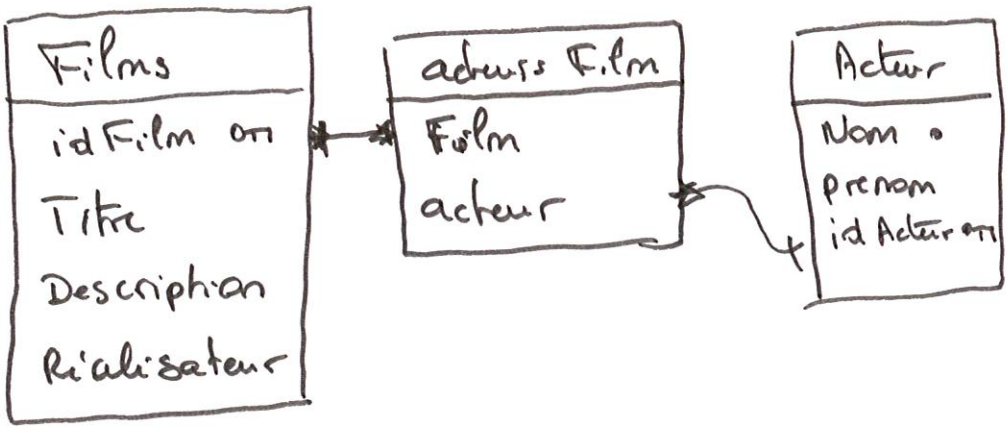
3. Compte Abonnés :

Clé primaire : idCpt

Clé étrangère : idAbonne

4. Les acteurs peuvent être repertoriés dans une relation supplémentaire. Chaque acteur comporterait une id. (clé primaire).

Une relation supplémentaire (acteursFilm) enregistrerait pour chaque Film, les acteurs principaux.



5. Il faut ajouter un attribut d'âge dans la relations Abonnes et Films.

Partie B

1. SELECT idCpt, Pseudo FROM ComptesAbonnes WHERE idAbonnes = 237;

2. Donne la moyenne des nombres d'étoiles issus des votes pour le film d'id 1542.

3. Cette requête renvoie les films préférés (id, titre et nombre d'étoiles) du compte 508 classés par ordre de préférence.

4. UPDATE ComptesAbonnes SET Pseudo = 'Champion' WHERE idCpte = 508;

Partie C

1. Cette fonction calcule la moyenne des écarts de notation entre deux utilisateurs pour une liste de films donnée.

2.
```
def conseilFilms ( idCpt):
    Conseil = []
    meilleursFilms = podiumCompte (idCpt).
    spectProches= spectateurs (meilleursFilms).
    for idSpectProche in spectProches:
        if distance (idCpt, idSpectProche,
            meilleursFilms) <10:
            conseil.append ((podiumCompte(
                idSpectProche) [:4])
    return conseil
```