
Partie I (12 points)

Structures de données dynamiques : (4pts)

Soit T un tableau de taille N.

Expliquer comment implémenter (en Pascal ou en C) deux piles P1 et P2 dans le même tableau T.

On doit s'assurer dans notre implémentation qu'aucune des deux piles ne déborde sur l'autre et que chaque pile peut avoir un nombre d'éléments inférieur ou égal à N.

L'implémentation consiste à :

- Définir les structures de données
- Réaliser la procédure **Initialiser** d'initialisation de ces structures de données
- Réaliser la fonction **PilePleine1** donne Vrai s'il ne reste aucun espace de libre sur le tableau pour la pile P1.
- Réaliser la fonction **PileVide1** qui donne Vrai si la pile P1 est vide.
- Réaliser la procédure **Empiler1** qui empile dans cette pile P1.
- Réaliser la fonction ou procédure **Dépiler1** qui dépile de la Pile P1.
- Réaliser la fonction **PilePleine2** donne Vrai s'il ne reste aucun espace de libre sur le tableau pour la pile P2.
- Réaliser la fonction **PileVide2** qui donne Vrai si la pile P2 est vide.
- Réaliser la procédure **Empiler2** qui empile dans cette pile P2.
- Réaliser la fonction ou procédure **Dépiler2** qui dépile de la Pile P2.

Structures de fichiers : (4pts)

- Ecrire un corps d'un algorithme (partie action) qui permet d'afficher en ordre croissant le contenu d'un fichier de données F2 (enregistrements de format fixe, non ordonné) en utilisant un index primaire dense (Tprim). Nous souhaitons limiter l'affichage uniquement aux enregistrements existants dans un deuxième fichier F1 organisé sous forme d'arbre de recherche m-aire. Nous définissons les structures comme suit:

```

TYPE
Typearticle = STRUCTURE
    Clé : Typeclé
    Info : Typeqq
FIN
    
```

```

TYPE
T = STRUCTURE
    Clé : Typeclé
    Adr : Typeadresse
FIN
    
```

```

TYPE
Typedresse = STRUCTURE
    Numbloc : ENTIER
    Depl : ENTIER
FIN
    
```

```

Type Tbloc1 = structure
Val : tableau[N-1] de Typeclé;
Fils : tableau[N] d'entier;
degré : entier;
fin;
    
```

```

TYPE Tbloc2 = STRUCTURE
Nb : ENTIER
Tab : TABLEAU[1..B] DE Typearticle
Fin
    
```

VAR

Tprim : TABLEAU[1..M.] DE T

FARBRE : FICHIER de Tbloc1 BUFFER buf1 entete(entier);

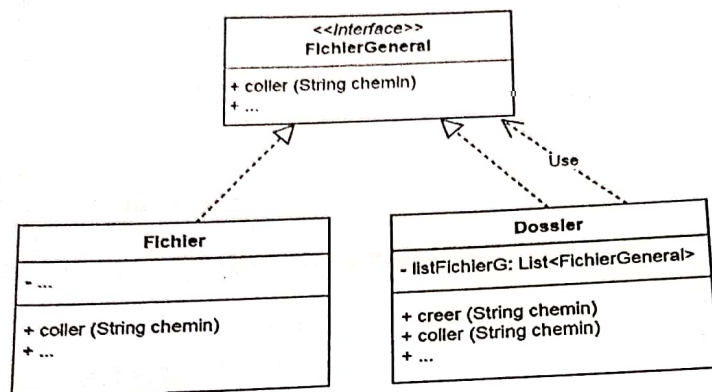
FDONNEE : FICHIER de Tbloc2 BUFFER buf2 entete(entier);

Nb :

- Le module suivant Recherche(c, F1 ,trouve, i, j) qui permet de réaliser une recherche dans un fichier organiser sous forme d'un arbre de recherche m-aire est déjà prédéfini.

Programmation Orientée Objet : (4pts)

« Composite Pattern » est un des modèles de conception orienté objet. Il sert à simplifier le traitement des structures en arbre en utilisant la récursivité et le polymorphisme. Le schéma ci-dessous illustre l'application du « Composite Pattern » dans le contexte d'une application de gestion de fichiers sous Windows.



Etant donné les suppositions suivantes :

- a- Un dossier peut contenir des fichiers et des dossiers qui sont tous stockés dans l'attribut « listFichierG »
- b- Pour coller un fichier, il suffit d'appeler la méthode « coller(chemin) » de la classe « Fichier » qui est considérée **prédéfinie**.
- c- Pour coller un dossier, il faut instancier un nouvel objet de la classe « Dossier », le créer en appelant la méthode « creer(chemin) » supposée **prédéfinie**, ensuite coller tout son contenu disponible dans « listeFichierG » en utilisant la récursivité et le polymorphisme.

Question. Donner le code Java de la classe « Dossier » en considérant les attributs et méthodes présentés dans le schéma ci-dessus. La méthode « creer » est supposée prédéfinie (Utiliser la méthode « creer » sans fournir son code).

Partie II (8 points). Architecture des ordinateurs 2

Exercice 1 (5 pts)

On désire réaliser une pile de type FIFO qui pourra être lue par trois processeurs différents P0, P1, P2. Les mots de la FIFO sont composés de deux champs: Data et N° du processeur (NP) qui reçoit cette data.

Lorsqu'un mot est écrit dans la FIFO, les deux champs sont écrits en même temps.

La lecture dans la FIFO est réalisée par un circuit logique (C). Lorsqu'un processeur demande une lecture, C vérifie d'abord (compare) le champ NP du mot qui se trouve en sortie de la FIFO et le numéro du processeur. La lecture ne peut se faire que si le numéro qui se trouve dans le champ NP du mot est le même que le numéro du processeur.

- Si les deux numéros sont les mêmes, alors le mot est lu et le champ information est envoyé vers le bus du processeur Pi qui a demandé la lecture.

- Si les deux numéros sont différents, alors la lecture ne se fait pas.

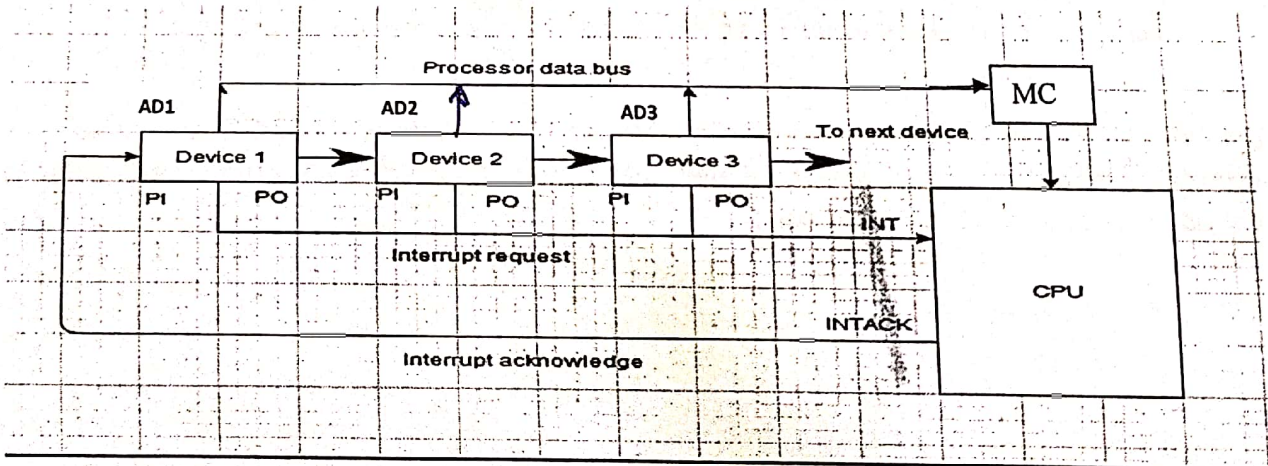
Attention: les deux processeurs peuvent demander des lectures en même temps.

Questions:

1. Représenter sur un schéma les entrées et les sorties du circuit C, ainsi que les connexions avec les différents processeurs, avec les bus de données et avec la FIFO (ne pas détailler l'intérieur de la FIFO ni le contenu du circuit C). (3 pts)
2. Donnez les équations logiques du circuit C. (2 pts) [sans passer par la table de vérité ni implémentation avec une ROM]

Exercice 2 : (3 pts)

Soit un système de gestion de priorité d'interruptions hiérarchiques basée sur la méthode de Daisy chain selon la figure suivante :



AD : Adresse de l'interruption.

PI : Priority In ; PO : Priority Out (entrée-sortie (INTAck venant du processeur) qui sera acheminé d'un contrôleur (Device) à un autre dans la chaîne Daisy ; sachant que dans cette chaîne, le contrôleur le plus proche du processeur est le plus prioritaire, et ainsi de suite dans la chaîne jusqu'au dernier contrôleur qui est le moins prioritaire.

Expliquez le fonctionnement de la chaîne « Daisy chain ».