

Chapitre 7:Les Files

1. Définition

On appelle file d'attente (ou tout simplement file) un ensemble formé d'un nombre variable, éventuellement nul de données, sur lequel les opérations suivantes peuvent être effectuées :

créer_file : permet de créer une file vide (création).

file_vide : permet de tester la vacuité d'une file, ou si la file est vide ou non (consultation).

enfiler : permet d'ajouter une donnée de type T à la file (modification).

defiler : permet d'obtenir une nouvelle file (modification).

premier : permet d'obtenir l'élément le plus ancien dans la file (consultation).

Opérations illégales : il y a des opérations définies sur la SD file d'attente exigent des pré-conditions :

defiler exige que la file soit non vide.

premier exige que la file soit non vide.

2. Propriétés

Dans ce paragraphe on va citer (énumérer) les propriétés qui caractérisent la sémantique des opérations applicables sur la SD file d'attente.

F1 : créer_file permet de créer une file vide.

F2 : si un élément entré (enfiler) dans la file résultante est non vide.

F3 : un élément qui entre (grâce à enfiler) dans la file d'attente devient immédiatement le premier : si la file vide sinon (file non vide) le premier reste inchangé.

F4 : une entrée et une sortie successive sur une file vide la laissent vide.

F5 : une entrée et une sortie successive sur une file non vide peuvent être effectuées dans n'importe quel ordre.

Illustration :

File non vide : 5 6 1

Cas 1 : 5 6 1

enfiler 8 → 5 6 1 8

defiler → 6 1 8

Cas 2 : 5 6 1

defiler → 6 1

enfiler 8 → 6 1 8

La structure de File obéit à la loi FIFO : First In, First Out.

3. Représentation physique

On distingue deux types de représentations physiques :

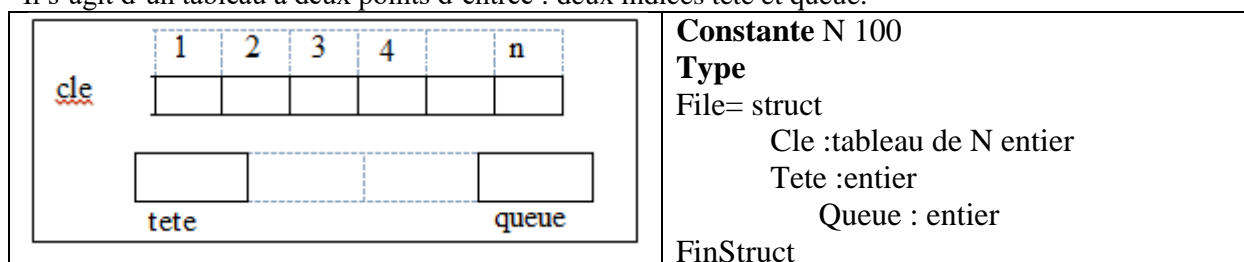
-représentation contiguë

-représentation chaînée

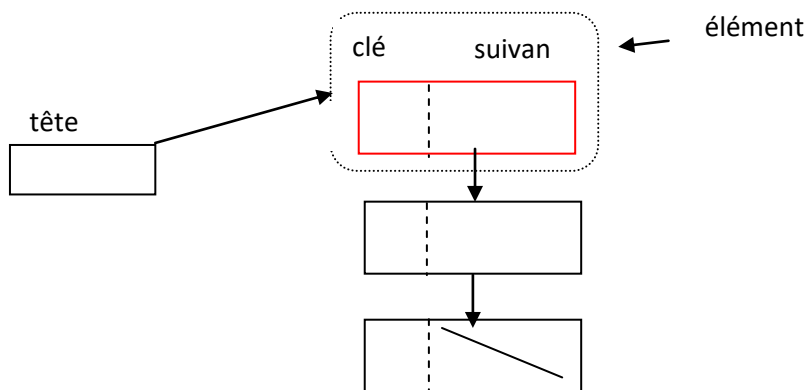
3.1 Représentation contiguë

TDA FILE concrétisé par une représentation contiguë

Il s'agit d'un tableau à deux points d'entrée : deux indices tête et queue.



3.2 Représentation chaînée



premier : coût une indirection en partant du pointeur tête

defiler : coût une indirection en partant du pointeur tête

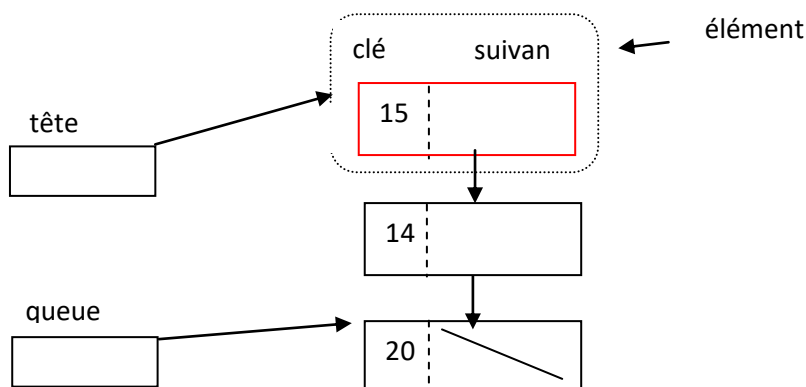
enfiler : coût il faut parcourir toute la file en partant du pointeur tête.

Ceci nécessite plusieurs indirections. Ainsi, **il ne faut pas retenir la solution proposée**

Remède : on a besoin d'une représentation physique à deux points d'entrées : tête et queue.

Le pointeur **tête** favorise l'implémentation efficace des opérations : premier et defiler.

Le pointeur **queue** favorise l'implémentation efficace de l'opération enfiler.



Remarque :

La SD file d'attente est structurée à deux points d'entrée. Par contre la SD pile est une structure à un seul point d'entrée.

4Matérialisation de la SD File comme type de donnée abstrait

/*représentation chaînée*/

Types

Cellule = Struct
 cle : entier
 Suiv : ^Cellule
 FinStruct

File = Struct
 tete : ^Cellule
 queue : ^Cellule
 FinStruct

Procédure creer_File (varF :File)**Début**F.tete \leftarrow NilF.queue \leftarrow Nil**Fin Proc****Fonction** File_vide (F :File): boolean**Debut**File_vide \leftarrow (F.queue=Nil)**fin Fn****Procédure** enfiler (x : Entier , Var F : File)**Var**

P : ^Cellule

Début

Allouer(P)

P^.cle \leftarrow xP^.Suiv \leftarrow Nil

Si File_vide(F) alors

F.tête \leftarrow PF.queue \leftarrow P

sinon

F.queue^.Suiv \leftarrow PF.queue \leftarrow P

FinSi

Fin Proc**Procédure** défiler (Var F : File)**Var**

Q : ^Cellule

Début

Assure (non File_vide(F))

Q \leftarrow F.TêteF.Tête \leftarrow F.Tête^.Suiv

Libérer(Q)

Si F.Tête = NIL **alors**F.queue \leftarrow NIL**FinSi****Fin Proc****Fonction** premier (F : File) : entier**Début**

Assure (non File_vide(F))

premier \leftarrow F.tête^.cle**Fin Fn**