COURS IV

LES FILES D'ATTENTE

Sommaire

Définition

Une file est une structure de données dynamique dans laquelle on insère des nouveaux éléments à la fin (queue) et on enlève des éléments au début (tête de file). L'application la plus classique est la file d'attente. La file sert beaucoup en simulation. Elle est aussi très utilisée aussi bien dans la vie courante que dans les systèmes informatiques. Par exemple, elle modélise la file d'attente des clients devant un guichet, les travaux en attente d'exécution dans un système de traitement par lots, ou encore les messages en attente dans un commutateur de réseau téléphonique. On retrouve également les files d'attente dans les programmes de traitement de transactions telle que les réservations de sièges d'avion ou de billets de théâtre.

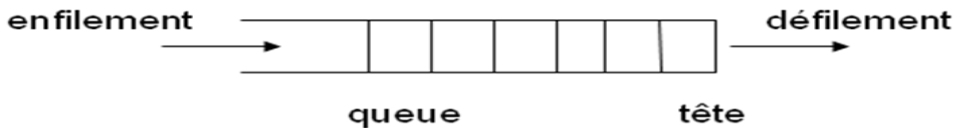
Une file d'attente peut être définie comme une collection d'éléments dans laquelle tout nouveau élément est inséré à la fin et tout élément ne peut être supprimé que du début.

Principe, domaine d'application

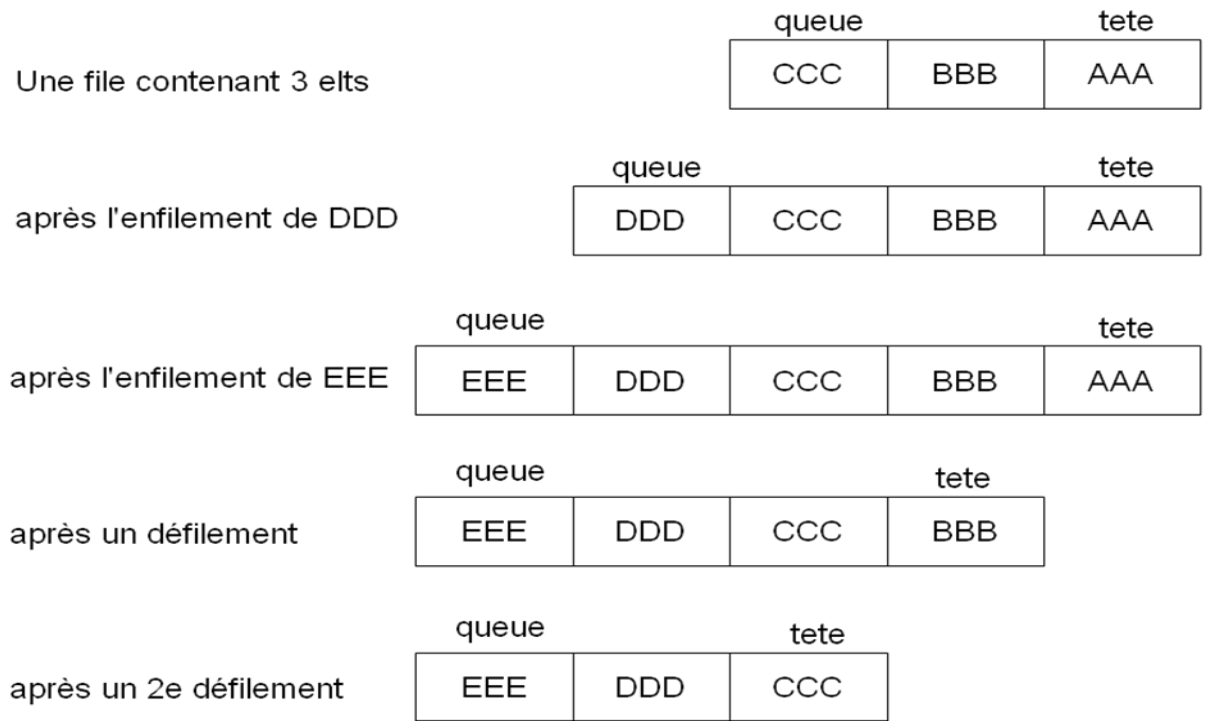
C'est le principe "**FIFO**", abréviation de "First In, First Out" qui veut dire " premier entré premier servi ".

La file d'attente est très utilisée dans les systèmes d'exploitation des ordinateurs et surtout dans les problèmes de simulation.

Nous verrons aussi que la file d'attente peut être utilisée pour le parcours des arbres et pour résoudre tant d'autres problèmes.



Exemple de File :



Modèle

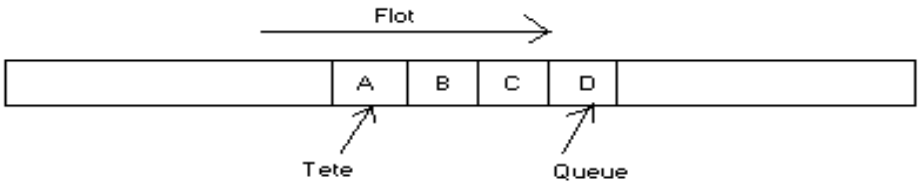
C'est l'ensemble des opérations définies comme suit :

CréerFile(F)	créer une file vide
Enfiler(F,Val)	ajouter Val en queue de file.
Défiler(F,Val)	retirer dans Val l'élément en tête de file.
Filevide(F)	tester si la file est vide.
Filepleine(F)	tester si la file est pleine.

Implémentation

- Représentation statique : au moyen de tableaux :
  ➔ le nombre maximum d'éléments est fixé une fois pour toute
  3 approches :
  - par flot → pas pratique, juste pour introduire la prochaine
  - par décalage → pas efficace
  - Tableau circulaire → standard (pratique et efficace

par flot :



Description Algorithmique

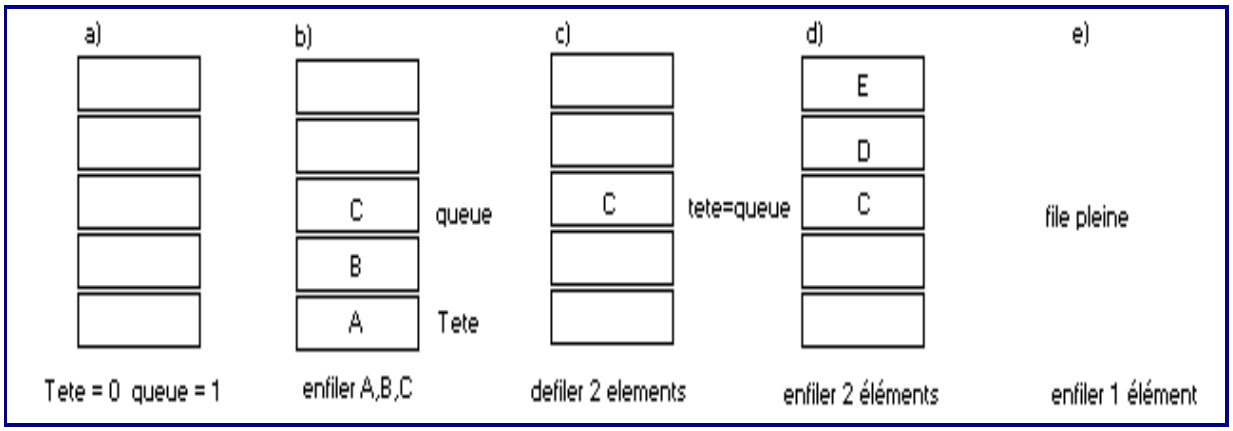
```
TYPE Filedattente = STRUCTURE
        Elements : TABLEAU(1..Max) DE Typeqq
        TêTe, Queue : ENTIER
FIN
VAR F : Filedattente
```

- Créerfile(F) : F.Queue :=0 ET F.TêTe := 1
- Filevide : Filevide := (F.Queue< F.TêTe)
- Filepleine(F) : Filepleine := (F.Queue = Max)
- Enfiler(F, X) : SI NON Filepleine(F)
  F.Queue := F.Queue + 1
  F.Elements(F.Queue) := X
  SINON
  "Overflow"
  FSI
- Defiler(F,X) : SI NON Filevide(F)
  X := F.Elements(F.TêTe)
  F.TêTe := F.TêTe + 1
  SINON
  "Underflow"
  FSI

Remarque 1 : A tout moment le nombre d'éléments est F.queue - F.tête + 1

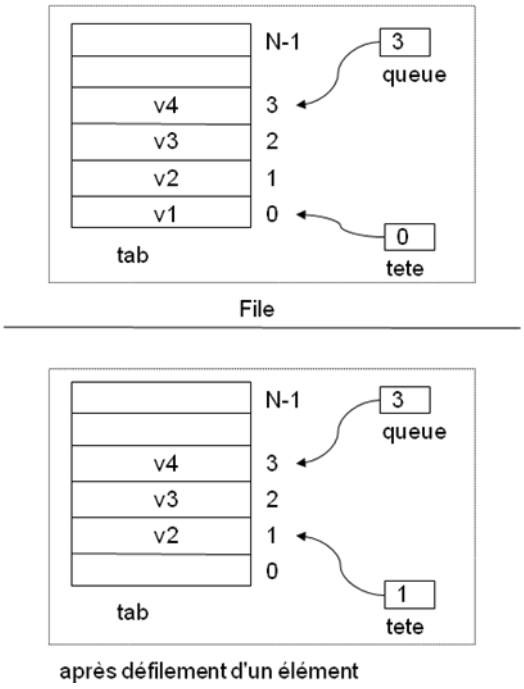
Remarque 2 : La file n'est pas vide si F.queue>=F.tête, donc la file est vide si non (F.Queue>=F.Tête), c'est à dire F.Queue<F.Tête.

Exemple1 :



⇨ C'est une solution inacceptable car on ne peut pas récupérer les emplacements X tels que X<F.Tête

Exemple2 :



par décalage :

A chaque défilement, on fait un décalage vers le bas. La tête n'est plus une caractéristique de la file d'attente puisqu'elle est toujours égale à 1.

Description Algorithmique

```
TYPE Filedattente = STRUCTURE
        Elements : TABLEAU(1..Max) de Typeqq
        Queue : ENTIER
FIN
VAR F : Filedattente
```

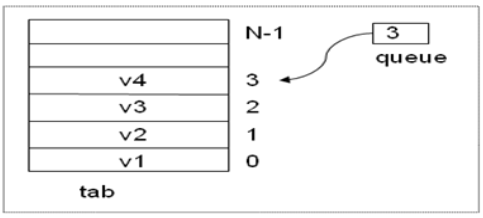
- Créerfile(F) : F.Queue := 0
- Filevide : Filevide := (F.Queue = 0)
- Filepeine : Filepleine := (F.Queue = Max)
- Defiler(F, X) :

```
SI NON Filevide(F) :
        X := F.Elements(1)
        POUR I := 1 à F.Queue - 1 :
                F.Elements(I) :=F.Elements(I + 1)
        FINPOUR
        F.Queue := F.Queue – 1
SINON
        " Filevide "
FSI
```

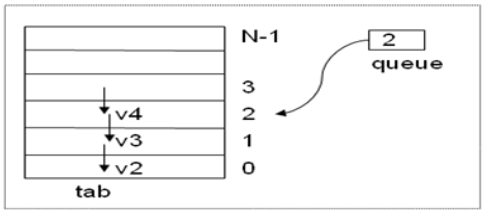
- Enfiler(F,X) :

```
SI NON Filepleine(F):
        F.Queue := F.Queue + 1
        F.Elements(F.Queue) := X
SINON
        "Filepleine"
FSI
```

Exemple :



File

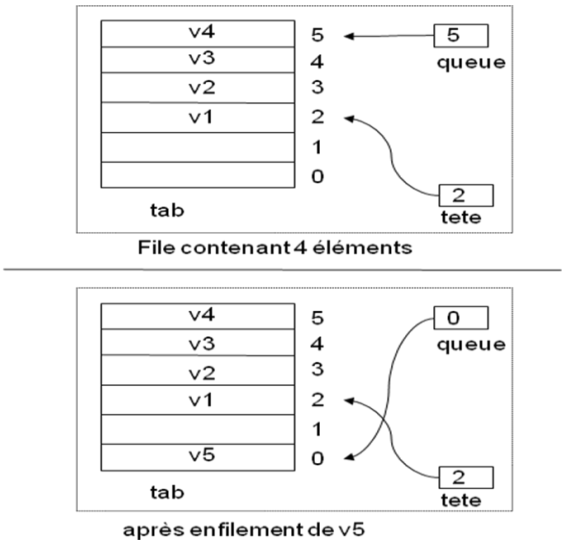


après défilement d'un élément

⇨ L'inconvénient de cette solution est que pour chaque défilement, on fait un décalage.

Tableau circulaire :

Revenons à la solution par flot et essayons d'utiliser le tableau de façon circulaire. Considérons l'exemple suivant :

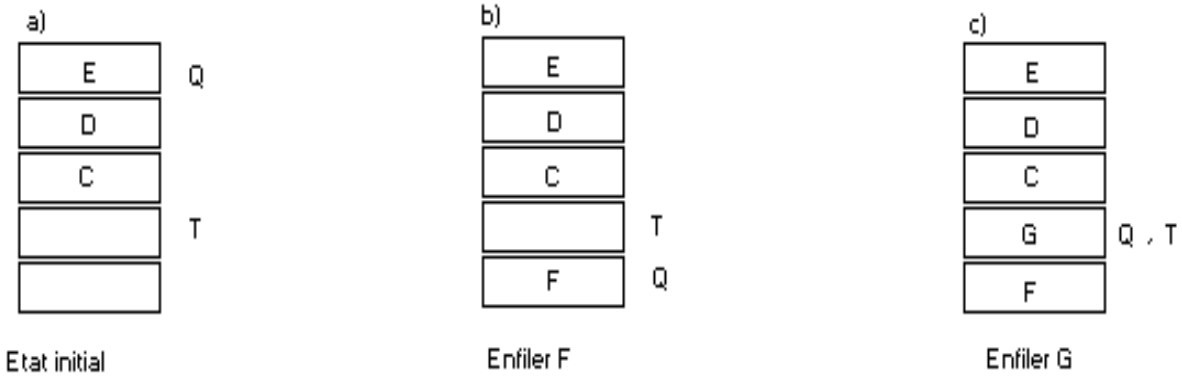


➢ Comment initialiser la file?

F.Queue < F.Tête impossible (contre exemple)

F.tête = F.Queue impossible (c'est le cas ou il reste un élément dans la file).

➔ Regardons ce qui se passe avec ces nouvelles considérations :



F.tête = F.queue constitue aussi le cas file pleine.

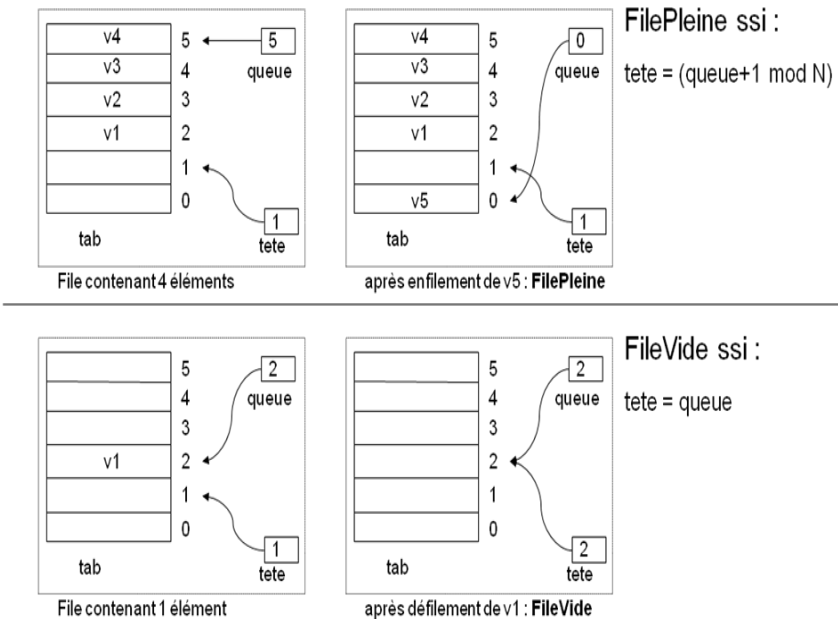
- même principe que l'approche par flots (même déclaration)
- les incrémentations se font modulo N => réutilisation des cases libérées.
- efficace : pas de boucle
- réutilise l'espace perdu par les défilements

Solution

F.Tête : pointe l'élément qui précède le premier.
F.Queue : pointe le dernier élément.
(F.Tête = F.Queue) constitue alors le cas file vide.
Initialisation (F.Tête = F.Queue := Max)

En d'autre terme :

par convention, l'élément d'indice tête sera sacrifié.
le 1er élément se trouve alors à l'indice (tête+1 mod N)



- Créerfile(F) : F.TêTe = F.Queue := Max
- Filevide(F) : Filevide := (F.TêTe = F.Queue)
- Filepleine(F) : Filepleine := (F.TêTe = (F.Queue Mod Max + 1))
- Enfiler(F, X) : SI NON Filepleine(F)

```
        (1)     SI F.Queue = Max
                    F.Queue := 1
                SINON
                    F.Queue := F.Queue + 1
                FSI
                F.éléments(F.Queue) := X
SINON
        " Overflow"
FSI
```

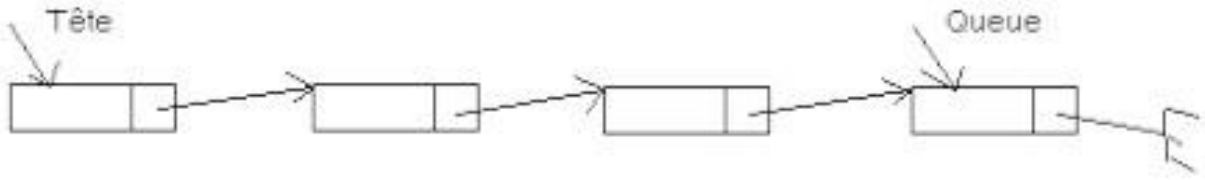
- Defiler(F, X) : SI NON Filevide(F)

```
(2)     SI F.TêTe = Max
            F.TêTe := 1
        SINON
            F.TêTe := F.TêTe + 1
        FSI
        X := F.éléments(F.TêTe)
SINON
    "Underflow"
FSI
```

Remarque:

L'alternative (1) est équivalente à : F.Queue := F.Queue Mod Max + 1

- Représentation dynamique : au moyen des listes linéaires chaînées



Description Algorithmique

```
TYPE S= STRUCTURE
        Info : Typeqq
        Suiv : POINTEUR(S)
FIN
TYPE Filedattente = STRUCTURE
        TêTe, Queue : POINTEUR(S)
FIN
VAR F : Filedattente
```

- Créerfile(F) : F.Tête := NIL
- Filevide(F) : Filevide := (F.TêTe = NIL)
- Enfiler(F, X) :
```
Allouer Q(S)
Aff_Val(Q, X)
Aff_Adr(Q, NIL)
SI NON Filevide(F)
        Aff_Adr(F.Queue, Q)
SINON
        F.TêTe := Q FSI
FSI
F.Queue := Q
```
- Defiler(F, X) :
```
SI NON Filevide(F)
        Sauv := F.TêTe
        X := Valeur(F.TêTe)
        F.TêTe := Suivant(F.TêTe)
        Liberer(Sauv)
SINON
        "Underflow"
FSI
```