Piles et Files

Dr. Adel thaljaoui

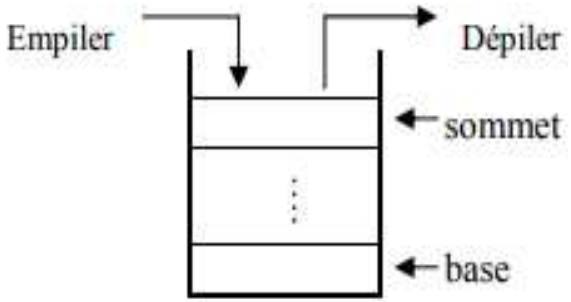
Adel.thaljaoui@gmail.com

Année universitaire 2017//2018

1. Piles

- Une pile est une liste chaînée d'informations dans laquelle :
  - Un élément ne peut être ajouté qu'au sommet de la pile,
  - Un élément ne peut être retiré que du sommet de la pile.

- Il s'agit donc d'une structure de type LIFO (Last In First Out). On ne travaille que sur le sommet de la pile. Les piles sont comparables à des piles d'assiettes.



- On associe à une pile les termes de :
  - **PUSH** pour empiler c'est-à-dire ajouter un élément,
  - **POP** pour dépiler c'est-à-dire supprimer un élément.

- Les piles servent à revenir à l'état précédent et sont utilisées pour :
  - implanter les appels de procédures (pour revenir à l'état d'avant l'appel),
  - annuler une commande,
  - évaluer des expressions arithmétiques,
  - etc.

1.1. Opérations autorisées

- Les opérations autorisées avec une pile sont :
  - empiler, toujours au sommet, et jusqu'à la limite de la mémoire,
  - dépiler, toujours au sommet, si la pile n'est pas vide,
  - vérifier si la pile est vide ou non.

- Déclaration de la pile :

```
Types
   Pile = ^Cellule
              Cellule = Struct
                           Elem : Entier
                           Suiv : Pile
                        FinStruct
Var
P : Pile
```

1.1.1. Empiler

- Empiler un élément revient à faire une insertion en tête dans la liste chaînée.

```
Procédure Empiler(x : Entier ; Var P : Pile)
Variables
Q : Liste
Début
  Allouer(Q)
  Q^.Elem ← x
  Q^.Suiv ← P
  P ← Q
Fin
```

1.1.2. Dépiler

- Dépiler revient à faire une suppression en tête.

```
Procédure Dépiler(Var P : Pile)
Var
X: entier
Q : pile
Début
  Si NON(Pile_Vide(P)) Alors
            x← P^.Elem
            Q← P
            P← P^.Suiv
            Libérer(Q)
   Sinon
            Ecrire(”impossible, la pile est vide”)
  FinSi
Fin
```

1.1.3. Procédures et fonctions de base

Procédure Initialiser(Var P : Pile)
Début
P ← Nil
Fin

Fonction Pile_Vide(P : Pile) : Booléen
Début
Pile_Vide← (P = Nil)
Fin

FONCTION SommetDePile (p : Pile) : entier
Debut
si (PileVide(p)) alors
ecrire ("la pile est vide")
sinon
SommetDePile ← p^.valeur
finsi Fin

2. Files

- Une file, ou file d'attente, est une liste chaînée d'informations dans laquelle :
  - Un élément ne peut être ajouté qu'à la queue de la file,
  - Un élément ne peut être retiré qu'à la tête de la file.

- Il s'agit donc d'une structure de type FIFO (First In First Out). Les données sont retirées dans l'ordre où elles ont été ajoutées.

- Les files servent à traiter les données dans l'ordre où on les a reçues et permettent de :
  - gérer des processus en attente d'une ressource système (par exemple la liste des travaux à éditer sur une imprimante)
  - construire des systèmes de réservation
  - etc.

- Déclaration d'une file :

```
Liste = ^Cellule
    Cellule = Struct
                    Elem : Entier
                    Suiv : Liste
                FinStruct
File = Struct
      Tête : Liste
      Queue : Liste
FinStruct
Variables
F : File
```

2.1. Opérations autorisées

- Les opérations autorisées avec une file sont :
  - enfiler toujours à la queue et jusqu'à la limite de la mémoire,
  - défiler toujours à la tête si la file n'est pas vide,
  - vérifier si la file est vide ou non.

Procédure Initialiser(Var F : File)
Début
F.Tête ← Nil
F.Queue ← Nil
Fin

2.1.1. Enfiler

- Enfiler un élément consiste à l'ajouter en queue de liste. Il faut envisager le cas particulier où la file était vide. En effet, dans ce cas, le pointeur de tête doit être modifié.

```
Procédure Ajouter(x : Entier ; Var F : File)
Variables
  P : Liste
Début
  Allouer(P)
  P^.Elem ← x
  P^.Suiv ← Nil
Si (F.Queue # Nil) Alors
  F.Queue^.Suiv ← P
Sinon
 F.Tête ← P
FinSi
 F.Queue ← P
Fin
```

2.1.2. Défiler

- Défiler est équivalent à dépiler et consiste à supprimer l'élément de tête si la file n'est pas vide. Si la file a un seul élément, il faut mettre à jour le pointeur de queue car on vide la file. Il faut conserver l'adresse de l'élément qu'on supprime pour libérer sa place.

```
Procédure Extraire( Var F : File)
Variables
 P : Liste
D ébut
  Si (F.Tête = Nil) Alors
    Ecrire(”impossible, la file est vide”)
  Sinon
  P ← F.Tête
  x ← F.Tête^.Elem
  F.Tête ← F.Tête^.Suiv
Libérer(P)
FinSi
Fin
```